INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.1224

Fascicle 1/3

(09/97)

SERIES Q: SWITCHING AND SIGNALLING

Intelligent Network

Distributed functional plane for intelligent network Capability Set 2: Part 1

ITU-T Recommendation Q.1224 – Fascicle 1/3

(Previously CCITT Recommendation)

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4 AND No. 5	Q.120–Q.249
SPECIFICATIONS OF SIGNALLING SYSTEM No. 6	Q.250–Q.309
SPECIFICATIONS OF SIGNALLING SYSTEM R1	Q.310–Q.399
SPECIFICATIONS OF SIGNALLING SYSTEM R2	Q.400–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	**Q.1200–Q.1999**
BROADBAND ISDN	Q.2000–Q.2999
General aspects	Q.2000–Q.2099
Signalling ATM adaptation layer (SAAL)	Q.2100–Q.2199
Signalling network protocols	Q.2200–Q.2299
Common aspects of B-ISDN application protocols for access signalling and network signalling and interworking	Q.2600–Q.2699
B-ISDN application protocols for the network signalling	Q.2700–Q.2899
B-ISDN application protocols for access signalling	Q.2900–Q.2999

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION Q.1224

DISTRIBUTED FUNCTIONAL PLANE FOR INTELLIGENT NETWORK CAPABILITY SET 2

Summary

This Recommendation defines the Intelligent Network (IN) Distributed Functional Plane (DFP) architecture for IN Capability Set 2 (IN CS-2). It extends the description of the IN DFP for IN Capability Set 1 (IN CS-1) by inclusion of service management functionality, call-unrelated capabilities, call party handling for more than two parties, and mobility aspects for wireless access.

This Recommendation provides:

– the IN DFP architecture for IN CS-2, encompassing the functional entities related to IN service execution and service management;

– static and dynamic models of the functional entities related to IN service execution and service management (including service switching/call control, service control, call-unrelated, specialized resource, and service data functions), to define how IN service control and service management interact with basic call processing and to understand the nature of the functional entity relationships required for IN CS-2;

– description of call party handling capabilities beyond two-party call set-up and clearing;

– SIB stage 2 descriptions and distributed service logic to identify information flows and functional entity actions for IN CS-2, and the sequencing of the information flows;

– detailed information flow descriptions, including information elements and functional descriptions, as the basis for specifying IN protocols;

– mobility aspects of the DFP, as the basis for defining requirements and information flows for wireless systems.

Within the Q.1220-series Recommendations, this Recommendation describes the DFP in a service and vendor/implementation independent manner, as constrained by the capabilities of the embedded base of evolvable network technology. This provides the flexibility to allocated distributed functionality into multiple physical network configurations, as described in Recommendation Q.1225, and to evolve IN from CS-2 to some future CS-N. It also provides a framework from which IN protocols are specified for IN CS-2, as described in Recommendation Q.1228.

Source

ITU-T Recommendation Q.1224 was prepared by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 12th of September 1997.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

© ITU 1998

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

Page

PART 1

1 General 1
1.1 Normative references 1
1.2 Abbreviations and acronyms 3
2 Scope of IN distributed functional plane for capability set 2 7
2.1 End user access 8
2.2 Service invocation and control 8
2.3 End user interaction 9
2.4 IN service management functionality 9
2.5 Call Party Handling 9
2.5.1 Overview 9
2.5.2 Background and motivation 10
2.5.3 Scope 10
2.5.4 Assumptions 11
2.5.5 Core Capabilities 11
2.6 Internetworking 12
2.7 Security 12
2.8 Out-Channel Call Related User Interaction (OCCRUI) 12
2.9 Out-channel Call Unrelated User Interaction (OCUUI) 13
2.10 Wireless access 13
2.11 Feature interactions 13
3 Distributed functional model for IN CS-2 14
3.1 Explanation of diagram 14
3.2 IN functional model 15
3.3 Definition of functional entities related to IN service execution 16
3.3.1 CCA function (CCAF) 16
3.3.2 CC function (CCF) 16
3.3.3 SS function (SSF) 16
3.3.4 SC function (SCF) 16
3.3.5 SD function (SDF) 17
3.3.6 SR function (SRF) 18
3.3.7 IA function (IAF) 18
3.3.8 CUS Function (CUSF) 19
3.3.9 SCUA function 19

Page

3.3.10 SM function (SMF) 19

3.4 Use of individual relationships between functional entities related to IN service execution 22

3.4.1 SCF-SSF relationship 22

3.4.2 SCF-SCF relationship 22

3.4.3 SCF-IAF relationships 23

3.4.4 SRF-CCF relationship 23

3.4.5 SCF-SRF relationship 23

3.4.6 SRF-SCF relationship 23

3.4.7 SRF-SMF relationship 24

3.4.8 SDF-SDF relationship 25

3.4.9 SCF-SDF relationship 25

3.4.10 SCF-CUSF relationship 26

3.4.11 CUSF-SSF relationship 26

3.4.12 CUSF-CCF relationship 26

3.4.13 SMF-SCF relationship 26

3.4.14 SMF-SDF relationship 28

3.4.15 SMF-SSF/CCF relationship 29

3.4.16 SMF-SRF relationship 31

3.4.17 SMF-SMAF relationship 32

3.4.18 SMF-SCEF relationship 34

3.4.19 SMF-SMF relationship 36

3.4.20 SMF-CUSF relationship 37

3.5 Overview of functional entity call/service logic processing models 39

4 SSF/CCF model 39

4.1 General 39

4.2 Basic Call Manager (BCM) 41

4.2.1 BCSM 41

4.2.2 CS-2 BCSM description 44

4.2.3 BCSM resume points and BCSM transitions in the IN CS-2 call model 70

4.2.4 BCSM indications for the CS-2 call model 80

4.2.5 BCSM detection points 87

4.2.6 DP Criteria 88

4.2.7 Trigger types and trigger precedence 96

4.2.8 DP processing 106

4.2.9 Out-Channel Call Related User Interaction (OCCRUI) 111

4.3 IN-Switching Manager (IN-SM) 113

4.3.1 IN-Switching State Model (IN-SSM) 114

Page

4.3.2 IN-SM Core Capabilities for Call Party Handling 120
4.3.3 The Connection View State (CVS) approach 124
4.3.4 The Hybrid Approach 155
4.3.5 IN-SSM EDPs 160
4.3.6 SSF resource control 160
4.4 Feature Interactions Manager (FIM)/Call Manager (CM) 162
4.4.1 FIM/CM Functions 162
4.4.2 Service logic instance interactions considerations 162
4.4.3 FIM mechanisms 167
4.5 Relationship of SSF/CCF Model Components 169
4.5.1 General 169
4.5.2 Typical sequence of model actions 170
4.6 Relationship of SSF/CCF to SCF 172
5 Specialized Resource Function (SRF) model 172
5.1 General 172
5.2 SRF Components 173
5.2.1 Functional Entity Access Manager (FEAM) 174
5.2.2 Resource Control Part (RCP) 175
5.2.3 Resource Function Part (RFP) 176
5.2.4 Data Part (DP) 176
5.3 Objects of SRF management 176
6 Service Control Function (SCF) model 177
6.1 General 177
6.2 SCF components 177
6.2.1 General 177
6.2.2 Service logic execution manager (SLEM) 179
6.2.3 SCF data access manager 180
6.2.4 Functional routine manager 181
6.2.5 Functional Entity Access Manager (FEAM) 181
6.2.6 SLP manager 181
6.2.7 Security manager 181
6.3 Functional routine categories 182
6.3.1 SLPI management functional routines 182
6.3.2 SLPI communication functional routines 182
6.3.3 Timer management functional routines 182
6.3.4 Data management interface functional routines 182
6.3.5 Asynchronous event handling functional routines 182

Page

6.3.6 Connection management functional routines 182
6.3.7 Specialized resource management functional routines 182
6.3.8 OAM functional routines 182

7 Service Data Function (SDF) model 182
7.1 General 182
7.2 SDF components 183
7.2.1 General 183
7.2.2 SDF data manager 183
7.2.3 Functional entity access manager 183
7.2.4 Security manager 184
7.3 Data types handled by SDF 184

8 Call Unrelated Service Function (CUSF) model 185
8.1 General 185
8.2 Basic Non-Call Manager (BNCM) 186
8.2.1 BCUSM 187
8.2.2 BCUSM description for CS-2 187
8.2.3 Transition for BCUSM 189
8.2.4 BCUSM DP criteria 189
8.3 Description of relationship model 190

9 Service Management Function (SMF) model 191
9.1 General 191
9.2 SMF components 191
9.2.1 General 191

PART 2

10 Mapping of the global functional plane to the distributed functional plane 193
10.1 Mapping of POIs and POSs to DPs and PICs 193

11 Information flow diagrams and distributed service logic in the DFP 198
11.1 Introduction 198
11.1.1 Functional model 198
11.1.2 Description of functional entities 198
11.1.3 Numbering of functional entity actions 199
11.1.4 Relationship with clause 12 (information flow descriptions) 200
11.1.5 Organization of clause 11 200
11.1.6 Generic security information flows 201

Page

11.1.7 SDF-SDF interactions 209
11.1.8 SCF-SCF interactions 212
11.2 SIB stage 2 descriptions 212
11.2.1 ALGORITHM SIB 212
11.2.2 AUTHENTICATE SIB 213
11.2.3 CHARGE SIB 220
11.2.4 COMPARE SIB 231
11.2.5 DISTRIBUTION SIB 232
11.2.6 END SIB 233
11.2.7 INITIATE SERVICE PROCESS SIB 236
11.2.8 JOIN SIB 243
11.2.9 LOG CALL INFORMATION SIB 248
11.2.10 MESSAGE HANDLER SIB 252
11.2.11 QUEUE SIB 267
11.2.12 SCREEN SIB 275
11.2.13 SERVICE DATA MANAGEMENT SIB 277
11.2.14 SERVICE FILTER SIB 282
11.2.15 SPLIT SIB 286
11.2.16 STATUS NOTIFICATION SIB 289
11.2.17 TRANSLATE SIB 297
11.2.18 USER INTERACTION SIB 299
11.2.19 VERIFY SIB 326
11.3 BASIC PROCESS SIBs 327
11.3.1 BASIC CALL PROCESS SIB 327
11.3.2 BASIC CALL UNRELATED PROCESS SIB 336
11.4 Stage 2 description of other distributed functionality 340
11.4.1 Activity test functionality 340
11.4.2 Call gap capability 345
11.5 Distributed service logic 347
11.5.1 SDL diagrams 347
11.5.2 Distributed service logic for SSF 348
11.5.3 Distributed service logic for assist/handoff SSF 371
11.5.4 Distributed service logic for SRF 378
11.5.5 Distributed service logic for SCF 385
11.5.6 Distributed service logic for SDF 440
11.5.7 Distributed service logic for CUSF 448
11.6 Mapping between information flows and SIBs 452

Page

PART 3

12 Relationships between FEs 457

12.1 General 457

12.2 Relationships 457

12.3 Information flows between FEs 458

12.4 SCF-SSF relationship 458

12.4.1 General 458

12.4.2 DP Specific Common Elements 458

12.4.3 Information flows between SCF and SSF 459

12.4.4 IE Definitions for SSF/CCF to SCF Information Flows 512

12.5 SCF-SRF relationship 527

12.5.1 General 527

12.5.2 Information flows between the SCF and SRF 527

12.5.3 IE Definitions for SCF to SRF Information Flows 531

12.6 SCF-SCF relationship 536

12.6.1 General 536

12.6.2 Information flows between the SCF and SCF 536

12.6.3 IE Definitions for SCF to SCF Information Flows 546

12.7 SCF-CUSF relationship 550

12.7.1 General 550

12.7.2 Information flows between the SCF and CUSF 550

12.7.3 IE Description for the SCF-CUSF information flows 556

12.8 SCF-SDF relationship 557

12.8.1 General 557

12.8.2 Information flows between the SCF and SDF 557

12.8.3 IE Description for the SCF-SDF information flows 562

12.9 SDF-SDF relationship 565

12.9.1 General 565

12.9.2 Information flows between the SDF and SDF 565

12.9.3 IE Description for the SDF-SDF information flows 568

12.10 IE Population Rules 569

12.10.1 SSF/CCF to SCF Information Flows 569

12.11 Summary of information flows and related SIBs 583

Page

PART 4

Annex A – Mobility aspects 605

A.1 General 605

A.2 Scope 605

A.3 Mobility Aspects of Distributed Functional Model for IN CS-2 605

A.3.1 Explanation of the diagram 605

A.3.2 Wireless access enhancements to the IN functional model 606

A.3.3 Definitions of wireless access specific functional entities related to IN service execution 607

A.3.4 Use of individual relationships between FEs for wireless access 608

A.4 Example mapping of wireless access FEs to physical platforms 609

A.4.1 Example mapping 1 609

A.4.2 Example mapping 2 610

A.4.3 Example mapping 3 611

A.4.4 Example mapping 4 612

A.4.5 Example mapping 5 613

A.4.6 Example mapping 6 614

A.4.7 Example mapping 7: RCF and CRACF in RS, CURACF in SCP 615

A.4.8 Example mapping 8: RCF and CRACF in RS, CURACF in standalone platform 616

A.4.9 Example mapping 9: RCF in Radio System, CRACF in standalone platform and CURACF in SCP 617

A.4.10 Example mapping 10: RCF in Radio System, CRACF in standalone platform and CURACF in standalone platform 618

Annex B – Telecommunications Management Network (TMN) concepts 619

B.1 Introduction 619

B.2 The TMN functional architecture 620

B.2.1 Operations Systems 621

B.2.2 Work Station Functions 622

B.2.3 The Human Machine Adaptation (HMA) 623

B.2.4 TMN information modelling 623

B.3 Applying TMN concepts to the IN 624

B.3.1 IN management functional model 624

B.3.2 Correspondence between the IN concept of SIB and the TMN concept of MO 626

B.3.3 IN management protocols 628

Page

B.4 Modelling aspects imported from TMN 628
B.4.1 Mappings of IN SMF onto TMN logical layers 628
B.4.2 Mapping of IN SMF onto TMN management functions 631
B.4.3 Mapping of the IN SCEF onto TMN logical layers 632
B.5 IN management and generic TMN management 633
B.5.1 Management process independence 635
B.5.2 SMF complexity 636
B.6 IN SMF-SMF internetworking relationship mapping to TMN 636
B.6.1 Fault management example 636

Annex C – IN SSF Q3 Management Information Model 637
C.1 Introduction 637
C.1.1 Technical approach 637
C.2 SSF functional decomposition 638
C.2.1 Rationale 638
C.2.2 Method 639
C.3 SSF management requirements 641
C.3.1 Rationale 641
C.3.2 Method 641
C.4 SSF management information model 643

Annex D – IN testing and fault management 643
D.1 Introduction 643
D.2 IN testing capabilities for the SSF/CCF 643
D.2.1 Translation Check 643
D.2.2 Trigger Data Check 644
D.2.3 SSF/CCF Query Test 644
D.2.4 Using the SSF/CCF testing capabilities 644
D.3 IN end-to-end testing 645
D.3.1 IN end-to-end testing information elements 646
D.3.2 SSF to SCF 647
D.3.3 SCF to SSF 647
D.3.4 SCF-SRF 647
D.3.5 SRF-SCF 647

Appendix I – Example/application of IN SSF Q3 management information model 647
I.1 Introduction 648
I.2 SSF functional decomposition 648
I.2.1 SSF model 648

Page

I.3 Example to illustrate development of information models and managed object requirements 650

I.3.1 General 650

I.3.2 Example of trigger management information model and managed object requirements 650

I.3.3 SMF functionality mapping to information model 651

I.3.4 SSF/CCF functionality mapping to information model 651

I.3.5 SSF/CCF information model managed object requirements 652

Appendix II – Information flows and call models for terminal mobility 652

II.1 General 652

II.2 Functional entity call/service processing models for wireless access 653

II.2.1 Overview 653

II.2.2 Terminal Access State Model (TASM) 653

II.2.3 Basic Non-Call Associate State Model for CURACF (BNCSM') 656

II.2.4 BCSM 658

II.3 Information flow enhancements for wireless access 658

II.3.1 General 658

II.3.2 Relationships 658

II.3.3 Information flows between FEs 658

II.3.4 SCF-SSF relationship 659

II.3.5 SCF-CRACF relationship 666

II.3.6 SCF-CURACF relationship 672

Recommendation Q.1224

DISTRIBUTED FUNCTIONAL PLANE FOR INTELLIGENT NETWORK CABILITY SET 2

(Geneva, 1997)

PART 1

1 General

General aspects of the DFP are contained in clause 1/Q.1204.

1.1 Normative references

The following ITU-T Recommendations and other references contain provisions which, through references in this text, constitute provisions of this Recommendation, or annexes to this Recommendation. At the time of adoption of this ITU-T Recommendation, the reference editions indicated were valid. All users of this Recommendation are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. Recalling that all Recommendations and other material incorporated by references herein are subject to future revision, all users of this Recommendation are therefore advised that changes in the reference text that constitute future decisions of the work of Organizations or Study Groups other than ITU Study Group 11, do not automatically apply as amended provisions of this Recommendation.

– ITU-T Recommendation M.3010 (1996), *Principles for telecommunications management networks*.

– ITU-T Recommendation M.3320 (1997), *Management requirements framework for the TMN X-Interface.*

– CCITT Recommendation Q.29 (1988), *Causes of noise and ways of reducing noise in telephone exchanges.*

– ITU-T Recommendation Q.700 (1993), *Introduction to CCITT Signalling System No. 7*.

– CCITT Recommendation Q.710 (1988), *Simplified MTP version for small systems*.

– ITU-T Recommendation Q.711 (1993), *Signalling System No. 7 – Functional description of the signalling connection control part*.

– ITU-T Recommendation Q.714 (1993), *Signalling System No. 7 – Signalling connection control part procedures*.

– Recommendation Q.762 (1993) *General function of messages and signals of the ISDN User Part of Signalling System No. 7*.

– ITU-T Recommendation Q.763 (1993), *Formats and codes of the ISDN User Part of Signalling System No. 7*.

– CCITT Recommendation Q.767 (1991), *Application of the ISDN user part of CCITT Signalling System No. 7 for international ISDN interconnections*.

- ITU-T Recommendation Q.771 (1993), *Signalling System No. 7 – Functional description of transaction capabilities*.
- ITU-T Recommendation Q.772 (1993), *Signalling System No. 7 – Transaction capabilities information element definitions*.
- ITU-T Recommendation Q.773 (1993), *Signalling System No. 7 – SCCP formats and codes*.
- ITU-T Recommendation Q.774 (1993), *Signalling System No. 7 – Transaction capabilities procedures*.
- ITU-T Recommendation Q.775 (1993), *Signalling System No. 7 – Guidelines for using Transaction Capabilities*.
- ITU-T Recommendation Q.931 (1993), *Digital Subscriber Signalling System No. 1 DSS 1 – ISDN user-network interface layer 3 specification for basic call control*.
- ITU-T Recommendation Q.932 (1993), *Digital Subscriber Signalling System No. 1 (DSS 1) – Generic procedures for the control of ISDN supplementary services*.
- ITU-T Recommendation Q.1290 (1995), *Glossary of terms used in the definition of intelligent networks*.
- ITU-T Recommendation Q.1400 (1993), *Architecture framework for the development of signalling and OA&M protocols using OSI concepts*.
- ITU-T Recommendation X.500 (1997) | ISO/IEC 9594-1:1997, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.
- ITU-T Recommendation X.501 (1997) | ISO/IEC 9594-2:1997, *Information technology – Open Systems Interconnection – The Directory: The Models*.
- ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1997, *Information technology – Open Systems Interconnection – The Directory: Authentication framework*.
- ITU-T Recommendation X.511 (1997) | ISO/IEC 9594-3:1997, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition*.
- ITU-T Recommendation X.518 (1997) | ISO/IEC 9594-4:1997, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation*.
- ITU-T Recommendation X.519 (1997) | ISO/IEC 9594-5:1997, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications*.
- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*.
- ITU-T Recommendation X.681 (1994) | ISO/IEC 8824-2:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- ITU-T Recommendation X.682 (1994) | ISO/IEC 8824-3:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.
- ITU-T Recommendation X.683 (1994) | ISO/IEC 8824-4:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.
- ITU-T Recommendation X.690 (1994) | ISO/IEC 8825-1:1995, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.
- ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation*.

1.2 Abbreviations and acronyms

This Recommendation uses the following abbreviations.

AC	Application Context
ACN	Application Context Negotiation
ACSE	Application Control Service Element
AD	Adjunct
ADSI	Analogue Display Service Interface
AE	Application Entity
AEI	Application Entity Invocation
AOC	Advice of Charge
APC	Apply Charging
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
API	Application Programming Interface
APR	Apply Charging Report
ASE	Application Service Element
ASR	Automatic Speech Recognition
BCP	Basic Call Process
BCSM	Basic Call State Model
BCUP	Basic Call Unrelated Process
BCUSM	Basic Call Unrelated State Model
BGID	Business Group Identity
BRI	Basic Rate Interface
CAC	Carrier Access Code
CCAF	Call Control Agent Function
CCF	Call Control Function
CDP	Customized Dialling Plan
CHA	Component Handler
CID	Call Instance Data
CM	Call Manager
CMIS	Common Management Information System
CPH	Call Party Handling
CS	Call Segment
CS	Capability Set
CSA	Call Segment Association
CSM	Call Segment Model
CUSF	Call Unrelated Service Function
CVS	Connection View State
DAP	Directory Access Protocol
DET	Determination

DFP	Distributed Functional Plane
DHA	Dialogue Handler
DLE	Destination Local Exchange
DN	Directory Number
DN	Distinguished Name
DP	Detection Point
DSA	Directory System Agent
DSL	Distributed Service Logic
DSP	Directory System Protocol
DSS 1	Digital Subscriber Signalling System No. 1
DTMF	Dual Tone Multi-Frequency
DUA	Directory User Agent
EDP	Event Detection Point
EDP-N	Event Detection Point-Notification
EDP-R	Event Detection Point-Request
EUI	Extended User Interface Server
FCI	Furnish Charging Information
FEA	Functional Entity Action
FEAM	Functional Entity Access Manager
FIM	Feature Interactions Manager
FRL	Facility Restriction Level
FSM	Finite State Machine
GEN	Generation
GFP	Global Functional Plane
GSL	Global Service Logic
GVNS	Global Virtual Network Services
HLSIB	High Level Service Independent Block
IAF	Intelligent Access Function
IEC	International Electrotechnical Commission
IMT-2000	International Mobile Telecommunications-2000
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
INCM	IN Conceptual Model
INDB	IN DataBase
INDBMS	IN DataBase Management System
IN-SM	IN Switching Manager
IN-SSM	IN Switching State Model
IP	Intelligent Peripheral
ISDN	Integrated Services Digital Network
ISDN-UP	ISDN User Part

ISO	International Organization for Standardization
ISUP	Integrated Services Digital Network-User Part
ISUP	ISDN-UP
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
LE	Local Exchange
MACF	Multiple Association Control Function
MSR	Message Storage and Retrieval
NAP	Network Access Point
NEF	Network Element Function
NFA	Network Functional Architecture
NM	Network Manager
NSAP	Network Service Access Point
OFC	Off-line Charging (billing/accounting information)
OLE	Originating Local Exchange
OLI	Originating Line Information
ONC	On-line Charging (user access information)
OSF	Operation System Function
OSI	Open Systems Interconnection
OUT	Output
PIC	Point in Call
PM	Personal Mobility
POC	Point of Control
POI	Point of Initiation
POR	Point of Return
POS	Point of Synchronization
PRI	Primary Rate Interface
PSTN	Public Switched Telephone Network
PTNX	Private Telecommunications Network Exchange
RCP	Resource Control Part
RDN	Relative Distinguished Name
REG	Registration
RFP	Resource Function Part
RLF	Radio Link Function
ROA	Recognized Operating Agency
ROS	Remote Operations
ROSE	Remote Operations Service Element
SACF	Single Association Control Function
SAO	Single Association Object
SCCP	Signalling Connection Control Part
SCE	Service Creation Environment

SCEF	Service Creation Environment Function
SCEP	Service Creation Environment Point
SCF	Service Control Function
SCF FSM	Service Control Function Finite State Machine
SCFID	Service Control Function Identifier
SCI	Send Charging Information
SCME	Service Control function Management Entity
SCME FSM	Service Control function Management Entity Finite State Machine
SCP	Service Control Point
SCSM	Service Control function Call State Model
SDF	Service Data Function
SDF FSM	Service Data Function Finite State Machine
SDL	Specification and Description Language
SDME	Service Data function Management Entity
SDP	Service Data Point
SDSM	Service Data function Call State Model
SF	Service Feature
SIB	Service Independent Building Block
SL	Service Logic
SLCP	Service Logic Control Program
SLMP	Service Logic Management Program
SLP	Service Logic Processing Program
SLPI	Service Logic Processing Program Instance
SM	Service Manager
SMAF	Service Management Access Function
SMF	Service Management Function
SMP	Service Management Point
SMS	Service Management System
SN	Service Node
SRF	Specialized Resource Function
SRF FSM	Specialized Resource Function Finite State Machine
SRME	Specialized Resource function Management Entity
SRSM	Specialized Resource function Call State Model
SS	Service Subscriber
SS7	Signalling System No. 7
SSCP	Service Switching and Control Point
SSD	Service Support Data
SSF	Service Switching Function

SSF FSM	Service Switching Function Finite State Machine
SSME	Service Switching Function Management Entity
SSME FSM	Service Switching Function Management Entity Finite State Machine
SSN	SCCP subsystem number
SSP	Service Switching Point
STI	Service Trigger Information
SU	Service User
TC	Transaction Capabilities
TCAP	Transaction Capabilities Application Part
TDP	Trigger Detection Point
TDP-N	Trigger Detection Point-Notification
TDP-R	Trigger Detection Point-Request
TMN	Telecommunications Management Network
TTS	Text-to-Speech Synthesis
UPT	Universal Personal Telecommunication
VPN	Virtual Private Network
WCR	Wireless Call Related
WCU	Wireless Call Unrelated

2 Scope of IN distributed functional plane for capability set 2

The scope of the IN Distributed Functional Plane (DFP) architecture for IN Capability Set 2 (CS-2) is driven by the service requirements of desired IN CS-2 services, and constrained by the capabilities of the embedded base of evolvable network technology. The scope of functionality required to support desired IN CS-2 services includes functionality to provide:

– end user access to call/service processing;
– service invocation and control;
– end user interaction with service control;
– service management;
– Call Party Handling;
– Internetworking;
– Security;
– Out-Channel Call Related User Interaction;
– Out-Channel Call Unrelated User Interaction;
– Wireless Access; and
– Feature Interactions.

The scope of each of these aspects is addressed below.

2.1 End user access

End user access to call/service processing for IN CS-2 will be provided via the following access arrangements[1]:

– analogue line interfaces;

– ISDN BRI and PRI; and

– traditional trunk and SS No. 7 interfaces.

2.2 Service invocation and control

Call/service processing for IN CS-2 builds upon the current call processing infrastructure of existing digital exchanges. It does so by using a generic model of existing call control functionality to process basic two-party calls, then adding service switching functionality to invoke and manage IN service logic. Once invoked, IN service logic is executed under the control of service control functionality, in conjunction with service data functionality. With this distributed approach to call/service processing, the existing call control functionality retains ultimate responsibility for the integrity of calls, as well as for the control of call processing resources. The following call/service processing constraints apply for IN CS-2:

a) Call control and service switching functionality are tightly coupled, thus the relationship between SSF and CCF is not standardized in IN CS-2.

b) A call is either between two or more end users that are external to the network and addressable via a directory number or combination of directory number and bearer capability, or a call is between one or more end users and the network itself.

c) A call may be initiated by an end user, or by an SCF within the network on behalf of an end user. To supplement a call, IN service logic may either be invoked by an end user served by an IN exchange, or by the network on behalf of an end user.

d) A call may span multiple exchanges. As such, each exchange only controls the portion of the call in that exchange – call processing is functionally separated between exchanges. IN service logic invoked on IN exchanges in such an inter-exchange call are managed independently by each IN exchange.

e) Existing exchanges can be viewed as having two functionally separate sets of call processing logic that coordinate call processing activities to create and maintain a basic two-party call. This functional separation is provided between the originating portion of the call and the terminating portion of the call. This functional separation should be maintained in an IN exchange to allow IN service logic invoked on the originating portion of the call (i.e. on behalf of the calling party) to be managed independently of IN service logic invoked on the terminating portion of the call (i.e. on behalf of the called party).

f) It is desirable to allow multiple IN-supported service logic instances to be simultaneously active for a given end user. It is also recognized that non-IN service logic will continue to exist in the network. As such, service feature logic instances mechanisms for IN CS-2 should:

   – determine which service logic to invoke for a given service request. This mechanism should select the appropriate IN-supported service logic or non-IN-supported service logic, and block the invocation of any other service logic for that particular service request;

[1] This does not preclude the use of these interfaces to support access from private or mobile networks.

– limit simultaneously active IN- and non-IN-supported service logic instances;

– ensure that simultaneously active IN-supported service logic instances adhere to the single-ended, single point of control restriction on IN CS-2 service processing.

g) The distributed approach and added complexity of call/service processing for IN CS-2 requires mechanisms for fault detection and recovery, allowing graceful termination of calls and appropriate treatments for end users.

2.3 End user interaction

End user interaction with the network to send and receive information is provided by service switching and call control resources, augmented by specialized resources. These specialized resources are controlled by service control functionality, and are connected to end users via call control and service switching functionality.

2.4 IN service management functionality

IN service management functionality is used to provision and manage the service control functionality, service data functionality, specialized resource functionality and the combined service switching/call control functionality in the network, outside of the context of call/service processing. The IN management functions are modelled according to the TMN functional architecture.

Recommendation M.3010 specifies the functional, information and physical architectures that support the TMN management services. A TMN management function is the smallest part of the TMN management service as perceived by the user of the service. It will generally consist of a sequence of actions on a defined managed object or objects. The TMN management functions are described in Recommendation M.3400.

Many of the IN management functions are not IN specific but can be mapped onto TMN management functions and reused for IN purposes.

Furthermore, the TMN uses OSI System Management Functions. These functions provide generic management controls and capabilities which can be used by specific TMN management services and functions. As a consequence of what is described above, a set of managed objects which are defined in the different system management functions, can be identified and reused for IN purposes.

In particular one can point out: "Summarization Function" (Recommendation X.738), Workload Monitoring Function" (Recommendation X.739), "Event Management Function" (Recommendation X.734), "Alarm Reporting Function" (Recommendation X.733) and "Performance Management Function" (Recommendation Q.822).

2.5 Call Party Handling

2.5.1 Overview

The Call Party Handling network aspect enables IN to influence multi-party calls. Call Party Handling for IN CS-2 includes three interrelated aspects:

1) Connection View States;

2) the Hybrid approach; and

3) Object-Oriented Modelling using SDLs.

The Connection View State approach is based on the Connection View (CV), as described in Recommendation Q.1214. The Connection View objects provide an SCF with a generic view of call and connection processing. Each Connection View State models a CV state in an SSF. Thus, the set of Call Party Handling functionality supported by an SSF can be defined as a catalogue of

Connection View States. The Connection View State model is evolvable and scaleable. Thus, the catalogue of Connection View States may be extended to describe future service and capability needs.

The Hybrid approach builds on the Connection View States supported by an SSF. To provide more complex Call Party Handling services, the Hybrid approach enables IN services that combine the Connection View State capabilities of an SSF with a bridging SRF in an external platform. A service that uses the Hybrid approach requires SRF capabilities only for those aspects of the service not covered by the Connection View State catalogue.

The Object-Oriented Modelling of the IN CS-2 Application Protocol defines the protocol for the Call Party Handling Recommendations by enabling validation of the Connection View representation, on which the Connection View States are based. SDLs are used to model the objects of the Connection View. These SDLs can be validated to ensure that the Connection View model is implementable. An added benefit is the ability to automatically create a test suite from these SDLs using the appropriate software tools.

2.5.2 Background and motivation

Recommendations Q.1211 and Q.1221 identify the IN CS-1 and IN CS-2 benchmark services, respectively. Call Party Handling could help satisfy the needs of the benchmark services that require that an SCF be able to influence the connections within a multi-party call. Examples of these services include Conference Calling, Call Hold, Call Transfer and Call Waiting.

Appendix I/Q.1214 and Appendix II/Q.1218 introduce the Call Party Handling concept, and suggest a partial solution. IN CS-2 provides a more complete solution to Call Party Handling by including details on the Connection View State approach, the Hybrid approach, and an Object-Oriented Modelling of the Connection View.

2.5.3 Scope

For IN CS-2, Call Party Handling focuses on:

- narrow-band access;
- trigger types and trigger processing;
- event detection points;
- two DFP modelling approaches: connection view state and hybrid;
- complementary aspects of these approaches;
- object-oriented modelling using SDLs for protocol definition and validation;
- protocol aspects (syntax and semantics/procedures);
- GFP modelling of call party handling.

The scope of the *Connection View State approach* for Call Party Handling is basic mid-call event processing plus two-party and multi-party call processing capabilities. This includes SSF/CCF capabilities to:

- detect end user signalling events during call processing for a basic two-party call (e.g. ISDN feature activation/indication, non-ISDN switch-hook flash).
  - interrupt call processing and notify an SCF of these events;
  - allow an SCF to inform an SSF/CCF on how to resume call processing during the stable phase of a basic two-party call (e.g. to hold and retrieve a party, to connect a party to a specialized resource, to perform a transfer of a party) or multi-party call (e.g. for call waiting, consultation hold, call transfer, and conference);

– address switch-based feature interactions related to these events prior to SCF involvement, and subsequent to SCF involvement;
– perform fault handling procedures.

The scope of the *Hybrid approach* for Call Party Handling is basic mid-call event processing plus SRF multi-party call processing capabilities. This includes SRF capabilities to:

- bridge multiple parties onto a single call;
- allow an SCF to inform an SRF on how to add or remove parties from a call.

The scope of the *Object-Oriented SDL* specifications is to define and validate the protocol for Call Party Handling for both of the above approaches.

2.5.4 Assumptions

The following limitations and assumptions apply to the scope for Call Party Handling capabilities:

1) The capabilities are intended to be defined in a way that:
   - minimizes the impact on existing IN capabilities and messages;
   - minimizes the amount of information passed between an SSF/CCF and an SCF;
   - minimizes the amount of information passed between an SRF and an SCF;
   - is consistent with existing switching system capabilities; and
   - yields uniformity of procedures for analogue and ISDN users.
2) The capabilities are intended to be backward compatible with IN CS-1.
3) A single point of control is assumed. That is, more than one SCF, or SCF and a switch-based feature, cannot simultaneously act on a particular BCSM.
4) Transfer of control from an SSF/CCF to an SCF for Call Party Handling can only occur from well-defined states.
5) The capabilities assume the existence of switch-based multi-party features, and build on these features in certain situations.

2.5.5 Core Capabilities

Generic, core capabilities required at the SSF/CCF to support call party handling. There are four basic capabilities that the SSF/CCF must be able to perform, for call party handling as described in the following subsections.

2.5.5.1 Core Capability 1

Core Capability one is the ability for a user to initiate midcall interactions and provide additional input. In order for IN to influence the call and perform call party handling, there must be a method for IN to detect the user hookflash or feature activation/indication (i.e. Midcall Detection Point), during an active call, from the A-party. The SSF/CCF must provide this information to the SCF. Since this action can be a prelude to a large number of potential IN service/features, a mechanism must exist to convey information from the user to the SCF in order to determine the appropriate subsequent steps to be taken by IN. A mechanism is required to allow the user to input appropriate information.

Possible support of this capability can be through the use of display phones with feature selection, use of SRFs to play announcements and to collect digits, or by offering dial tone to the user for the input of service codes. Although various possibilities exist to support this capability, that which is standardized should lend support to both analogue as well as ISDN BRI and PRI access (Analogue trunks are not defined for IN CS-2).

2.5.5.2 Core Capability 2

Core Capability two is the ability to connect to a resource/transfer each call party. The IN call party handling capabilities include the ability to independently route each party participating in a given call to a point external to the SSF/CCF. Each call party will independently require transport, with coordination, provided by IN, to provide correlation of the independent transport paths and the intended destinations. The call parties of a given call can be transferred at the same time or over a period of time.

2.5.5.3 Core Capability 3

Core Capability three is the ability to present the current half call view to the SCF. The SCF should be able to observe the present status of the call at a given SSF/CCF based on the originating or terminating half call model. This view should be sufficient for service logic within the SCF to be able to determine the status of current connections to the controlling leg. This view should be able to provide information for each call party relating to the Basic Call State Model (BCSM), as well as the current conditions or events. Precise level of call topology information needed by the SCF is not defined in IN CS-2.

2.5.5.4 Core Capability 4

Core Capability four is the ability for the SSF/CCF to combine selected transferred paths within the SSP into a Call. A mechanism must exist that will allow the SSF/CCF to regain control of the individual transport paths of the call parties, after a transfer or connection to resource has been made. Upon notification from IN service logic, the SSF/CCF is to combine selected individual paths within the SSP into a call on behalf of the controlling leg. The form of the resultant call is limited by the connection capabilities provided by the SSF/CCF. For example, a given SSF/CCF may only support a two-way connection and another SSF/CCF may support both two-way and three-way connections.

2.6 Internetworking

IN CS-2 identifies the SCF-SCF, SMF-SMF and SDF-SDF relationships in addition to the IN CS-1 SCF-SDF relationship for internetworking purposes. Distributed service logic, but not distributed service control, is supported. Additionally, internetwork management interactions and distributed data handling processes are supported.

2.7 Security

IN CS-2 supports security through the provision of a number of security assisting functions. A security assisting function requires the cooperation of two or more functional entities. The security assisting functions support both internal network operations and interworking between two or more networks. Security features do not automatically guarantee network integrity, but rather they are some of the tools which allow secured systems to be built.

2.8 Out-Channel Call Related User Interaction (OCCRUI)

The "Out-Channel Call Related User Interaction" (OCCRUI) network aspect is used to provide new IN services requiring the transparent transfer of information between a User and a Service Logic. Functionally, it is composed of two Information Elements which are UTSI in the "User to Service Logic" direction and STUI in the "Service Logic to User" direction.

The OCCRUI network aspect is defined in the following way:

– **Service independent transfer of information**: the OCCRUI mechanism is a generic mechanism. At the UNI and at the NNI, the network does not analyse the information

conveyed; the network has just to recognize the OCCRUI Information Elements and send them to the SSF. Therefore the introduction of a new IN service does not imply enhancements of the network. In other words the STUI/UTSI Information Elements are just "envelopes".

– The UTSI/STUI information is **conveyed within the Basic Call Control messages and within the Transport message** (during the Active and Alerting phases of the call). In particular, the OCCRUI network aspect does not require additional signalling "call state machine" nor additional Call Control messages.

– The User terminal should handle this network aspect: it may be an Intelligent Terminal or an existing Terminal that has been enhanced.

2.9 Out-channel Call Unrelated User Interaction (OCUUI)

Call Unrelated User Interaction between the user and the service processing is the ability to communicate information outside the context of a call, required for services/service features such as message waiting indication, registration of the user location. As this capability is not supported by the BCSM, which is Call Related, the Basic Call Unrelated State Model (BCUSM) is introduced in IN CS-2.

2.10 Wireless access

IN CS-2 provides for the support of both wireline and wireless access by users, through the functional level. Requirements are driven by such systems as IMT-2000 but are not limited to this wireless access technology. Functional Entity descriptions, relationships and information flows are provided. For IN CS-2 the access of a wireless user is modelled separately from the access of a wireline user.

2.11 Feature interactions

The feature interaction mechanisms provided in IN CS-2 are a first step beyond the limitations of IN CS-1 towards the ultimate goal to support feature/service cooperation and to avoid feature/service interference. These mechanisms are means to manage feature/service interferences which have been recognized prior to service deployment. Use of these mechanisms is a network operator option.

The following cases of feature/service interactions can be distinguished which may require different types of feature/service interaction handling:

Case A: IN based – switch-based: an IF from an IN service logic to a switch-based service logic may be required.

Case B: switch-based – IN-based: an IF from a switch-based service logic to an IN-based service logic may be required.

Case C: IN-based – IN-based: an IF between IN-based service logics may be required.

Recommendation Q.1214 already provides general descriptions for feature interaction handling in the SSF with respect to all three cases. The specific mechanisms of IN CS-2 are additions to these descriptions.

Case A: IN-based – switch-based: This case shall be treated by the use of the INServiceInteractionIndicator mechanism. This mechanism builds on (standardized) basic network services. It allows an IN service logic to allow/deny or modify switch-based service logic execution via call-related signalling. The INServiceInteractionIndicator mechanism is extensible, so that new service interactions may be covered as required. New service interactions may arise due to the

introduction of new basic network services or new IN services. In case the INServiceInteractionIndicator mechanism is used across network boundaries, the relevant signalling and the relevant procedures need to be standardized or bilaterally agreed.

Case B: switch-based – IN-based: Treating this case similar as case C would be possible without any additional requirements on the SSF and on the INAP. However, this is not defined for IN CS-2.

Case C: IN-based – IN-based: In IN CS-2, this case can occur only, if the two IN service logics are not triggered in the same BCSM. For IN-IN feature interference handling, there are two different approaches, which may be applied either singly or in combination.

a) *SSF- and/or single SCF-oriented approach*

The triggering mechanisms are used and enhanced to avoid feature interferences. In IN CS-1 trigger precedence rules are already in place, which may be used to avoid feature interferences. In IN CS-2, service compatibility checks shall be added to the triggering mechanism descriptions. The service compatibility check in the SSF is based on the INServiceCompatibilityIndication of the active service logic and of the newly triggered service logic. The network operator may assign an INServiceCompatibilityIndication to an IN service via SSF administration. During triggering, the SSF performs a generic compatibility check procedure using an administrable exclusion matrix for INServiceCompatibilityIndications.

For each triggering of an IN Service during a call the according INServiceCompatibilityIndication is derived and shall be propagated through the network. For the case an INServiceCompatibilityIndication was already received from the network, the new derived INServiceCompatibilityIndication is added to the received one. The ISUP is required to convey the INServiceCompatibilityIndication. In case the INServiceCompatibilityIndications are used across network boundaries, the relevant signalling and the relevant procedures need to be standardized or bilaterally agreed.

The INServiceCompatibilityIndication shall be sent in the initial INAP message to the SCF for information purposes. For the given IN Service the INServiceCompatibilityIndication derived during triggering may be overwritten using the parameter INServiceCompatibilityResponse. Whether or not the overwrite is allowed is a network operator specific option.

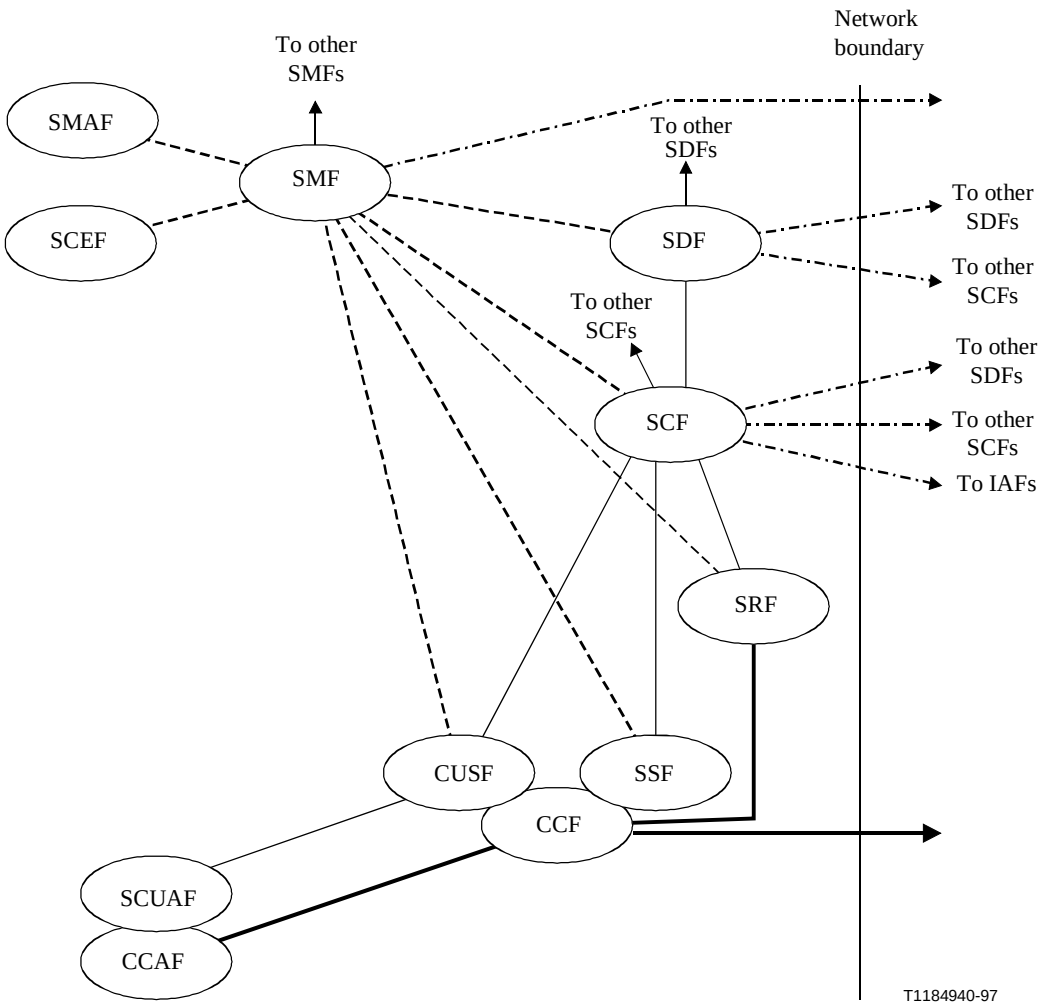
b) *Two SCF-oriented approach*

This approach is based on the exchange of information between the two involved SCFs (bidirectional). This approach is not defined for IN CS-2.

3 Distributed functional model for IN CS-2

3.1 Explanation of diagram

Figure 3-1 identifies the IN DFP model for IN CS-2. This diagram depicts the functional entities and relationships applicable to IN CS-2. This diagram is a subset of the generic IN DFP model described in clause 2/Q.1204. A general explanation of functional entities, relationships, and the diagram are contained in 2.1/Q.1204.



- - - - - - - -	Management Relationship
————	IN Service Control
━━━━	Bearer Connection Control
-·-·-·-·-	Internetworking Relationship

CCAF	Call Control Agent Function
CCF	Call Control Function
IAF	Intelligent Access Function
CUSF	Call Unrelated Service Function
SCEF	Service Creation Environment Function
SCF	Service Control Function
SCUAF	Service Control User Agent Function
SDF	Service Data Function
SMAF	Service Management Access Function
SMF	Service Management Function
SRF	Specialized Resource Function
SSF	Service Switching Function

Figure 3-1/Q.1224 – IN CS-2 DFP architecture

3.2 IN functional model

As stated in 3.1, the IN DFP for IN CS-2 is a subset of the general IN DFP. In particular:

– only the CCAF, CCF, SSF, SCF, SDF, SRF and SMS functional entities are included;

– SMS-SCEF and SMS-SMAF relationships are not addressed.

3.3 Definition of functional entities related to IN service execution

3.3.1 CCA function (CCAF)

The CCAF is the Call Control Agent (CCA) function that provides access for users. It is the interface between user and network call control functions. It:

a) provides for user access, interacting with the user to establish, maintain, modify and release, as required, a call or instance of service;

b) accesses the service-providing capabilities of the Call Control Function (CCF), using service requests (e.g. set-up, transfer, hold, etc.) for the establishment, manipulation and release of a call or instance of service;

c) receives indications relating to the call or service from the CCF and relays them to the user as required;

d) maintains call/service state information as perceived by this functional entity;

e) interfaces to the SCUAF for call-unrelated service if needed.

3.3.2 CC function (CCF)

The CCF is the Call Control (CC) function in the network that provides call/service processing and control. It:

a) establishes, manipulates and releases call/connection as "requested" by the CCAF;

b) provides the capability to associate and relate CCAF functional entities that are involved in a particular call and/or connection instance (that may be due to SSF requests);

c) manages the relationship between CCAF functional entities involved in a call (e.g. supervises the overall perspective of the call and/or connection instance);

d) provides trigger mechanisms to access IN functionality (e.g. passes events to the SSF);

e) manages basic call resource data (e.g. call references).

3.3.3 SS function (SSF)

The SSF is the service switching (SS) function, which, associated with the CCF, provides the set of functions required for interaction between the CCF and a service control function (SCF), and associated with the NCSF for the call-unrelated service handling if necessary. It:

a) extends the logic of the CCF to include recognition of service control triggers and to interact with the SCF;

b) manages signalling between the CCF and the SCF;

c) modifies call/connection processing functions (in the CCF) as required to process requests for IN provided service usage under the control of the SCF;

d) interfaces to the CUSF for handling call-unrelated interactions;

e) supports the relay case, in which it ensures the relay of information between the SCF and SRF possibly using the Out-Channel Call Related User Interaction (OCCRUI) capabilities.

3.3.4 SC function (SCF)

The SCF is a function that commands call control functions in the processing of IN provided and/or custom service requests. The SCF may interact with other functional entities to access additional logic or to obtain information (service or user data) required to process a call/service logic instance. It:

a) interfaces and interacts with Service Switching Function/Call Control Function (SSF/CCF), Specialized Resource Function (SRF), Service Data Function (SDF), other Service Control Functions (SCF), and Call Unrelated Service Function (CUSF) functional entities;

b) contains the logic and processing capability required to handle IN provided service attempts, both related to a call and not related to a call;

c) interfaces and interacts with other SCFs in a secured fashion for distributed service control and unsolicited service notifications. As a consequence of distributed service control, the result of service logic execution is transferred between two SCFs;

d) interfaces and interacts with SDFs for secured data acquisition and manipulation of data;

e) provides a point of interconnection to the network for the purpose of internetworking, effectively hiding the specific structure of the network;

f) interfaces and interacts with SRF for call-related interactions by indicating to the SRF the User Interaction script to be run, by providing to the SRF the additional information it requests during the User Interaction script execution and by waiting for the end of the User Interaction script execution;

g) interfaces and interacts with SRF for call-unrelated interactions by monitoring the availability of resources at the SRF, requesting control of some SRF resources outside the context of a call;

h) provides security mechanisms, for the purposes of internetworking, to enable secured information transfer across the boundary between networks.

3.3.5 SD function (SDF)

The SDF contains customer and network data for real-time access by the SCF in the execution of an IN provided service. For example, the SDF may store both user and terminal related data. It:

a) interfaces and interacts with SCFs for secured manipulation and acquisition of data through simple database requests of data management scripts;

b) interfaces and interacts with other SDFs as required, enabling the hiding of data location in the network. This knowledge can be used for data distribution transparency (e.g. to the SCF);

c) provides security mechanisms, for the purposes of internetworking, to enable secured information transfer across the boundary between networks;

d) interfaces and interacts with other SDFs enabling copying of data together with the access rights to the data;

e) provides authentication and access control facilities for providing secure access to service data;

f) facilitates the cooperation of traffic management to prevent or solve congestion situation in data acquisition;

g) provides data support for security services. This data support can be used by the SDF itself for secured data management;

h) facilitates the cooperation of a robust recovery mechanism for copying of data (e.g. in the case the SDF is unavailable);

i) provides data access scripts (methods) which may be invoked by the SCF in order to simplify the information transfer via the SCF-SDF interface. Such kind of data access scripts do provide simplified data manipulation on an entry. The SCF continues to provide service specific processing logic and command call control functions in the SSF.

NOTE – The SDF contains data relating to the provision or operation of IN provided services. Thus it does not necessarily encompass data provided by third party such as credit information, but may provide access to this data.

3.3.6 SR function (SRF)

The SRF provides the specialized resources required for the execution of IN provided services (e.g. digit receivers, announcements, conference bridges, etc.). It:

a) interfaces and interacts with SCF and SSF (and with the CCF);

b) may contain the logic and processing capability to receive/send and convert information received from users;

c) may contain functionality similar to the CCF to manage bearer connections to the specialized resources.

3.3.6.1 SRF Automatic Speech Recognition (ASR)

The ASR resource allows the IN services user to input commands and data in his/her own voice. ASR can be both speaker-independent and speaker-dependent. In case of speaker-dependent ASR, a mechanism should be provided which enables the user to directly manage his/her voice templates used for recognizing commands and data: such a mechanism should allow the user to review, update, delete and insert both:

– the voice templates; and

– the correspondences between the templates and the SRF internal format of the recognized voice (e.g. between a voice input name and the corresponding string of ASCII characters).

This mechanism could be either controlled by the SCF or directly performed by the SRF with no intervention by the SCF. In the latter case, the SRF would inform the SCF of the result of the operation, should this have been requested by the SCF. The basic ASR resource should provide for the recognition of isolated words (i.e. the ten digits and a number of basic commands such as "yes" and "no" spoken at least in the local network provider language) in a speaker-independent manner over the PSTN.

Considering that multilingual ASR could also be useful, it is recognized that the SRF should handle the indication of the requested language to be used for voice inputs, in the same way as announcement generation described above.

3.3.6.2 SRF Text-to-Speech

The SRF can have a Text-to-Speech (TTS) function. This functionality consists of two logical functions. The first function converts the input text in a phonetics-prosodic representation. The second function produces the synthesized voice signal, processing and connecting of voice elements.

3.3.7 IA function (IAF)

The Intelligent Access Function provides access between the SCF of an IN-structured network and an entity which is not an IN-structured network. This latter entity may be other networks or customers (private networks, simple databases used for instance in the CCR service, terminals and PABXs). It:

a) provides access to and from the SCF of the IN-structured network;

b) maps the information between the internal and external representation;

c) resides in the entity which is not an IN-structured network.

3.3.8 CUS Function (CUSF)

The CUSF is the Call Unrelated service (CUS) function which, associated with the CCF and the SSF, provides a set of call-unrelated service functions required for out-channel interaction with a SCUAF. It also provides the set of functions required for interaction between the SCUAF and a SCF. It:

a) establishes, manages and releases the relationship between the instance in the SCUAF and the network for the call-unrelated interaction between users and service processing;

b) recognizes a call-unrelated service control trigger and interacts with the SCF;

c) provides the trigger mechanisms for call-unrelated interaction to access IN functionality (e.g. manages call-unrelated interaction events and passes them to the SCF);

d) modifies call-unrelated interaction processing functions (in the CUSF) as required to process requests for IN provided service usage under the control of the SCF;

e) supports call-unrelated userinteraction which may be user initiated or SCF initiated.

3.3.9 SCUA function

The SCUAF is the Service Control User Agent (SCUA) function that provides access for users. It is the interface between a user and the Call Unrelated Service Function (CUSF). It:

a) provides for user access, interacting with the user to establish, maintain, and release, as required, an instance of call-unrelated service;

b) accesses the functions for call-unrelated interaction processing in the Call Unrelated Service Function (CUSF), and the service invocation capabilities of the CUSF, using service requests (e.g. location registration, attach. etc.) for the invocation of call-unrelated services;

c) receives indications relating to call-unrelated services from the CUSF and relays them to the user as required;

d) maintains service state information as perceived by this functional entity.

NOTE – Whether the SCUAF abstracts a new FE for call associate supplementary services is not defined for in IN CS-2. IN CS-2 does not define either what relationship should be used to model call-related user interaction.

3.3.10 SM function (SMF)

The SMF is the Service Management Function. This subclause describes a number of IN SMF functionalities. These functions can be grouped into five categories:

- Service Deployment Functions;
- Service Provisioning Functions;
- Service Operation Control Functions;
- Billing Functions;
- Service Monitoring Functions.

1) Service Deployment Functions include:

   – Service Scripts Allocation:

     This subfunction passes the service scripts and determines for which part of the network the service scripts are relevant and manages the relevant network elements.

– Service Generic Data Allocation:

  This subfunction passes the service generic data and determines for which part of the network the service generic data are relevant and manages the relevant network elements.

– Signalling Routing data Introduction and Allocation:

  This subfunction passes the signalling routing data and determines for which part of the network the signalling routing data are relevant and manages the relevant network elements. It downloads the signalling routing data into the SS7 network and determines the relevant SS7 network elements for allocation of the signalling routing data.

– Trigger Data Introduction and Allocation:

  This subfunction passes the trigger data and determines for which part of the network the trigger data are relevant and manages the relevant network elements. It downloads the trigger data into the PSTN.

– Specialized Resource Data Introduction and Allocation:

  This subfunction passes the specialized resource data and determines for which part of the network the specialized resource data are relevant and manages the relevant network elements.

– Service Testing:

  This subfunction collects the service software from the Service Creation Environment Function to be loaded into a stand-alone IN-network, in order to test the newly developed service. The function enters service and service subscriber specific data. It performs management related test operations.

2) Service Provisioning Functions include:

– Customer Specific Data Introduction and Allocation:

  This subfunction collects service subscriber specific data and administrates that in subscriber databases and contract databases. The function translates the service and subscriber data into network specific data. This subfunction determines for which part of the network the data are relevant and manages the relevant network elements.

3) Service Operation Control Functions include:

– Service Maintenance:

  Service Maintenance included the following functionality:

  • Software Maintenance:

    Software maintenance consists of the modification of service logic (modification of service logic is a function of the SCEF). Introduction of modified script in the IN structured network is done in service deployment.

– Updating Service Generic Data:

  This subfunction passes the service generic data and determines for which part of the network the service generic data are relevant and manages the relevant network elements.

– Updating Customer Specific Data:

  This subfunction provides the control functions for service subscriber specific data and administration to those subscriber databases and contract databases. This subfunction determines for which part of the network the data are relevant and manages the relevant network elements.

– Updating Signalling Routing Data:

  This subfunction provides the control functions for the signalling routing data and determines for which part of the network signalling routing data are relevant and manages the relevant network elements. It downloads the signalling routing data into the SS7 network and determines the relevant SS7 network elements for allocation of the signalling routing data.

– Updating Trigger Data:

  This subfunction provides the control functions for the trigger data and determines for which part of the network the trigger data are relevant and manages the relevant network elements. It downloads the trigger data into the PSTN.

– Updating Specialized Resource Data:

  This subfunction provides the control functions for the specialized resource data and determines for which part of the network the specialized resource data are relevant and manages the relevant network elements.

– Adjustment of the SMAF:

  The service subscriber/network operator interface to the SMF is provided by the SMAF. The interface to the service subscriber and network operator has to be accommodated to the adjustments in their data. For instance, a service subscriber who has changed peripheral-type (customer specific data: DTMF-telephone to VTX-terminal). This change of peripheral also may cause a change of menu options.

– Service Reconfiguration:

  This activity consists of the re-allocation of service scripts, service generic data and customer specific data. For instance, the reason for service reconfiguration could be a change in the network configuration or improvement of the performance of services.

– Service (de)activation:

  This activity gives the network operator the possibility to (de)activate (part of) a service temporarily. For instance, for maintenance purposes a televoting service which is only used on set times.

– Service dismantlement:

  A service will be taken out of operation.

– Security:

  In the SMF, two types of security can be distinguished: access control and data control. Access control covers the identification, authentication and authorization (command control) of both service subscriber and network operator. Data control covers the control of the input of data by both the service subscriber and the network operator.

4) Billing Functions include:

– Generating and Storing Charging Records:

  This subfunction monitors the service usage. This function logs the call records.

– Collecting Charging Records:

  This subfunction collects the call records and the management detail records. Then it uniforms and correlates them. This function logs the call records.

– Modification of Tariffs:

  This subfunction determines the tariff structure and the tariff for a newly developed service or changes them for an existing one.

5) Service monitoring includes:

– Initiating Measurements and Collecting Measurement Data:

  This subfunction monitors the service usage and service performance. It also monitors network performance. Therefore, it needs measurement results from the underlying parts: the SS7 management function and Network management function.

– Analysis and Reporting of Measurement Data:

  This subfunction analyses the service usage and service performance. It also analyses the results of the initiation and collection measurement data function.

– Receive information from fault monitoring data:

  This subfunction acts upon the receipt of fault monitoring data from Network elements. Implications and impact on Service performance will be computed and the appropriate action taken.

3.4 Use of individual relationships between functional entities related to IN service execution

3.4.1 SCF-SSF relationship

The SCF-SSF relationship is used for information flows between an SCF and an SSF in the public network. This relationship, with the aid of the SCF-SDF/SRF, and SDF-SDF relationship, provides a variety of services, service features like Free-phone, UPT, VPN, etc. Details of service drivers can be found in Recommendation Q.1221.

3.4.2 SCF-SCF relationship

This SCF-SCF relationship is used for information flows between two SCFs in the public network. The relationship supports call-related operations, which can be for intranetwork or internetwork purposes.

For IN CS-2, the SCF-SCF relationship supports information flows for services during call processing.

For IN CS-2 the SCF-SCF relationship is also one of a limited set of relationships to support internetworking. As such, it provides a point of interconnection to the network, effectively hiding the specific structure of the network and providing access security to the network from other public networks.

The SCF-SCF relationship is used for various purposes. Each of these purposes can apply to the intranetwork, or internetwork cases. Specifically, the relationship is used for the following:

a) **Distribution of Service Logic**: between SCFs. It is possible that a given SCF may only contain a service logic program to complete part of a service, such as Customized Call Routing. Service completion may rely on the outcome of service logic in another SCF to be performed. This will require coordination and synchronization between the SCFs and may be provided by explicit, solicited notifications.

   In case of error situations, generic error indications returned locally (within the same network) to the SCF by its associated SDF may be interpreted to service specific denial causes by the SCF of that network and can be sent as such to the SCF of the interconnecting network.

b) **Unsolicited Notifications**: of events taking place within the domain of a given SCF are provided to other SCFs in an unsolicited manner. For example, after an internetwork

handover when the initial network provides the anchor point, subsequent handovers may require notifications to the initial network to optimize internetwork connections.

3.4.3 SCF-IAF relationships

Because of the different nature of IAF, in IN CS-2, two different relationships are defined between the SCF and IAF:

– when the IAF belongs to another network;

– when the IAF belongs to a customer (examples of customers are listed in 3.3).

These relationships differ from the point of view of security, charging or reliability requirements.

3.4.4 SRF-CCF relationship

At present, it is considered that the CCF is under control of the SSF: a SRF-SSF/CCF relationship exists for connection control to specialized resources; in IN CS-1, this relationship is supported by the interface protocols identified in Recommendation Q.1218, which would also be used for the IN CS-2 SRF.

The relationship between the IN CS-2 SRF and the CCF is not defined for IN CS-2. (e.g. in the case of releasing a connection on which a resource is used). The SRF may contain functionality similar to the CCF to manage bearer connections to specialized resources, but no call model is specified.

3.4.5 SCF-SRF relationship

This SCF-SRF relationship is used when the SCF delegates some of the service logic to the SRF, which executes a specialized type of service logic, known as *User Interaction-scripts*. This can avoid long response times, which are unavoidable if functions are physically distributed over two networks nodes, the SCP and the IP.

User Interaction-scripts allow the grouping of the user interaction parts of the service into functional blocks which use SRF resources in the most efficient way. The transition from one to the other is triggered by internal results (e.g. error condition) or external decision (e.g. user choice, or result from a database interrogation).

Each of the User Interaction-scripts represents a generic action which may be parameterized. For instance, user authentication is a generic action, which may be described independently of, say, the length of the user identity and the length of the PIN. These two numbers are parameters of the User Interaction-scripts.

User Interaction-scripts can take into account the experience acquired in the ergonomy of vocal services, thus offering state-of-the-art user interaction which can be reused in new services.

3.4.6 SRF-SCF relationship

This subclause describes the relationship between the SRF and the SCF for connection to and control of specialized resources and User Interaction-scripts.

- *Call-Related Interaction:*

  For all cases in which call-related interaction is considered, the establishment of this relationship must be preceded by the establishment of a relationship between the SCF and the SSF. In IN CS-1, the user-interaction is defined as a chain of "Questions and Answers", which result in the SCF sending several commands to the SRF in sequence. This process is optimized by making the IN CS-2 SRF perform an indivisible block of "Questions and Answers" called a "User Interaction-script". This User Interaction-script offers "On-line" help, DTMF codes receipt anticipation and even "management by interruption".

Introducing the User Interaction-script concept extends SRF capabilities in such a way that the IN CS-1 defined SRF capabilities are a subset of the new SRF capabilities. The IN CS-2 SRF receives from the SCF the indication of the particular User-Interaction script that is to be run, executes it and is responsible for the whole user-interaction. The SRF then communicates the results back to the SCF. Depending on the User Interaction script to be run, the SRF may request to the SCF additional information and receive them during the User Interaction script execution. This kind of SCF/SRF relationship is illustrated by Figure 3-2.

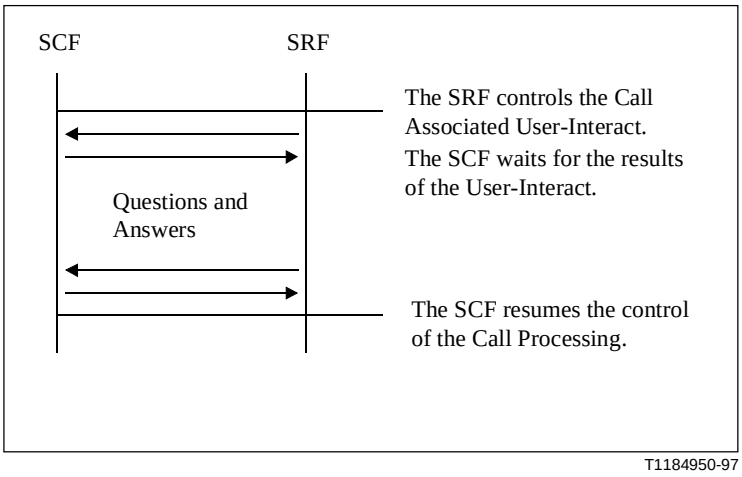


Figure 3-2/Q.1224 – SCF-SRF interaction chart

- *Call-Unrelated Interactio*:

  The establishment of this relationship does not need to be preceded by the establishment of a relationship between the SCF and the SSF when a call-unrelated interaction is concerned. Call-unrelated interaction is based on a set of procedures to exchange information with SCF when SRF is not processing a call. These procedures are generic in that they may be used for many services or operations, typically for management capabilities implementing in the logic. SRF could also use call-unrelated signalling during the processing of a call: this could be done independently of the call processing. Some examples of call-unrelated interaction are provide as follows:

  – the SCF monitoring the availability of resources at the SRF;

  – the SCF requesting control of some SRF resources outside the context of a call, e.g. for automatic deletion of messages stored at the SRF (e.g. based on data/time parameters rather than commands from the user).

3.4.7 SRF-SMF relationship

The relationship between the IN CS-2 SRF and the SMF deals with the management of the service specialized resources. A typical example could be the updating of stored announcements from a centralized management function in a coordinated manner with the logic updates of the service(s) using those announcements. Another case could be a process of collecting information on resources usage or resources availability in the network. The SMF is able to:

– create, deploy and manage User Interaction-scripts on the SRF;

– manage configuration data associated to the User Interaction-scripts on the SRF;

– manage vocal data required by the SRF User Interaction-scripts.

The SRF could also provide for some procedures to allow a service subscriber to directly manage the resources they have been subscribed to, which are located in the SRF; in this case, the managing procedures are offered to a user connected to the SRF within the context of an IN call.

3.4.8 SDF-SDF relationship

This SDF-SDF relationship is used for information flows between two SDFs in the public network. The relationship supports both access to service data, which can be for intranetwork or internetwork purposes.

For IN CS-2, the SDF-SDF relationship supports information flows for services when no call is actually taking place (call-unrelated) as well as during call processing. For the most part, the call-unrelated actions are to support the registration, authentication, encryption and handover procedures for terminal and personal mobility.

For IN CS-2 the SDF-SDF relationship is also one of a limited set of relationships to support internetworking. As such it provides a point of interconnection to the network, effectively hiding the specific structure of the network and providing access security to the network from other public networks.

The SDF-SDF relationship is used for various purposes. Each of the purposes can apply to the intranetwork, or internetwork cases. Specifically, the relationship is used for the following:

a) **Secured Data Acquisition**: between SDFs. In this case a SDF requires data which is not stored within the SDF to complete a request for service data by a SCF. A crucial aspect of this relationship is security necessary to support network isolation and internetworking.

b) **Secured Copying of Service Data**: between SDFs. In this case a SDF copies a portion of service data which is controlled by another SDF. A crucial aspect of this relationship is the keeping of the original security and access information relation to the service data once the data has been copied and ensuring that copies of data are deleted when no longer required.

3.4.9 SCF-SDF relationship

This SCF-SDF relationship is used for information flows between a SCF and a SDF in the public network. The relationship supports access to service data, which can be for intranetwork or internetwork purposes.

For IN CS-2, the SCF-SDF relationship supports information flows for services when no call is actually taking place (call-unrelated) as well as during call processing. For the most part, the call-unrelated actions are to support the registration, authentication, encryption and handover procedures for terminal and personal mobility.

For IN CS-2 the SCF-SDF relationship is also one of a limited set of relationships to support internetworking. As such it provides a point of interconnection to the network, effectively hiding the specific structure of the network and providing access security to the network from other public networks.

The SCF-SDF relationship is used for various purposes. Each of the purposes can apply to the intranetwork, or internetwork cases. Specifically, the relationship is used for the following:

– **Secured Data Acquisition**: from a SDF by a SCF. In this case a SCF requires data to complete its processing of service logic. This relationship is the same as the SCF-SDF relationship as defined in IN CS-1 and uses the same operations. It differs from the IN CS-1 relationship in the provision of additional mechanisms for supporting security necessary to support network isolation and internetworking.

3.4.10 SCF-CUSF relationship

The SCF-CUSF relationship is used for information flows between an SCF and an CUSF in the public network. This relationship provides service or service features which need call-unrelated user interaction. This provides, for example, user location registration, user authentication, supplementary service activation or de-activation. Details of service drivers can be found in Recommendation Q.1221.

3.4.11 CUSF-SSF relationship

It is considered that the CUSF and the SSF may have a relationship, but the relationship is not defined in CS-2, and is not mandatory. This relationship may be used to influence the basic call processing in the SFF/CCF (e.g. activation or de-activation of Call Forwarding) via call-unrelated interaction since the CUSF (with the aid of the Non-IN Feature manager in the SSF) represents the switch-based call-unrelated service processing.

3.4.12 CUSF-CCF relationship

It is considered that the CUSF and the CCF may have a relationship, but the relationship is not defined in CS-2, and is not mandatory. This relationship may be used to manage Call References.

3.4.13 SMF-SCF relationship

This relationship manages the entities/components and data related to the SCF. The entities/components include:

- the Service Logic Execution Manager;
- the Service Logic Selection/Interaction Manager;
- the Service Logic Processing Program Instance;
- the SCF Data Access Manager;
- the Resource Manager;
- the SLP Manager.

This relationship performs:

- *Configuration Management Functions*
  - SLP Configuration:
    - the SMF should be able to introduce new SLPs along with the corresponding service key and revision number, and likewise be able to remove an SLP identified by its service key;
    - when instructed by the SMF to remove an SLP the SCF should prevent the creation of new call instances of the SLP;
    - support any active call instances of the SLP as long as they exist;
    - notify the SMF on removal of the SLP after all the active call instances of the SLP have terminated.
  - Updating the Service Configuration data:
    - within the SCF resides information about the location and capabilities of resources in the network (SSF, SRF) accessible to the SCF. To maintain consistency across SCFs this information should be modified at the SMF, and the modifications broadcast to the SCFs affected. Modifications to the service configuration data would be done via interaction with the SCF Data Access Manager;

    - this interface would be used both when new resources are added or removed and also in real time during fault conditions.
  - Configuration verification:

    The SMF should be able to determine the current configuration of the SCF at any time. This may be facilitated by the use of a version number associated with the SLPs stored in the SCF.

- *Fault Management Functions*
  - Reporting of Alarms and Operations measurements:
    - alarms and logs can be generated within a service instance by the inclusion of specific SIBs in the SLP (e.g. the Log Call Info SIB);
    - additional non-service specific alarms and logs may also be generated under instruction from the SMF (as the result of specific timers and counters);
    - these Alarms and logs can be stored locally until collected by the SMF or be sent immediately to the SMF;
    - the SMF is responsible for collating, filtering and redistributing these alarms and logs.
- *Performance Management Functions*
  - Call Gapping:

    When a SCF invokes call gapping to protect its resources in an overload situation, the SMF is notified. The SMF uses this information to adjust the engineering of the network such that overload occurs less frequently.
  - Updating Service configuration data:

    The SMF will use the update service configuration data facilities to control network configuration in the result of resource failure or congestion within the network.
  - Traffic measurements:

    The following traffic measurements should be available to the SMF from the SCF:
    - number of operations from SSF;
    - number of operations from SSF completed;
    - number of operations from SRF;
    - number of operations from SRF completed.
- *Testing functions*
  - Initiate the testing of newly developed services and collation of test results:
    - when an SMF deploys an SLP to an SCF, a set of test vectors for the SLP is included;
    - the SMF should be able to instruct the SCF to run the test vectors for an SLP;
    - the SCF runs the test vectors for the SLP on request and reports the results back to the SMF.
  - Logic tracing:

    The SCF should provide the functions to enable the tracing of the execution of a Service Logic Program and the results passed back to the SMF, for example: breakpoints and manipulation of data within an SLP.

- *Security Management Functions*
  - Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the SCF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  - Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

    The following features are required:
    - logging of invalid access attempts and generation of alarms;
    - activation and deactivation of appropriate security features in each FE to enable correct interface operation;
    - distribution of security-related information.

3.4.14 SMF-SDF relationship

The entities/components in the SDF that are involved in this relationship include:

- the SDF Data Manager;
- the Functional Entity Access Manager;
- the SDF security manager.

This relationship performs:

- *Configuration Management Functions*
  - Schema Management:

    When a new service is deployed in the network new data schemas associated with the service need to be loaded into the SDF. Likewise if a new version of an existing service is created evolution of existing data to the new schema is required.
  - Updating Service Data:

    The requirements of the SMF to read and write to the SDF are similar to the requirements of the SCF and thus similar facilities should be provided.
  - Updating Operational and Security Data:
    - the SMF is required to manage the setting of the Access control attributes and other operation control attribute associated with the service data;
    - operational requirements for definition and storage of the agreements within SDFs and possibly other FEs must be specified. For the purpose of this subclause, the agreement defines the conditions under which data transfer (between networks) may occur and may be activated.
  - Data Distribution and Replication:

    The contents of the SDF may be physically split across many platforms and replicated for the purposes of reliability and speed. The location of data should be configurable from the SMF. Procedures are required to maintain consistency of the data.

- *Fault Management Functions*
  - Reporting of Alarms, Charge Records and Operations measurements:
    - alarms and logs may be generated under instruction from the SMF (as the result of specific timers and counters);
    - these Alarms and logs can be stored locally until collected by the SMF or be sent immediately to the SMF;
    - the SMF is responsible for collating, filtering and redistributing these alarms and logs.
- *Security Management Functions*
  - Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the SDF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  - Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

    The following features are required:
    - logging of invalid access attempts and generation of alarms;
    - activation and deactivation of appropriate security features in each FE to enable correct interface operation;
    - distribution of security related information.

3.4.15 SMF-SSF/CCF relationship

The entities/components that are involved in this relationship include:

- the Basic Call Manager entity in the CCF;
- the IN-Switching Manager entity in the SSF;
- the Feature Interactions Manager/Call Manager in the SSF.

This relationship performs:

- *Configuration Management Functions*
  - Trigger Table Configuration:

    The SMF should be able to introduce new trigger data along with the corresponding service key and revision number, and likewise be able to remove a trigger data identified by its service key. When instructed by the SMF to remove a trigger data the SSF/CCF should remove it immediately. The SMF should also be able to activate or deactivate existing trigger data within the SSF/CCF.

    Management of trigger data to be set per trigger (note that this represents the SSF/CCF trigger information model) includes:
    - Trigger type:
      - Trigger Detection Point-Request;
      - Trigger Detection Point-Notification.
    - Detection Point in BCSM;

    - Trigger criteria;
    - Routing to SCF;
    - Service key;
    - Version number;
    - Fault handling (to indicate call handling on an error being encountered):
      - terminate call as dialled;
      - deny call;
      - play specified announcement and terminate call;
      - route call to specified alternate destination.
    - Association to SSF/CCF user resource.
  - Service Interaction Functions:

    The SSF should be able to predict at service deployment time some unwanted service interactions. If this occurs, new trigger data should be disallowed and an error handling mechanism invoked with the SMF.
  - Configuration verification:

    The SMF should be able to determine the current configuration of the SSF/CCF at any time. This may be facilitated by having the SMF request the SSF/CCF to provide to the SMF a copy of the trigger information for an SSF/CCF user resource.
- *Fault Management Functions*
  - Reporting of Alarms, Charge Records and Operations measurements:
    - alarms and logs can be generated at the SSF/CCF under instruction of either the SMF or the SCF;
    - these Alarms and logs can be stored locally until collected by the SMF or be sent immediately to the SMF;
    - the SMF is responsible for collating, filtering and redistributing these alarms and logs.
- *Performance Management Functions*

  The following capabilities should be available to the SMF from the SSF/CCF:
  - Call gapping Information (i.e. the number of calls blocked per call gap and the number of calls blocked per call gap type), with the following parameters per call gap:
    - type of call gapping (i.e. manual, SCF overload, destination overload);
    - status of call gap (e.g. active passive);
    - criteria for call gap (e.g. calling region, called region, calling party, called party);
    - gapping duration interval;
    - treatment given to gapped calls (e.g. play announcement, busy tone).
  - Traffic measurements (maybe per service), such as:
    - number of unsuccessful call attempts due to caller abandon, SCF failure or SSF failure;
    - number of unsuccessful calls;
    - number of queries sent to SCF;
    - average waiting time per call.
  - Selective blocking/unblocking of trigger tables for congestion management.

- *Testing Functions*

  See details in Annex D: "IN testing and Fault Management".

- *Security Management Functions*
  - Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the SSF/CCF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  - Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

    The following features are required:
    - logging of invalid access attempts and generation of alarms;
    - activation and deactivation of appropriate security features in each FE to enable correct interface operation;
    - distribution of security-related information.

3.4.16 SMF-SRF relationship

The relationship between the IN CS-2 SRF and the SMF deals with the management of the specialized resources: a typical example could be the updating of stored announcements from a centralized management function in a coordinated manner with the logic updates of the service(s) using those announcements. Another case could be a process of collecting information on resources usage or resources availability in the network. For this reason some generic SRF management functions are required as well as specific management functions for specific types of SRF.

The entities/components that are involved in this relationship include:

- the functional entity access manager;
- the SRF Resource Manager.

This relationship performs:

- *Configuration Management Functions*
  - Version control:

    All forms of SRF whether a voice server or another SRF function should have a common configuration management procedure. Part of this procedure should include the use of a revision number to be used when the SMF introduces new data to the SRF.
  - Configuration verification:

    The SMF should be able to determine the current configuration of the SRF at any time. This may be facilitated by the use of a version number associated with the service specific information stored in the SRF.
  - Update Transactions:

    Regardless of the type of SRF used, maintaining the consistency between the SLPs, trigger points and service keys are such that a common transaction process is required. For instance the SRF data associated with a service should not be deleted until the trigger has been deleted and all existing service instances have finished.

– Prompt and collect user information parameters:

  The SMF manages prompt and collect user information parameters in the SRF as required.

– Announcements Configuration:

  The SMF should be able to introduce new announcements along with the corresponding announcement ID and revision number, and likewise be able to remove that announcement.

- *Fault Management Functions*
  – Reporting of Alarms and Operations measurements:
    - alarms and logs may be generated under instruction from the SMF (as the result of specific timers and counters);
    - these Alarms and logs can be stored locally until collected by the SMF or be sent immediately to the SMF;
    - the SMF is responsible for collating, filtering and redistributing these alarms and logs.
- *Performance Management functions*
  – Performance measurements:

    The following performance measurements should be available:
    - number of times an announcement is used;
    - number of times "prompt and collect user information" is used;
    - number of failures to get user information due to timeout.
- *Security Management Functions*
  – Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the SRF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  – Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

    The following features are required:
    - logging of invalid access attempts and generation of alarms;
    - activation and deactivation of appropriate security features in each FE to enable correct interface operation;
    - distribution of security related information.

3.4.17 SMF-SMAF relationship

There are two kinds of SMAF users, service administrator and service subscriber. Service Administrator consists of both the Service Provider and Network Provider. The relationship manages the entities/components and data related to SMAF, respectively.

The entities/components of SMAF include:

– SMF Access Manager;

– User Interface.

This relationship performs:

- *Fault Management Functions*
  - Reporting of Alarms for Service Administrator:
    - alarms and logs can be generated within a service instance by the inclusion of specific SIBs in the SLP (e.g. the Log Call Info SIB). Additional non-service specific alarms and logs may also be generated under instruction from the SMF (as the result of specific timers and counters);
    - these Alarms and logs can be stored locally until collected by the SMF or be sent immediately to the SMF;
    - the SMF is responsible for collating, filtering and redistributing these alarms and logs.
  - Reporting of Alarms for Service Subscriber:
    - alarm summaries and logs can be generated within a service instance.
- *Testing Functions for Service Administrator*
  - Initiate the testing of newly developed services and collation of test results. The Administrator may request the SMF to perform testing via the SMAF. The tests are performed by the SMF.
- *Performance Management Functions*
  - Performance Measurements:
    - operation measurements/log collection;
    - generating statistics on access traffic;
    - generating statistics on the usage of specific services;
    - generating statistics on the usage of subscriber-specific services;
    - retrieval of service-specific statistic data;
    - retrieval of subscriber-specific statistic data.
  - Performance monitoring:
    - monitoring service/subscriber traffic;
    - monitoring the status of services.
- *Security Management functions*
  - Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the SMAF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  - Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

The following features are required:

- logging of invalid access attempts and generation of alarms;
- activation and deactivation of appropriate security features in each FE to enable correct interface operation;
- distribution of security-related information;
- limiting operation according to system access level.

NOTE – Additional functions may be required from a service subscriber perspective; for example, controlling passwords, controlling access capability, controlling access level.

- *Configuration Management Functions*

  The configuration aspects described in this relationship are involved when the subscriber or service administrator performs modifications through the SMF-SMAF path (e.g. direct customization of a user profile by a service subscriber, or service administrator on behalf of a service subscriber). Modification of service specific or service subscriber specific information can also be accomplished through other relationships, if defined as part of the service definition (e.g. UPT registration can be accomplished through a SSF/CCF-SCF-SDF path).

  – For Service Administrator:

    The detailed operations which need to be available are the following:

    - initiation of subscriber-specific service data;
    - deletion of service-specific service data;
    - deletion of subscriber-specific service data;
    - activation /deactivation of specific services;
    - activation/deactivation of subscriber-specific service;
    - setting collection criteria for subscriber-specific statistic data;
    - downloading subscriber profile to allow customization;
    - limiting operation according to system overload level.

  – For Service Subscriber:

    The detailed operations which need to be available are the following:

    - updating subscriber profile (access information);
    - updating subscriber-specific service data;
    - activation/deactivation of access to specific subscriber service.

3.4.18 SMF-SCEF relationship

The SCEF-SMF interface purpose is to convey the output of the SCEF to the SMF. Once a new service has been installed on the SMF from the SCEF, the task of deploying the service is the role of the SMF and initiated from within the SMF.

This relationship performs:

- *Configuration Management functions*

  – Service Configuration Control:

    - the SCEF is required to manage the storage of multiple revisions of services during the service creation process;

    - the SMF is required to manage multiple versions of services during service deployment and migration. The SMF is responsible for the coordination of the separate revision histories for a service between the SMF and the SCEF;
    - the revision history needs to include SLPs, trigger information and service schemas. The capability needs to exist that allows earlier version of services to be reintroduced, or services to be rolled back.
  - Creation of Service Keys and Service Names:
    - each installed service needs to have a unique name. Each Service Logic Program within a service needs to have a unique identifier, the Service Key. The SMF is responsible for managing the allocation of these service keys to new services;
    - this can be achieved through each SMF having a Service Key range. The SMF can then allocate a Service Key to each new service as it is installed or by passing responsibility of service by service allocation of Service Keys to an SCEF.
  - Service Number Assignment:

    Services that are triggered on dialled digits will need to have a set of numbers assigned to them during service deployment, and to a user during service subscription. This number is assigned to the service from the appropriate number plan by the SMF at deployment of service if the appropriate DP is used by the service.
  - Configuration Verification:

    Not defined in IN CS-2.
- *Testing Functions*

  Not defined in IN CS-2.
- *Generic Management Scripting interface*

  Not defined in IN CS-2.
- *Security Management Functions*
  - Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the SCEF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  - Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

    The following features are required:
    - logging of invalid access attempts and generation of alarms;
    - activation and deactivation of appropriate security features in each FE to enable correct interface operation;
    - distribution of security-related information.

3.4.19 SMF-SMF relationship

This relationship manages the interworking between entities/components and data residing in different networks. This relationship is the TMN X interface as defined in Recommendation M.3020.

The entities/components of SMF include:

– SMAF Access manager;

– SCF/SDF/SRF/SSF-CCF Access Manager;

– CMIP/CMISE Access Manager.

Communicating management systems in separate networks must share a common view or understanding of the interface over which management information is to be exchanged. Some form of context negotiations may be required to establish this common understanding within each entity. In TMN terms, this is referred to as "Shared Management Knowledge" (SMK).

An accepted management principle used in internetworking is that IN FEs SCF and SDF in one provider's network are (only) managed by SMFs within that provider's network. Stated another way, SMFs in one provider's network should not directly control SCFs and SDFs of another provider's network.

Functionally, an SMF of provider A communicates with an SMF of provider B, and an SCF, SDF of provider A communicates with an SCF, SDF of provider A's network requests changes in an SCF, SDF in provider B's network by negotiating and forming an agreement with an SMF in provider's B's network. The SMF in provider B's network then takes the necessary steps to produce the change consistent with provider B's internal policies and procedures.

The SMF-SMF relationship performs:

- *Fault Management Functions*
  - Reporting of Alarms:

    Alarms specific to the SMF-SMF interface connection should be readily available in each domain. Logs may be generated under instruction from the home SMF. These Alarms and logs can be stored locally until collected by the home SMF or be sent immediately to the SMF in the other domain.
- *Testing Functions*

  Not defined in IN CS-2.
- *Performance Management Functions*
  - Performance Measurements:

    Operation measurements/log collection specific to SMF-SMF interface connection.
  - Performance Monitoring:

    Not defined in IN CS-2.
- *Security Management Functions*
  - Interface Functions:
    - The security management functions provide the functionality needed for secure access to the functions of the SMF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity.
    - The selection of active features is determined by the nature of the interaction. It is required that both domains have agreed upon a set of security mechanisms resident in-place before interconnection takes place.

- *Configuration Management Functions*
  - activation/Deactivation of SMF-SMF interface connection;
  - enable/Disable use of the resources required by the SMF-SMF interface connection by the home SMF and the SMF in the other domain, respectively (control of resources resides within each domain).

3.4.20 SMF-CUSF relationship

The entities/components that are involved in this relationship include:

- Basic call-unrelated manager;
- IN-Non-Switching Manager;
- Feature interaction manager/call-unrelated manager.

This relationship performs:

- *Configuration Management Functions*
  - Call Unrelated Trigger Table Configuration:

    The SMF should be able to introduce new trigger data along with the corresponding service key and revision number, and likewise be able to remove a trigger data identified by its service key. When instructed by the SMF to remove a trigger data the CUSF should remove it immediately. The SMF should also be able to activate or deactivate existing trigger data within the CUSF.

    Management of call-unrelated trigger data to be set per trigger (note that this represents the CUSF trigger information model) includes:

    - Trigger type:
      - Trigger Detection Points-Request;
      - Trigger Detection Points-Notification.
    - Detection Point in BCUSM;
    - BCUSM DP Trigger criteria;
    - Routing to SCF;
    - Service key;
    - Version number;
    - Fault handling (to indicate call-unrelated handling on an error being encountered):
      - terminate call-unrelated function;
      - deny call-unrelated function;
      - terminate SCF function;
      - deny SCF function;
      - play specified announcement and terminate call-unrelated function;
      - play specified announcement and terminate SCF function;
      - route SCF function to specified alternate destination;
      - route call-unrelated function to specified alternate destination.
    - Association to CUSF user resource;

- Service Interaction Functions:

  The CUSF should be able to predict at service deployment time some unwanted service interactions. If this occurs new trigger data should be disallowed and an error handling mechanism invoked with the SMF.

- Configuration verification:

  The SMF should be able to determine the current configuration of the CUSF at any time. This may be facilitated by having the SMF request the CUSF to provide the SMF with a copy of the trigger information for a CUSF user resource.

– Updating the Service Configuration data:

  Within the SCF resides information about the location and capabilities of resources in the network (SSF) accessible to the SCF. To maintain consistency across SCFs this information should be modified at the SMF, and modifications broadcast to the SCFs affected. Modifications to the service configuration data would be done via interaction with the SCF Data Access Manager.

- *Fault Management Functions*

  – Reporting of Alarms, Charge Records and Operations measurements:

    Alarms and logs can be generated at the CUSF under instruction of either the SMF or the SCF. These Alarms and logs can be stored locally until collected by the SMF or be sent immediately to the SMF. The SMF is responsible for collating, filtering and redistributing these alarms and logs.

- *Performance Management Functions*

  The following capabilities should be available to the SMF from the CUSF:

  – SCF-related Call gapping Information (i.e. the number of calls blocked per call gap and the number of calls blocked per call gap type) with the following parameters per call gap:
    - type of call gapping (i.e. manual, SCF overload, destination overload);
    - status of call gap (e.g. active passive);
    - criteria for call gap (e.g. calling region, called region, calling party, called party);
    - gapping duration interval;
    - treatment given to gapped calls (e.g. play announcement, busy tone).

  – Traffic measurements (may be per service), such as:
    - number of unsuccessful call-unrelated attempts due to function abandon, SCF failure or SSF failure;
    - number of unsuccessful call-unrelated functions;
    - number of queries sent to SCF;
    - average waiting time per call-unrelated functions.

  – Selective blocking/unblocking of trigger tables for congestion management;

  – Updating Service configuration data:

    The SMF will use the updated service configuration data facilities to control network configuration in the result of resource failure or congestion within the network.

- *Testing Functions*

  See details in Annex D: "IN testing and Fault Management".

- *Security Management Functions*
  - Interface Functions:
    - the security management functions provide the functionality needed for secure access to the functions of the CUSF. This functionality should apply security procedures to ensure data confidentiality, origin authentication and data integrity;
    - the selection of active features is determined by the nature of the interaction.
  - Security Control Functions:

    The provision of security features for network interfaces does not automatically guarantee network integrity, rather they provide the tools which can be used by the IN management functions to control and manage access to the network.

    The following features are required:
    - logging of invalid access attempts and generation of alarms;
    - activation and deactivation of appropriate security features in each FE to enable correct interface operation;
    - distribution of security related information.

3.5 Overview of functional entity call/service logic processing models

IN call/service logic processing encompasses call and connection processing in the SSF/CCF, service logic execution in the SCF, and the use of supporting resources and data in the SRF and SDF, respectively. This subclause describes this IN call/service logic processing in terms of call modelling and modelling of service logic processing.

- Call modelling provides a high-level service and vendor/implementation independent abstraction of IN call and connection processing in the SSF and CCF. This abstraction provides an observable view of SSF/CCF activities and resources to the SCF, enabling the SCF to interact with the SSF in the course of executing a service logic.
- The modelling of service logic processing provides an abstraction of SCF activities and resources needed to support this service logic execution, as well as an abstraction of SRF and SDF activities and resources available to the SCF.

Since this modelling only provides an observable (i.e. external) view of SSF/CCF, SCF, SRF and SDF activities and resources, this modelling does not imply an obligation to vendors to implement functional entities into products as a one-to-one mapping of functional entity model components.

The modelling in this subclause is based on the modelling objectives, assumptions, and architecture described in clause 3/Q.1204, and makes use of the tools identified in the annexes of Q.1204, as applicable to IN CS-2.

4 SSF/CCF model

4.1 General

A model of the SSF/CCF is shown in Figures 4-1 and 4-2. Figure 4-1 shows the SSF/CCF model for a single-ended service logic instance related to a calling or called party. Figure 4-2 shows the SSF/CCF model for separate single-ended service logic instances related to the calling and called parties on the same call. The purpose of this model is to provide a framework for call modelling subjects with respect to the SSF/CCF.

The aspects of the SSF/CCF model briefly described below include the Basic Call Manager (BCM), the IN-switching manager (IN-SM), the Feature Interactions Manager (FIM)/Call Manager (CM), the relationship of the BCM to the IN-SM, the relationship of the BCM and IN-SM to the FIM/CM, and the functional separation provided in the SSF/CCF. Additional details are provided in subsequent subclauses.

a) ***BCM*** – The BCM is not a functional entity. It provides an abstraction of a part of a switch that implements basic call and connection control to establish communication paths for users and to interconnect such communication paths. It detects basic call and connection control events that can lead to the invocation of IN service logic instances or should be reported to active IN service logic instances, and manages CCF/SSF resources required to support basic call and connection control.

   The BCM also implements the BCSM and the DP processing.

   The DP processing is the entity of the BCM that interacts with the FIM/CM as described in the FIM/CM description below.

b) ***IN-SM*** – The entity in the SSF that interacts with the SCF in the course of providing IN service features to users. It provides the SCF with an observable view of SSF/CCF call/connection processing activities, and provides the SCF with access to SSF/CCF capabilities and resources. It also detects IN call/connection processing events that should be reported to active IN service logic instances, and manages SSF resources required to support IN service logic instances. The IN-SM interacts with the FIM/CM as described below.

c) ***FIM/CM*** – The entity in the SSF that provides mechanisms to support multiple concurrent instances of IN service logic instances and non-IN service logic instances on a single call. In particular, the FIM/CM can prevent multiple instances of IN and non-IN service logic instances from being invoked. The ability of the FIM/CM to arbitrate between multiple instances of IN and non-IN service logic instances is outside the scope of IN CS-2. The FIM/CM integrates these interactions mechanisms with the BCM and IN-SM to provide the SSF with a unified view of call/service processing internal to the SSF for a single call.

   The management of IN-IN service interaction should be done based on ServiceCompatibilityIDs, if ServiceCompatibilityIDs are supported in the network. The management of IN service interaction with network-based features should be done based on ServiceInteractionIndicators. Global management of feature interaction also calls for conveying the ServiceCompatibilityIDs of all IN services invoked during a call and the ServiceInteractionIndicators via basic network signalling to the originating and to the terminating instances.

   In IN CS-2, the FIM/CM supports:

   – Prevention of non-IN service logic instance invocation according to instructions from an IN service logic.
   – Prevention of multiple IN service logic invocations according to ServiceCompatibilityIDs which may be assigned to an IN service by the basic network operator via administration. The FIM performs a generic compatibility check procedure using an administrable exclusion matrix for ServiceCompatibilityIDs.

d) ***BCM relationship to IN-SM*** – The relationship that encompasses the interaction between the BCM and the IN-SM, through the FIM/CM. The information flow related to this interaction is not externally visible and is not standardized for IN CS-2. However, an understanding of this subject is required to identify how basic call and connection processing and IN call/connection processing may interact.

e) ***BCM and IN-SM relationships to FIM/CM*** – The relationships that encompass the interaction between the BCM and the FIM/CM, and the IN-SM and the FIM/CM. The information flows related to these interactions are not externally visible and are not standardized for IN CS-2. However, an understanding of this subject is required in order to unify the BCM, IN-SM and FIM/CM.

f) ***Functional separation in the SSF/CCF*** (Figure 4-2) – The functional separation of processes and resources in the SSF/CCF that provides a means of handling service logic instance interactions for IN CS-2. This functional separation serves to isolate single-ended service logic instances related to the calling party from single-ended service logic instances related to the called party for the same call. Within the scope of IN CS-2, there is no functionality in the SSF for handling service feature interactions between the separate SSF calling party processes and SSF called party processes.

Other aspects shown in Figure 4-1 are not addressed for IN CS-2, but are assumed to exist.

4.2 Basic Call Manager (BCM)

A brief description of the BCM is provided in 4.2.1. The particular BCM subjects addressed below include the Basic Call State Model (BCSM), basic call and connection events that can lead to the invocation of IN service logic instances, and basic call and connection events that should be reported to active IN service logic instances. A high-level description of these subjects is provided below.

4.2.1 BCSM

In this Recommendation, the BCSM provides a high-level model description of CCF activities required to establish and maintain communication paths for users. As such, it identifies a set of basic call and connection activities in a CCF and shows how these activities are joined together to process a basic call and connection (i.e. establish and maintain a communication path for a user).

Many aspects of the BCSM are not externally visible to IN service logic instances. However, aspects of the BCSM that are reflected upward to the IN-FM and FIM/CM are visible to IN service logic instances. Only these aspects of the BCSM will be the subject of standardization. As such, the BCSM is primarily an explanatory tool for providing a representation of CCF activities that can be analysed to determine which aspects of the BCSM will be visible to IN service logic instances, if any, and what level of abstraction and granularity is appropriate for this visibility.

The BCSM identifies points in basic call and connection processing when IN service logic instances are permitted to interact with basic call and connection control capabilities. In particular, it provides a framework for describing basic call and connection events that can lead to the invocation of IN service logic instances or should be reported to active IN service logic instances, for describing those points in call and connection processing at which these events are detected, and for describing those points in call and connection processing when the transfer of control can occur.

Figure 4-3 shows the components that have been identified to describe a BCSM, to include: Points in Calls (PICs), Detection Points (DPs), BCSM transitions, and events. PICs identify CCF activities associated with one or more basic call/connection states of interest to IN service logic instances. DPs indicate states in basic call and connection processing at which transfer of control from non-IN to IN service logic can occur. Other states may exist (e.g. within PICs) where call processing is suspended but IN service logic cannot be invoked. BCSM transitions indicate the normal flow of basic call/connection processing from one PIC to another. Entry events cause BCSM transitions into PICs. Exit events represent the result of PIC processing.

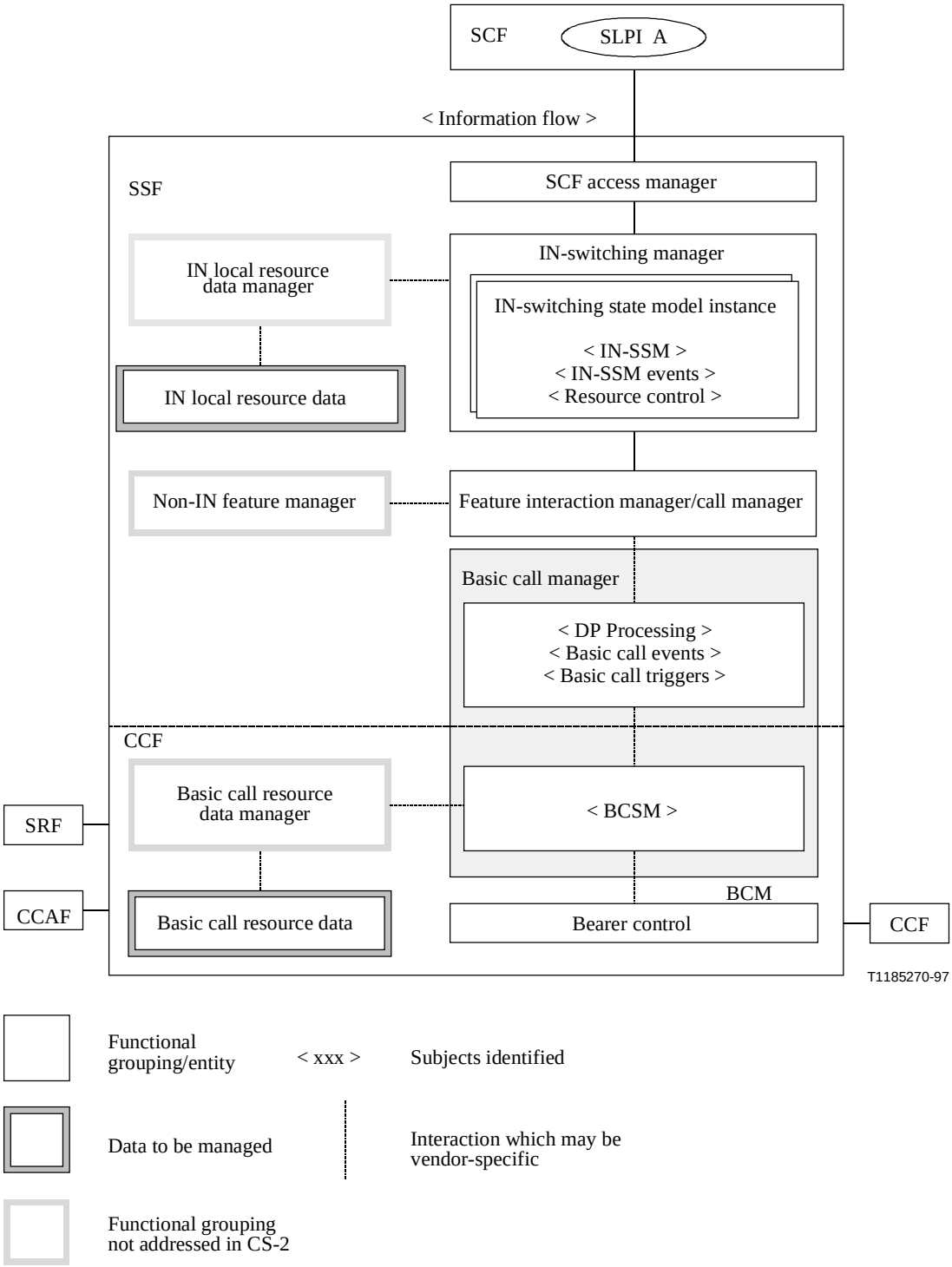


Figure 4-1/Q.1224 – SSF/CCF model – Single-ended SLPI related to calling or called party

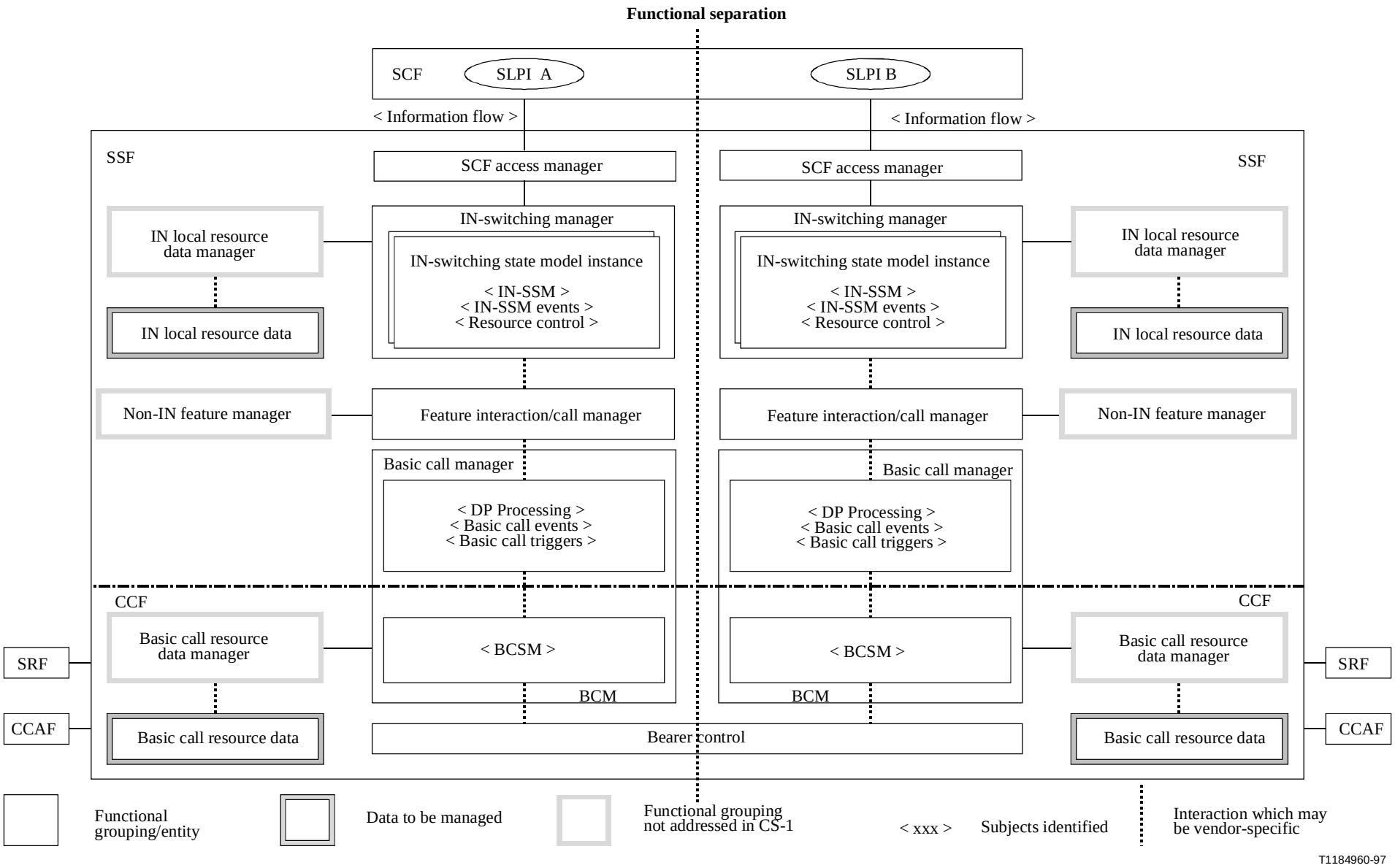


Figure 4-2/Q.1224 – SSF/CCF model – Separated single-ended SLPIs related to calling and called parties

The BCSM for IN CS-2 models existing switch processing of basic two-party calls, and reflects the functional separation between the originating and terminating portions of calls reflected in Figure 4-1. In addition, though CCAF functionality is not explicitly modelled in the BCSM, a mapping is required between access signalling events and BCSM events, for each access arrangement supported by IN CS-2.

Since the BCSM is generic, it may describe events that do not apply to certain access arrangements. It is important to understand and describe how each access arrangement applies to the BCSM.

The BCSM description which is provided hereafter only describes the basic transitions which occur when processing basic two-party calls. The PIC entry event is normally the Resume instruction received from the SSF after the processing of a DP: "basic" transition DP to PIC. PIC exit events are the result of the processing of an event received from the CCAF: "basic" transitions PIC to DP/PIC. The transitions which may occur due to an SCF instruction ("extended" transitions from DP to DP/PIC) and which may change the basic evolution of the call are not described here. Tables 4-1 and 4-2 in 4.2.3 describe the complete set of possible BCSM transitions.

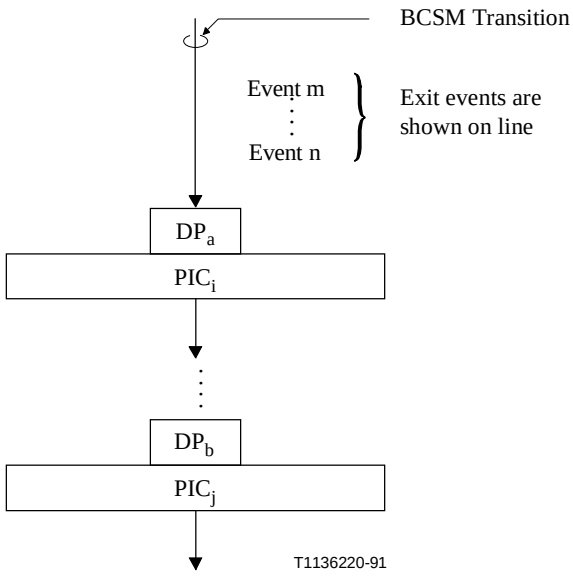


Figure 4-3/Q.1224 – BCSM components

4.2.2 CS-2 BCSM description

The BCSM for IN CS-2 described in this subclause is based on the overall BCSM in Annex A/Q.1204 and Recommendation Q.1214, refined as applicable to IN CS-2.

It reflects the functional separation between the originating and terminating portions of calls as illustrated in Figures 4-4 and 4-5. These figures show an originating half BCSM and a terminating half BCSM, each of which is managed by a functionally separate BCM in the SSF/CCF. The description is a starting point to identify the aspects of the BCSM that are visible to IN service logic instances, and the nature of the information flows between the SSF/CCF and SCF (see clause 12).

In the following descriptions, the PICs are related at a high level to Q.931 ISDN call states. This is not intended to be a detailed formal definition of the relation between the PICs and Q.931 ISDN call states, but is intended as a point of reference to use in understanding the PICs. In particular, there are a number of possible ways in which the Q.931 call states may be traversed in certain situations which are not considered below.

To enable independence between services offered during one call session when the PICs may be traversed several times, it is necessary – at each PIC – to maintain available a specific set of data until the calling (controlling) user releases and to ensure that software resources are returned to a coherent status when call processing passes through the PICs.

For each PIC, an initial list of BCSM information that must be maintained, if available, is given. Information that is available at all PICs is given at the beginning of the O- and T-BCSM descriptions.

The information that is sent to the SCF at a given trigger detection point is a subset of the information described here. Other information may be available at a given PIC that is not used by processing at the PIC or is only used by underlying call processing.

In order to maintain uniqueness of DP names between the originating and terminating half BCSMs, "O" and "T" is prefixed to certain originating and terminating DP names, respectively.

For ease of reference, the DPs associated with the BCSM transition implied by each entry and exit event for each PIC are listed along with the PIC descriptions.

4.2.2.1 Originating BCSM for IN CS-2

The originating half of the BCSM corresponds to that portion of the BCSM associated with the originating party (see Figure 4-4).

The following information is available at all PICs in the O-BCSM:

– Service Address Information: see Recommendation Q.1290.
– Calling Party Category: see Q.762 Calling Party's Category signalling information.
– SRF/SSF capabilities: see Recommendation Q.1290. Used to decide if an assist of hand-off procedure is to be used.
– Call Gapping Encountered: see Recommendation Q.1290.
– Terminal Type: see Recommendation Q.1290. The SCF uses this to determine the most appropriate form of user-interaction to use (e.g. in-band announcements). This information is only available at originating or terminating local exchanges.
– Location Number: see Q.762 Location Number signalling information. Used if the calling party is a mobile subscriber.
– ISDN Access Related Information: see Q.762 Access Transport Parameter.
– Original Called Party ID: see Q.762 Original Called Party Number element.
– Redirecting Party ID: see Q.762 Redirecting Party ID element.
– Redirection Information: see Q.762 Redirecting Information element.
– Additional Calling Party Number: see Q.762 Generic Number element.
– Forward GVNS: see Q.762 Forward GVNS element.
– Service Compatibility ID: see Q.762 Service Compatibility ID element. This information is available at the SSF/CCF.

The description for each of the PICs in the originating half of the BCSM are described below:

NOTE – See BCSM Indications of 4.2.4 for more information concerning PICs.

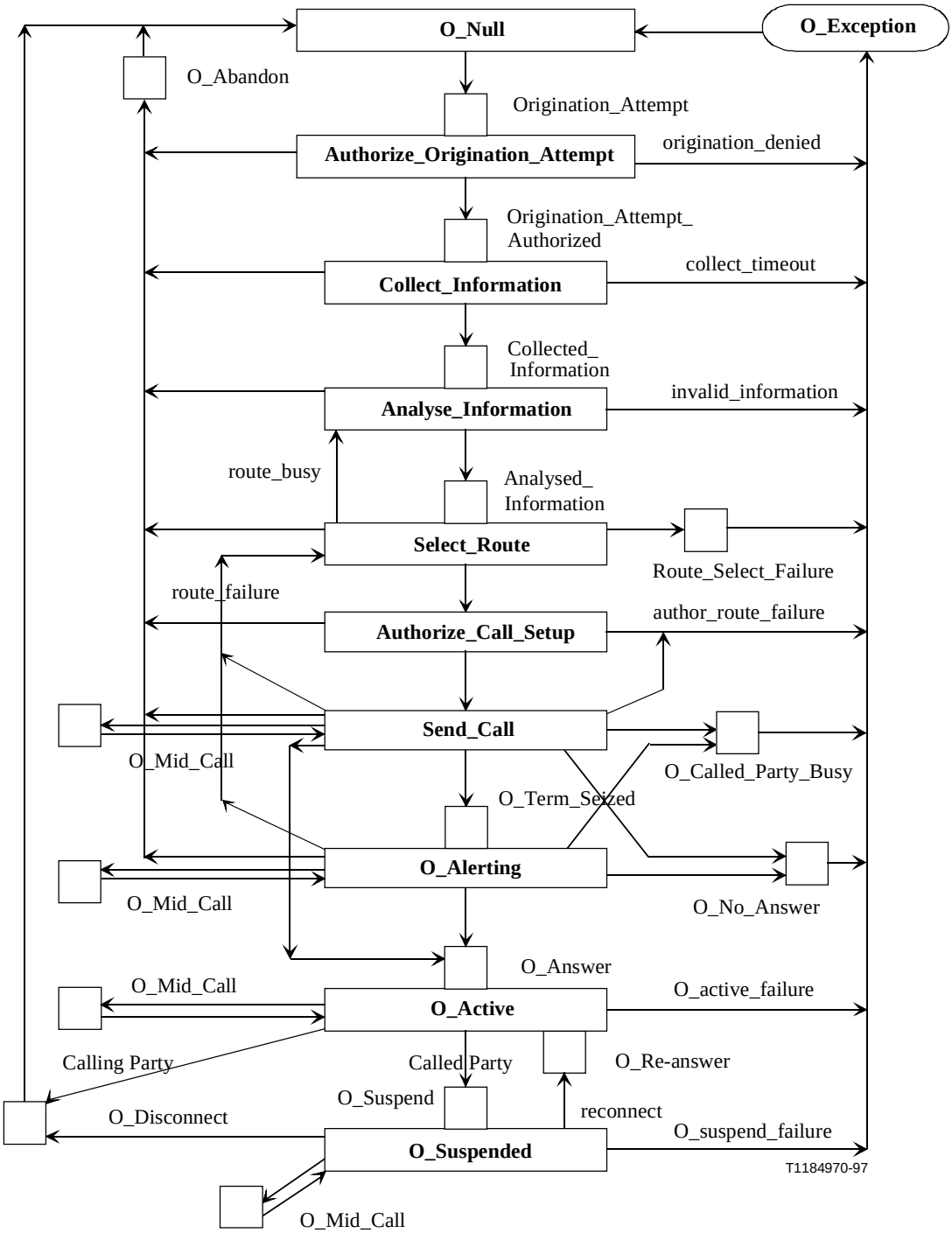


Figure 4-4/Q.1224 – Originating BCSM for CS-2

4.2.2.1.1 O_Null

Entry event: Disconnect and clearing of a previous call (DPs: O_Disconnect and O_Abandon), or default handling of exceptions by SSF/CCF completed.

Functions: Interface (line/trunk) is idled (no call exists, no call reference exists, etc.) Supervision is being provided.

Information available: After detecting the Origination Attempt event, it is assumed that the SSF/CCF has the following information available associated with the originating call portion, with

restrictions as noted. If the SSF/CCF determines that the origination is denied, the cause of the failed authorization is also known.

– Bearer Capability: see Q.762 User Service Information and Q.931 Bearer Capability information element.

– Calling Party Number: see Q.762 Calling Party Number signalling information. This information is available at the SSF/CCF for a non-ISDN line and may be available for SS7 trunks, but is not available from trunks supported by conventional signalling or private-facility trunks. For a DSS 1 interface, this is determined by the information provided in the SETUP message or by the default number assigned to the caller (see ISDN SETUP information below).

– SRF Available: see Recommendation Q.1290.

– Service Profile Identifier (SPID): see Annex A/Q.932. This information may be available at the SSF/CCF if the calling party is served by a BRI interface on this SSF/CCF.

– Called Party Number: see Q.762 Called Party Number signalling information. Used to identify the called party in the forward direction. Available only for trunks or ISDN lines.

– Charge Number: see Recommendation Q.1290. This information is available at the SSF/CCF for a non-ISDN line or ISDN interface served by the SSF/CCF and may be available for trunks supported by SS7, but is not available for trunks supported by conventional signalling or private-facility trunks.

– Class of Service: see Recommendation Q.1290.

– Calling Party Business Group ID (BGID): see Q.1290 Business Group ID. This information is available for a non-ISDN line, ISDN interface, private-facility trunk group, or possibly an SS7 trunk when the caller is a member of a Business Group.

– Calling Facility Group: see Recommendation Q.1290. Available on conventional or SS7 trunks.

– Calling Facility Group Member: see Recommendation Q.1290. Available on conventional or SS7 trunks.

– Travelling Class Mark: see Recommendation Q.1290.

– Feature Code – see Q.762 Feature Code Signalling Information where this parameter is defined for national use only. Available, if used, for a party served by an ISDN interface using *en bloc* sending or for an SS7 trunk.

– Access Code – see Recommendation Q.1290. Available, if used, for a party served by an ISDN interface using *en bloc* sending.

– Operator Services Information – see Recommendation Q.1290. This information element is not included in a SETUP message containing the keypad information element.

– ISDN SETUP feature-related information – see Recommendation Q.931. The SSF/CCF receives a SETUP message from a DSS 1 interface and this SETUP message can contain the following information:

  - Bearer Capability: see Q.931 Bearer Capability information element.
  - Progress Indicator: see Q.931 Progress Indicator information element.
  - Keypad Facility: see Q.931 Keypad Facility information element. This information element is not expected in a SETUP message also containing the called party number, called party number subaddress, transit network selection, or Operator Services Information elements.
  - Feature activation: see Q.932 Feature Activation information element.

- Calling party number: see Q.931 Calling Party Number information element.
- Called party number: see Q.931 Called Party Number information element.

  The called party number information element is sent when *en bloc* sending is used and the keypad information element is not present. When the type of number and numbering plan identification field within the called party number information element is set to "unknown," the SSF/CCF treats the string as if it has been received within a keypad information element. In this case, it is not expected to be sent with the transit network selection or Operator Services Information elements.
- Calling party subaddress and called party subaddress: see Q.931 Calling Party Subaddress and Called Party Subaddress information elements.
- Transit Network Selection: see Q.931 Transit Network Selection information element. This information element is included in a SETUP message containing other information elements than the Keypad information.
- Additional Calling Party Number: see Q.931 Calling Party Number element.
- Facility Information: see Q.931 Facility element. This element may identify USI Information or Facility Information.
- Other information, as defined by Recommendation Q.932, Generic Procedures for the Control of ISDN Supplementary Services, can be included. Some of this information may be of interest to the SCF.

– ISDN User Part IAM feature related information. The IAM can contain the following information (see Recommendations Q.762/Q.763):
  - Nature of connection indicators: see Q.763 Nature of Connection Indicators parameter.
  - Forward call indicators: see Q.763 Forward Call Indicators parameter. The caller's access is identified as ISDN or non-ISDN, and an indication is given of whether an end-to-end SS7 supported connection is required.
  - User service information: see Recommendation Q.762. User Service Information parameter. For the purposes of IN CS-2, this parameter identifies the call as circuit-mode/speech, circuit-mode/3.1 kHz audio, circuit-mode/unrestricted digital information (64 kbit/s), or circuit-mode/restricted digital information.
  - Called party number: see Q.762 Called Party Number parameter.
  - Calling party number: see Q.762 Called Party Number Parameter.
  - Generic Number: see Q.762 Generic Number Parameter. More than one generic number parameter may be present within a given IAM.
  - Generic name: see Recommendation Q.1290.
  - Charge number: see Recommendation Q.1290.
  - Transit network selection: see Q.763 Transit Network Selection parameter. This parameter, if present, identifies the Carrier Identification Code and the Circuit Code.
  - Carrier selection: see Recommendation Q.1290.
  - Feature Code: see Q.762 Feature Code Signalling Information where this parameter is defined for national use only.
  - Generic digits: see Recommendation Q.762. May contain a travelling class mark (network operator specific).
  - Other parameters may be included in the IAM. These parameters may be included because of features provided by other switches in the connection (e.g. information relating to the call being forwarded).

– Any information relating to switch-based features that have already been invoked for the call will also be available.

Exit event

– Indication of desire to place outgoing call (e.g. off-hook, Q.931 setup message, ISDN-UP IAM message). (DP: Origination_Attempt).

– The following exception exit events are applicable to the O_Null PIC. For this PIC, if the call encounters one of these exceptions during O_Null PIC processing, the exception event is not visible because there is no corresponding DP.

  • The O_Abandon occurs when the calling party disconnects. For example, this event can result from one of the following:

    – the SSF/CCF receives an on-hook indication from a caller served by a non-ISDN line, following switchhook flash timing;

    – the SSF/CCF receives a call clearing message from a caller served by an ISDN interface;

    – the SSF/CCF receives a disconnect indication from a conventional trunk or private facility trunk;

    – SSF receives a Release Message from an SS7 trunk.

Corresponding Q.931 call state: O_Null

4.2.2.1.2 Authorize_Origination_Attempt

Entry event: An indication is available that the originating terminal needs to be authorized. (DP: Origination_Attempt).

Functions:

– The originating terminal rights should be checked using the calling party's identity and service profile. The authority /ability of the party to place the call with given properties (e.g. bearer capability, line restrictions) is verified. The types of authorization to be performed may vary for different types of originating resources (e.g. for lines vs. trunks).

– Other features which might be required during this PIC are not described in the IN CS-2 BCSM.

Information available: After detecting the Origination_Attempt_Authorized event, it is assumed that the SSP has the same information available associated with the originating call portion as it did after detecting the Origination_Attempt event in the O_Null PIC.

If the SSP determines that the origination is denied, the cause of the failed authorization is also known.

Exit event:

– An indication is received that the authorization is successful. The O_BCSM moves to the Collect_Information PIC. (DP: Origination_Attempt_Authorized).

– A disconnection indication is received from the originating party. (DP: O_Abandon).

– An indication is received that the call origination is denied. The O_BCSM moves to the O_Exception PIC.

4.2.2.1.3 Collect_Information

Entry event: Authority/ability to place outgoing call verified. (DP: Origination_Attempt_Authorized).

Functions:

– Initial information package/dialling string (e.g. service codes, prefixes, dialled address digits) being collected from originating party. Information being examined according to dialling plan to determine end of collection. No further action may be required if an *en bloc* signalling method is in use (e.g. an ISDN user using *en bloc* signalling, an incoming SS7 trunk).

– The SSF/CCF shall be able to support subsequent digit collection according to trigger criteria assigned before sending the query. For example, if a feature code (e.g. *64) is entered, the SSF/CCF may:

  • collect digits according to the normal dialling plan; or

  • collect a variable number of digits.

Information available: After the SSF/CCF determines that information collection is complete, it is assumed that the SSF/CCF has the following information available associated with the originating call portion:

– Charge Number, Calling Party Number, Calling Party BGID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, other feature-related information, Additional Calling Party Number, Facility Information. This information is available for each access type under the conditions defined in the O_Null PIC.

– Collected Information: As described below.

From a non-ISDN line or DSS 1 interface, the collected information consists of one or more of the following:

– Access Codes within a Customized Dialling Plan (CDP): see Recommendation Q.1290.

  The Customized Dialling Plan (CDP) in force may specify that after a given access code is dialled, more digits are to be collected according to the "normal dialling plan," i.e. the dialling plan in force. In this case, Access Code and Collected Address Information are known. If the CDP in force specifies that after a given access code is dialled, a variable number of digits are to be collected, then Access Code and Collected Digits are known.

– Feature Code: see Q.762 Feature Code Signalling Information where this parameter is defined for national use only.

  If the numbering plan in force specifies that after a given feature code is dialled, more digits are to be collected according to the "normal dialling plan", then Feature Code and Collected Address Information are known. If the dialling plan in force specifies that after a given feature code is dialled, a variable number of digits are to be collected, then Feature Code and Collected Digits are known. The service associated with the feature code is dependent upon the users service profile.

– Facility Code: see Recommendation Q.1290. This information may be provided if and when facility selective service signalling is supported.

– Feature Activation: see Recommendation Q.932 Feature Activation information element. If the CDP in force specifies that after a given feature activator is received, more digits are to be collected according to the numbering plan, then Feature Activation Indicator and Collected Address Information are known. If the CDP in force specifies that after a given

feature activator is received, a variable number of digits are to be collected, then Feature Activation Indicator and Collected Digits are known.

– Prefix: see Recommendation Q.1290.

– Carrier Access Code/Carrier Identification Code: see Recommendation Q.1290. The caller may dial a Carrier Access Code (CAC) (e.g. a 10XXX or 101XXXX for use on this call). When the caller is served by an ISDN interface, a Carrier Identification Code, i.e. XXX or XXXX, may be received by the SSF/CCF within the transit network selection information element of the ISDN SETUP message.

– Collected Address Information: see Recommendation Q.1290. Available as per the numbering plan.

– Numbering Plan Indicator: see Q.762 Numbering Plan Indicator signalling information.

– Collected Digits: see Recommendation Q.1290. The numbering plan in force may specify that after a given Feature Activation, Feature Code, or Access Code within a CDP is dialled, a variable number of digits are to be collected using normal inter-digit timing. In this case, these collected digits are also known at this time.

From a conventional trunk interface, the Collected Information consists of one or more of the following:

– Charge Number: see Recommendation Q.1290. This is only known from a conventional trunk when Multi-Frequency signalling is used on the originating trunk (network operator specific). In this case, the charge number is provided in the second stage of overlap out pulsing.

– Collected Address Information: as defined above for non-ISDN line or DSS 1 interface.

– Carrier Identification Code: see Recommendation Q.1290. This is known if MF signalling is used on the originating trunk (network operator specific).

– Numbering Plan Indicator: see Q.762 Numbering Plan Indicator signalling information. The address received is expected to conform to Recommendation E.164.

– Prefix: as defined above for non-ISDN line or DSS 1 interface.

– Carrier Selection: see Recommendation Q.1290. This information is only provided when MF signalling is used on the originating trunk (network operator specific).

– Originating Line Information: see Recommendation Q.1290. This information is only known when MF signalling is used on the originating trunk (network operator specific). In this case, the Originating Line Information is sent during the second stage of overlap out pulsing.

From an SS7 trunk interface, the Collected Information consists of the information provided in the ISDN User Part called party number and transit network selection parameters, and possibly a Travelling Class Mark and other feature-related information as described above for contents of the ISDN User Part IAM feature related information.

From a private-facility trunk, the collected information consists of one or more of the following:

– Access Code within a CDP: as defined above for a non-ISDN line;

– Feature Code: as defined above for a non-ISDN line;

– Facility Code: as defined above for a non-ISDN line;

– Collected Address Information: as defined above for a non-ISDN line;

– Numbering Plan Indicator – as defined above for a non-ISDN line;

– Prefix: as defined above for a non-ISDN line;

– Carrier Access Code: as defined above for a non-ISDN line;

– Travelling Class Mark: if provided in the generic digits parameter of the IAM (see Q.1290);

– Facility Restriction Level: see Recommendation Q.1290.

Exit events:

– Availability of complete initial information package/dialling string from originating party. (This event may have already occurred in the case of *en bloc* signalling, in which case the waiting duration in this PIC is zero.). (DP: Collected_Info).

– The following exception exit events are applicable to this PIC: CollectTimeout, CollectInfoFailure, InvalidInformation and O_Abandon.

  - The CollectTimeout event is detected when enough information to process the call was not received by the SSF/CCF before a normal interdigit timer expires. For an SS7 trunk, this event corresponds to the IAM not containing the information necessary to process the call. In this case there may be no timing involved (timing may be involved for ISUP overlap sending).
  - The CollectInfoFailure event is detected when the SSF/CCF is unable to perform the information collection due to a lack of switch resources (e.g. no digit receivers are available).
  - The Invalidinformation event occurs when the information received from the caller is not valid, for instance the information received violates the dialling plan in force.
  - O_Abandon event, as described in the O_Null PIC. In this case, the event is visible because there is a corresponding DP.

Comment: Some digit analysis is required to determine the end of dialling. However, it is assumed that this analysis may be modelled as separable from the rest of digit analysis, which occurs in the Analyse_Information PIC. There is no intention to specify an implementation. However, a switch should externally present the separable view described for closed numbering plans. (Note 1)

In the case of ISDN *en bloc* sending, the receipt of a SETUP message detected at the Origination_Attempt_Authorized DP causes the BCSM to pass through the Collect_Information PIC to the Collected_Information DP, without further processing in the Authorized_Origination_Attempt PIC. Note that the BCSM transitions to Collected_Information DP when the initial information package/dialling string is received from the calling party – this occurs when enough information is received to proceed with call processing (e.g. as in the case of ISDN overlap sending of MF out pulsing). Specifically, for the digit-by-digit collection case, if the Collected_Information DP is armed as a Trigger Detection Point-Request (TDP-R), the SSF sends an initialling DP request (i.e. Initial DP or CollectedInformation information flow) to the SCF when enough is received to determine if the TDP criteria is met. It suspends BCSM processing but will collect further digits. It is network operator specific to determine when complete information is available. (Note 2)

NOTE 1 – This separable view is provided by supporting distinct DPs. The Collected_Information DP is used after digit collection and the Analysed_Information DP is used after the rest of the digit analysis.

NOTE 2 – In some networks, it may be not possible for the CCF/SSF to determine when the called number information is complete. Therefore, TDP criteria for Collected_Information DP may be met in such networks before the called number information is complete.

Corresponding Q.931 call state: 1. Call Initiated and (optionally) 2. Overlap Sending.

4.2.2.1.4 Analyse_Information

Entry event: Availability of complete initial information package/dialling string from originating party (DP: collected_Information) or route busy event reported from the Select Route PIC.

Function: Information being analysed and/or translated according to dialling plan to determine routing address and call type (e.g. local exchange call, transit exchange call, international exchange call).

In the case that the SSF had received a list of CDPN in a Connect or AnalyseInformation operation, the SSF/CCF shall use the next CDPN in the list.

The following provides details as to when the next CDPN/Route should be tried under each condition.

Parameters Received:	DestinationRoutingAddress
Result:	Busy or No Answer
Order:	CDPN1

Since no forwarding condition is specified, only the first CDPN is tried.

Parameters Received:	DestinationRoutingAddress, ForwardingCondition
Result:	Busy/No Answer
Order:	CDPN1
	CDPN2
	CDPN3

So long as the forwarding condition is met, the next CDPN is tried.

Parameters Received:	RouteList, DestinationRoutingAddress
Result:	No route out of local switch (busy)
Order:	Route1, CDPN1
	Route2, CDPN1
	Route3, CDPN1

No other CDPNs need be tried since there are no routes out of the switch.

Parameters Received:	RouteList, DestinationRoutingAddress
Result:	Route busy is found to be at a switch other than local switch
Order:	Route1, CDPN1

No other routes need be tried since the local route is successful, but a busy condition was encountered elsewhere. No other CDPNs are tried since forwarding condition does not apply.

Parameters Received:	RouteList, DestinationRoutingAddress
Result:	No answer received
Order:	Route1, CDPN1

No other routes need be tried since the route is successful, but a no answer condition was encountered.

No other CDPNs are tried since forwarding condition does not apply.

Only after the appropriate CDPNs/Routes have been tried will the call move onto the O_Called_Party_Busy DP, O_No_Answer DP, or Route_Select_Failure DP as appropriate.

One of the results of processing in this PIC is determination of routing address:

i) called party number only (called party number is served by the SSF);

ii) called party number and route index, where the route index is a pointer to a trunk group to route an outgoing call attempt on (called party number is served by another SSF);

iii) called party number and route index, where the route index is a pointer to a list of trunk groups to route an outgoing call attempt on (called party number is served by another SSF).

Information available: After the SSF/CCF determines the information has been analysed, it is assumed that the SSF/CCF has the following information available associated with the originating call portion:

– Charge Number, Calling Party Number, Additional Calling Party Number, Calling Party BGID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, Facility Information and other feature-related information. This information is available for each access type under the conditions defined in the O_Null PIC.

– Analysis Results (of the Collected Information): as described below.

From a non-ISDN line or DSS 1 interface, this consists of one or more of the following:

– Called Party Number: as per dialling plan.

– Numbering Plan Indicator: see Q.762 Numbering Plan Indicator signalling information.

– Type Of Call : see Recommendation Q.1290.

– Carrier: see Recommendation Q.1290.

– Carrier Identification Code: see Recommendation Q.1290. Available for Internetwork carrier calls.

– Carrier Selection: see Recommendation Q.1290. Available for Inter Serving Area ID carrier calls. See Recommendation Q.1218.

– Route List: see Recommendation Q.1290.

– Collected Information: Access Code within a CDP, Feature Code, Feature Activation, Prefix, Carrier Access Code/Carrier Identification Code, Collected Address Information/Digits -as described under the Collect_Information PIC.

From a conventional or SS7 trunk interface, this consists of one or more of the following:

– Charge Number: as defined in the O_Null PIC (for an SS7 trunk) or Collected_Information PIC (for a conventional trunk when EAMF signalling is used).

– Called Party Number and Numbering Plan Indicator (as defined above for non-ISDN line or DSS 1 interface).

– Carrier Identification: available for Inter Serving Area ID carrier calls. See Recommendation Q.1218.

– Carrier Selection: see Recommendation Q.1290. Available for Inter Serving Area ID carrier calls. See Recommendation Q.1218.

– Originating Line Information: see Recommendation Q.1290. Available for Inter Serving Area ID carrier calls. See Recommendation Q.1218.

– Route Index: see Q.1290. Available if this call does not terminate on this SSF/CCF.

– Collected Information: Collected Address Information, Prefix, Carrier Identification Code, Feature Code, Facility Code: see description under the Collect_Information PIC.

From a private-facility trunk, this consists of one or more of the following, depending on the type of private-facility trunk:

– Called Party Number and Numbering Plan Indicator (as defined above).

– Type Of Call (as defined above).

– Carrier: see Recommendation Q.1290. Private network/facility, intra-serving area, or a specific Inter Serving Area ID or international). See Q.1218.

– Carrier Identification Code: see Recommendation Q.1290. Available for internetwork carrier calls.

– Carrier Selection: see Recommendation Q.1290. Available for Inter Serving Area ID carrier calls. See Recommendation Q.1218.

– Travelling Class Mark: see Recommendation Q.1290. Available if received on the facility.

– Route List: see Recommendation Q.1290.

– Facility Restriction Level: as described under the Collected_Information PIC.

– Collected Information: Collected Address Information/Digits, Access Code within a Customer Dialling Plan, Feature Code, Carrier Access Code, Prefix – see description under the Collected_Information PIC.

Exit events:

– Availability of routing address and nature of address. (DP: Analysed_Information.)

– The following exception exit events are applicable to this PIC: InvalidInformation and O_Abandon:

  • The InvalidInformation event (e.g. wrong number).

  • The O_Abandon event, as described in the O_Null PIC. In this case, the event is visible because there is a corresponding DP.

Comments: Note that routing address does not necessarily mean that the final physical route has been determined (e.g. route list has not been searched, hunt groups have not yet been searched, directory number has not yet been translated to physical port address), though this may be the case (e.g. when routing to a specific private facility).

Corresponding Q.931 call state: Not applicable.

4.2.2.1.5 Select Route

Entry events: Availability of routing address and call type. (DP: Analysed_Information) or route busy event reported from the Send_Call or O_Alerting PICs.

Functions:

– Routing address and call type are being interpreted. The next route is being selected. This may involve sequentially searching a route list, translating a directory number into physical port address, etc. The individual destination resource out of a resource group (e.g. a multi-line hunt group, a trunk group) is not selected. In some cases (e.g. an analogue line interface), a single resource (not a group) is selected.

– The individual destination resource of a resource group (e.g. a specific line if a multiline hunt group) is not selected.

– When the entry event is the route failure event from the Send_Call PIC (see below), the SSF/CCF must first check the Route Failure Condition 1, Route Failure Condition 2, or

Route Failure Condition 3 as defined under the Send_Call PIC exit events. If these conditions are true, then the call shall proceed to the Analyse_Information PIC in order to use the next CDPN.

If these three conditions are not met, then depending on the location in the network where the route is busy, the action is one of the following:

– If the trunk group selected for the call is busy at this switch, the SSF/CCF attempts to route the call on the next trunk group that has been specified for the call (when a route list is being searched or alternate routes are specified by the SCF). Call processing moves to the Analyse_Information PIC when one of two conditions occurs: all private-facility trunk groups have been tried and routing over a public facility is allowed, or routing to a particular intra or internetwork carrier has been tried and an alternate carrier is allowed.

– If all of the trunk groups (private and public) have been tried and no route is available, the Route_Select_Failure event is detected.

– If route busy is detected at another switch, an indication of this condition may be received via SS7 signalling. In this case, a Route_Select_Failure event is detected.

Information available: After the SSF/CCF determines the route that has been selected, it is assumed the SSF/CCF has the following information available with restrictions as noted:

– Charge Number, Calling Party Number, Additional Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, Facility Information and other feature-related information – This information is available for each access type under the conditions defined in the O_Null PIC.

– Analysis Results: See description in the Analyse_Information PIC.

– Routing Information: When more than one route has been specified for the call (either by the SCF or as part of the information stored at the SSF/CCF), the SSF/CCF remembers what routes have been tried for this call and which route to select next. If the call is to an Inter Serving Area ID carrier, the routing information includes Circuit Code information. See Recommendation Q.1228.

Exit events:

– Route Selected event.

– Unable to select a route (e.g. unable to determine a correct route, no more routes on route list) or indication from the terminating half BCSM that call cannot be presented to the terminating party (e.g. network congestion). (DP: Route_Select_Failure.)

– The route busy event leading to the Analyse_Information PIC as described above, or the following four conditions are met:

  a) unable to select a route (e.g. unable to determine a correct route, no more routes on route list) or indication from the terminating half BCSM that call cannot be presented to the terminating party (e.g. network congestion);

  b) the forwarding condition is busy, as specified by the ForwardingCondition parameter;

  c) the route was determined by switch translations at the Analyse_Information PIC;

  d) there are more CDPNs left to try.

  Route busy is a non-IN transition which is part of a basic call. This is needed in case the SSF/CCF needs to process a list of CDPNs. These CDPNs can be returned in a Connect or Analyse_Information message. Upon being unable to select a route, the route busy BCSM transition returns to the Analyse_Information PIC to analyse the next CDPN in the list.

– Originating party abandons call (DP: Abandon).

4.2.2.1.6 Authorize_Call_Setup

Entry events: Route Selected event.

Function: The authority of the calling party to place this particular call is verified.

Information available: After the SSF/CCF determines the call set-up has been authorized, it is assumed the SSF/CCF has the following information available with restrictions as noted:

– Charge Number, Calling Party Number, Additional Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, Facility Information and other feature-related information – This information is available for each access type under the conditions defined in the O_Null PIC.

– Analysis Results – see description in the Analyse_Information PIC.

– Routing Information – see description in the Select_Route PIC.

– Facility Restriction Level – as described under the Collect_Information PIC.

Exit events:

– Call Setup Authorized event. The Call Setup Authorized event occurs when the authority to place the call is verified. For an SS7-supported trunk interface, if the received IAM indicates that a continuity check is being performed on the call connection and the call terminates to a non-ISDN line or ISDN interface, the Call Setup Authorized event occurs when an ISDN-UP Continuity Message (COT) with a successful indication is received.

– Originating party abandons call (DP: O_Abandon).

– The Authorization Failure event occurs when the authority to place the call is denied (e.g. business group restriction mismatch, toll restricted calling line). For an SS7-supported trunk interface, the Authorization Failure event occurs when the continuity check procedure results in failure. This event causes a BCSM transition to O_Exception.

4.2.2.1.7 Send_Call

Entry events: Call Setup Authorized event.

Functions: The SSF/CCF sends an indication of the desire to set up a call to the specified Called Party ID to the terminating call portion. The information that may be passed to the terminating call portion is: Charge Number, Calling Party ID; Calling Party BGID; Calling Party Category (determined by the Class of Service information or ISDN-UP originating line information parameter); Bearer Capability; Called Party ID; Calling Party Subaddress; Called Party Subaddress; Carrier; Route Index; Carrier Identification Code. Circuit Code, and Carrier Selection; and TCM. Other feature-information not used by the processing modelled by this PIC (e.g. call forwarding, generic name, and business group information) may also be passed to the terminating call portion.

For an ISDN caller, during the processing that this PIC models, the SSF/CCF returns a CALL PROCeeding message. For non-ISDN lines, conventional trunks, and private-facility trunks, no treatment is applied.

For SS7-supported trunks, if the received IAM indicates a continuity check is required in this circuit, the procedures for performing the continuity check are followed. If the continuity check is successful, an indication of continuity success is passed to the terminating call portion. If the received IAM indicates a continuity check is being performed on a previous circuit in the connection, upon receiving a COT with a successful indication, this indication is passed to the terminating call portion.

Information available: After the SSF/CCF determines the call has been delivered (to the terminating half), it is assumed the SSF/CCF has the following information available with restrictions as noted:

– Charge Number, Calling Party Number, Additional Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, Facility Information and other feature-related information – This information is available for each access type under the conditions defined in the O_Null PIC;

– Analysis Results: see description in the Analyse_Information PIC.

– Routing Information: see description in the Select_Route PIC.

– Facility Restriction Level: as described under the Collect_Information PIC.

Exit events:

– A route failure event is detected when:

  i) an indication of a T_Busy event specifying route busy; or

  ii) a Call Rejected event specifying route busy (received when the route is found to be busy at a switch other than the local switch) is received from the terminating call portion. In both cases, the originating call portion returns to the Select_Route PIC. This event is not detected at a DP in IN CS-2.

  iii) the following four conditions are met, hereafter called Route Failure Condition 1:

    a) an indication of a T_Busy event specifying route busy (received when the route at the local switch is found to be busy) is received from the terminating call portion (presentation failure event from the Present Call DP);

    b) the forwarding condition is busy, as specified by the ForwardingCondition parameter;

    c) the route was determined by switch translations at the Analyse_Information PIC;

    d) there are more CDPNs left to try.

  iv) the following four conditions are met, hereafter called Route Failure Condition 2:

    a) a Call Rejected event specifying route busy (received when the route is found to be busy at a switch other than the local switch) is received from the terminating call portion (presentation failure event from the Present Call PIC);

    b) the forwarding condition is busy, as specified by the ForwardingCondition parameter;

    c) the route was determined by the switch translations at the Analyse_Information PIC;

    d) there are more CDPNs left to try.

  v) the following four conditions are met, hereafter called Route Failure Condition 3:

    a) O_Called_Party_Busy event or O_No_Answer event occurs (as specified below);

    b) the forwarding condition is met, as specified by the ForwardingCondition parameter;

    c) there are more CDPNs left to try.

  NOTE – The Route_Failure event takes precedence over the O_Called_Party_Busy and O_No_Answer events.

  In all five cases, the originating call portion returns to the Select_Route PIC. This event is not detected at a DP in IN CS-2.

– An O_Answer event occurs when an indication of a T_Answer event is received from the terminating call portion. This event causes call processing to move to the O_Answer DP.

– An O_Term_Seized event occurs when an indication of a call accepted event is received from the terminating call portion or when certain abnormal cases occur in ISDN when the call is offered to an ISDN interface and no user equipment has responded, but an "indication to apply audible ringing" is sent from the terminating call portion to the originating call portion and as a result, audible ringing is to be sent to the caller. In ISDN this occurs, for example, when a call is offered to the interface with a SETUP message and the call setup timer T303 is initiated. If T303 expires and no response is received, the switch retransmits the SETUP message, re-initializes T303, and sends in-band audible ringing back to the caller. If the call is from an ISDN user, a PROGress message is also sent containing progress indicator #10, "delay in response at called interface" and progress descriptor #8, "in-band tone or pattern not available". In this case, the calling party receives in-band audible ringing, however, the called party has not accepted the call. When the O_Term_Seized event occurs, the treatment applied depends on the originating access type.

– A service feature request is received from the originating party: e.g. hook-flash, ISDN feature activator. (DP: O_Mid_Call). The Send_Call PIC is not applicable for the O_Mid_Call DP in case the SSF/CCF is located at transit level (due to restrictions of the basic network signalling).

No additional actions are taken in a non-ISDN line or private facility trunk.

For a call originating from an ISDN interface, the caller also receives an ALERTing message or under certain conditions a PROGress message containing progress indicator information set to "in-band information or pattern now available".

For SS7-supported trunks, an Address Complete Message (ACM) is sent.

In these cases, audible ringing, if applicable, is being sent from the originating call portion of the terminating switch.

– The O_No_Answer event is an IN event. That is, it can only occur when an O_No_Answer trigger is assigned and detected or when requested by a RequestReportBCSMEvent. If the O-No_Answer timer expires or an indication of the T_No_Answer event is received before an O_Answer event is detected (i.e. before the called party answers), the SSF/CCF reports the event to the SCF.

– The O_Called_Party_Busy event occurs when an indication of a T_Busy event specifying user busy is received from the terminating portion of the call (i.e. network-determined-user-busy). This event also occurs when an indication of a Call Rejected event specifying user busy (i.e. user-determined-user busy) is received from the terminating portion of the call. For calls originating from non-ISDN lines, conventional trunks, and private-facility trunks, if an indication of busy is received from the terminating portion of the call and no originating triggers or requested events apply, busy tone is provided.

  In addition to these busy events, the following "Call Rejected" conditions are also treated as O_Called_Party_Busy events:

  i) a Termination Denied event is received from the terminating call portion (Authorize_Termination PIC); or

  ii) an indication of a Call Rejected event is received from the terminating call portion (T_Alerting PIC) that does not specify busy. An example of a Call Rejected event that does not specify busy is the case when a call is offered to an ISDN interface; and either

     a) no user has responded with an ALERTing or CONNect message when the call setup timers expire; or

b) if the terminating user returns a call clearing message with a cause value of something other than "busy".

In this case, the terminating portion of the call is cleared.

– For SS7-supported trunk interface, the authorization_route_failure event occurs when the continuity check procedure results in failure. This event causes a BCSM transition to the O_Exception.

– Originating party abandons call (DP: O_Abandon).

4.2.2.1.8 O_Alerting

Entry event: O_Term_Seized event (DP: O_Term_Seized).

Function:

– Wait for the terminating party to answer. At this point, the caller receives in-band audible ringing (from the terminating switch). For a call originating from an ISDN interface, the caller also receives an ALERTing message or, under certain conditions, a PROGress message containing progress indicator information set to "in-band information or pattern now available".

– An indication of a call progress event may be received from the terminating call portion. This may result in a CPG being sent on an SS7-supported trunk (if the originating access is an SS7-supported trunk) or an ALERTing or PROGress message being sent on an ISDN interface (if the originating access is an ISDN interface).

Information available: When the SSF/CCF is in this PIC, it is assumed the SSF/CCF has the following information available with restrictions as noted:

– Charge Number, Calling Party Number, Additional Calling Party Number, Calling Party Business Group ID, Class of Service, Bearer Capability, Calling Facility Group, Calling Facility Group Member, Service Profile Identifier, Facility Information, other feature-related information: This information is available for each access type under the conditions defined in the O_Null PIC.

– Analysis Results: see description in the Analyse_Information PIC.

– Routing Information: see description in the Select_Route PIC.

– Facility Restriction Level: as described under the Collect_Information PIC.

Exit events:

– The O_Answer event occurs when an indication of a T_Answer event is received from the terminating portion of the call (e.g. terminating party goes off hook, Q.931 Connect message received, ISDN-UP Answer message received) (DP: O_Answer). When the O_Answer event occurs, the treatment applied is described in the Send_Call PIC.

– A service feature request is received from the originating party: e.g. hook-flash, ISDN feature activator. (DP: O_Mid_Call.)

– A route failure event is detected when:

i) the following conditions are met, hereafter called Route Failure Condition 3:

a) O_Called_Party_Busy event or O_No_Answer event occurs (as specified below);

b) the forwarding condition is met, as specified by the ForwardingCondition parameter;

c) there are more CDPNs left to try.

NOTE – The Route Failure event takes precedence over the O_Called_Party_Busy and O_No_Answer events.

In this case, the originating call portion returns to the Select_Route PIC. This event is not detected at a DP in IN CS-2.

– The O_No_Answer event from this PIC is the same as the O_No_Answer event defined as an Exit Event from the Send_Call PIC. (DP: O_No_Answer.)

– From this PIC, the O_Called_Party_Busy event occurs either when:

  i) a Call Rejected event specifying user busy is received; or

  ii) when an indication of a Call Rejected event not specifying busy is received from the terminating call portion (as described in the Send_Call PIC).

  In addition, for a call to an ISDN user, after the SETUP message is offered and an ALERTing message has been received (i.e. the terminating call portion is in the T_Alerting PIC), the ISDN user may reject the call. This Call Rejected event is treated as an O_Called_Party_Busy event by the originating call portion. (DP: O_Called_Party_Busy.)

– Originating party abandons call (DP: O_Abandon).

4.2.2.1.9 O_Active

Entry event: Indication from the terminating half BCSM that the call is accepted and answered by terminating party. (DP: O_Answer.)

Function: In this PIC several processes may be initiated:

- Connection established between originating and terminating party. Message accounting/charging data may be being collected. Call supervision is being provided.
- The called party may be put on hold and returned to the active phase by a service logic.
- The called party may be put on hold by a service logic and when the calling party disconnects, the calling party can be reconnected to the held call by the Reconnect information flow. The calling user receives appropriate information (e.g. re-ring) and a reconnection timer is applied.

Information available: Once the SSF/CCF has received an indication from the terminating half BCSM that the call has been answered, it is assumed the SSF/CCF has the following information available with restrictions as noted:

– Information as per the O_Alerting PIC.

– Feature Activation: A service or feature request from the originating party (e.g. DTMF, hook flash, ISDN feature activator, Q.931 HOLD or RETRIEVE message). (O_Mid_Call DP.)

Exit events:

– A service/service feature request is received from the originating party (e.g. DTMF, hook flash, ISDN feature activator, Q.931 HOLD or RETRIEVE message) or a new reply from calling party (Reconnect procedure). (DP: O_Mid_Call.)

– A disconnect indication is received from the terminating party via the terminating half BCSM. (DP: O_Suspend.) A disconnect timing is associated with this BCSM transition.

– A disconnect indication (e.g. on-hook, Q.931 disconnect message, SS7 release message) is received from the originating party. (DP: O_Disconnect.)

– An indication of expiration of the reconnection timer is received. (O_Exception.)

– A connection failure occurs. (O_Exception.)

Comments:

– Disconnect treatment and timing is different for call attempts originating from ISDN and analogue line interfaces.

Corresponding Q.931 call state: 10. Active

Q.931 call states corresponding to disconnect: 11. Disconnect request, 12. Disconnect indication and 19. Release request.

4.2.2.1.10 O_Suspended

Entry event: A suspend indication is received from the T_BCSM when the terminating party has disconnected (e.g. on-hook). (DP: O_Suspend.)

Function:

- The connection between the originating and terminating party is maintained and depending on the incoming network connection, appropriate backward signalling takes place.
  - In case that a disconnect indication is received from the T_BCSM, this PIC is immediately exited to the O_Disconnect DP without any action. As an option, the call can be continued for an appropriate period in order to offer follow-on initiated by O_Mid_Call.
  - If the re-answer indication from the T_BCSM is received, the originating and terminating parties are reconnected.
  - Other features which might be required during this PIC are not described in the IN CS-2 BSCM.
- The called party may be put on hold and returned to the active phase by a service logic.
- The called party may be put on hold by a service logic and when the calling party disconnects, the calling party can be reconnected to the suspended call by the Reconnect information flow. The calling user receives appropriate information (e.g. re-ring) and a reconnection timer is applied. The timer may have been started in the active phase.

Information available: It is assumed that the information available in this PIC is the same as the information available in the O_Active PIC.

Exit event:

– Connection to the terminating party is resumed. The O_BCSM returns to the O_Active PIC.

– A service feature request is received from the originating party, e.g. hook flash, ISDN feature activator of facility or a new reply from calling party (Reconnect procedure). (DP: O_Mid_Call.)

– A disconnection indication is received from the originating party. (DP: O_Disconnect.)

– A disconnection indication is received from the terminating party. (DP: O_Disconnect.)

– An indication of expiration of the timer waiting for re-answer request is received from the T_BCSM. (DP: O_Disconnect.)

– A trigger at O_Mid_Call is not initiated during an appropriate period. (DP: O_Disconnect.)

– An indication of expiration of the calling party reconnection timer is received. (O_Exception.)

– An exception event is encountered. (O_Exception).

NOTE 1 – A Call Retention timer may exist. Disconnect treatment and timing is different for call reconnection, call suspension and call retention.

NOTE 2 – After the release of the outgoing connection, the originating party may initiate another call, e.g. follow-on calling.

4.2.2.1.11 O_Exception

Entry event: An exception condition is encountered (as described above for each PIC).

Function: Default handling of the exception condition is being provided. This includes general actions necessary to ensure no resources remain inappropriately allocated, such as:

– If any relationships exist between the SSF and SCF(s), send an error information flow to the SCF(s) closing the relationships and indicating that any outstanding call handling instructions will not run to completion. (See note)

– If an SCF previously requested that call parameters be provided at the end of the call, these should be included in the error information flow.

– The SSF/CCF should make use of vendor-specific procedures to ensure release of resources within the SSF/CCF so that line, trunk, and other resources are made available for new calls.

NOTE – This should be handled in the physical plane via an ABORT protocol to close the relationship (i.e. close the TCAP transaction) and indicate that any outstanding operations will not be run to completion.

Information available: Once the SSF/CCF has determined an exception condition has occurred, it is assumed the SSF/CCF has information available as when the exception within the PIC occurred.

Exit event: Default handling of the exception condition by the SSF/CCF completed (BCSM transition to O_Null PIC).

4.2.2.2 Terminating BCSM for IN CS-2

The terminating half of the BCSM corresponds to that portion of the BCSM associated with the terminating party (see Figure 4-5).

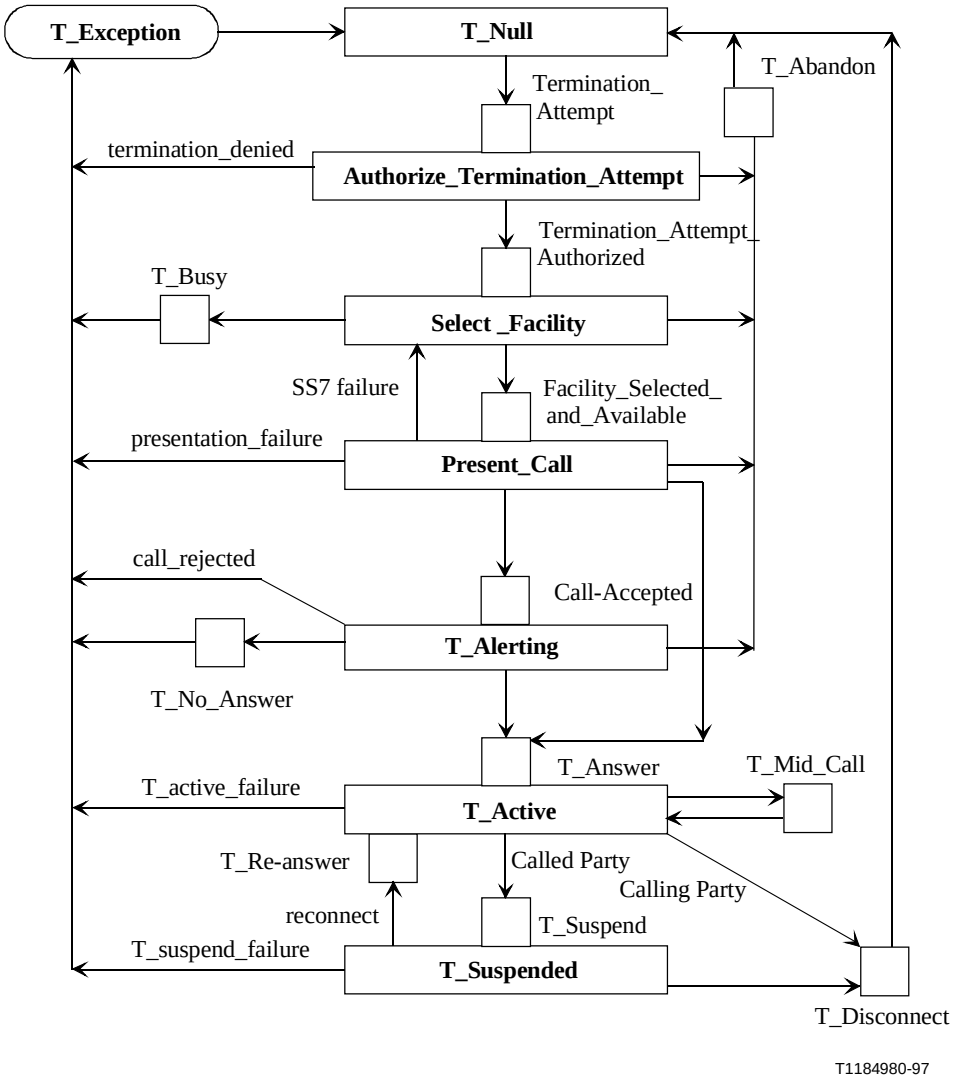


Figure 4-5/Q.1224 – Terminating BCSM for CS-2

The following information is available at all PICs in the T-BCSM:

– Service Address Information: see Recommendation Q.1290.

– SRF/SSF capabilities: see Recommendation Q.1290. Used to decide if an assist of hand-off procedure is to be used.

– Call Gapping Encountered: see Recommendation Q.1290.

– Terminal Type: see Recommendation Q.1290. The SCF uses this to determine the most appropriate form of user-interaction to use (e.g. in-band announcements). This information is only available at originating or terminating local exchanges.

– Location Number: see Recommendation Q.762 Location Number signalling information. Used if the calling party is a mobile subscriber.

– Original Called Party Number: see Recommendation Q.762. This information is received from the originating call portion.

– Redirecting Party ID: see Recommendation Q.762. This information is received from the originating call portion.

– Redirection Information: see Recommendation Q.762. This information is received from the originating call portion.

– ISDN Access Related Information: see Q.762 Access Transport Parameter.

The descriptions for each of the PICs in the terminating half of the BCSM are described below.

NOTE – See BCSM Indications of 4.2.4 for more information concerning PICs.

4.2.2.2.1 T_Null

Entry event: Disconnect and clearing of a previous call (DPs: T_Disconnect or T_Abandon), or default handling of exceptions by SSF/CCF completed.

Function: Interface (line/trunk) is idled (no call exists, no call reference exists, etc.). Supervision is being provided.

Information available: Once the SSF/CCF has detected the Termination_Attempt event it is assumed the following information is available and associated with the terminating portion of the call with restrictions as noted (information associated with the originating portion of the call as per the Send_Call PIC is assumed to be still available):

– Charge Number, Calling Party Number; Calling Party Business Group ID, Bearer Capability – Available for each access type under conditions identified in the O_Null PIC. This information is received from the originating call portion.

– Calling Party Category: see Recommendation Q.762. Determined by the Class of Service information, ISDN User Part originating line information parameter (see Recommendation Q.1290), or information from EAMF signalling.

– Called Party Number, Carrier, Carrier Identification Code, Circuit Code, Carrier Selection; Route Index; and Travelling Class Mark – This information is received from the originating call portion.

– Class of Service of Terminating Access: see Recommendation Q.1290. This is either a Customer Class of Service (see Recommendation Q.1290), a Trunk Class of Service (see Recommendation Q.1290), or a Private-Facility Class of Service (see Recommendation Q.1290) for the Terminating Access (Dialled Number, Circuit, or trunk group).

– Called Party Subaddress: see Recommendation Q.931.

– Calling Party Subaddress: see Recommendation Q.931.

– Called Party BGID: see Recommendation Q.1290. This information is determined in this PIC when the terminating party is a member of a Business Group and is served by a non-ISDN line or DSS 1 interface on this SSF/CCF.

– Service Compatibility ID: see Recommendation Q.1290. This information is received from the originating call portion and may be enhanced by SSF/CCF.

Exit event:

– Indication of incoming call received from originating half BCSM. (DP: Termination_Attempt.)

– The following exception exit events are applicable to this PIC: T_Abandon. If the call encounters T_Abandon during PIC processing, the exception event is not visible because there is no corresponding DP.

– The T_Abandon occurs when an indication of call disconnection is received from the originating part of the call.

Corresponding Q.931 call state: 0_Null

4.2.2.2.2 Authorize_Termination_Attempt

Entry event: Termination_Attempt event. (DP: Termination_Attempt.)

Function: Verifies the authority to route this call to the terminating access (e.g. DN or trunk group), e.g. check business group restrictions, restricted incoming access to line, or bearer capability compatibility.

Information available: It is assumed that the SSP has the same information available for the terminating call portion after the Termination_Authorized event is detected as it does when the Termination_Attempt event is detected.

Exit events:

– Termination_Attempt_Authorized event. This event occurs when the switch has verified the authority to terminate the call to the terminating access. (DP: Termination_Attempt_Authorized.)

– The Termination Denied event occurs when the authority to route these calls to the terminating user is denied. (This causes a BCSM transition to the T_Exception.)

– The T_Abandon event occurs when an indication of clearing is received from the originating portion of the call. (DP: T_Abandon.)

4.2.2.2.3 Select Facility

Entry event: Termination_Attempt_Authorized event (DP: Termination_Attempt_Authorized) or an SS7 failure occurs causing a re-attempt. The SS7 failure in the Present_Call can be caused by a timer expiry upon sending the first Circuit Reservation Message (CRM) or a continuity check failure.

Function: The busy/idle status of the terminating access is determined.

– For a non-ISDN line, if the line is already involved with an existing call, the line is treated as network-determined user busy.

– For a call terminating to an ISDN interface (on a non-shared DN/CT), network-determined user busy is the detection of one or more of the following conditions:

  - Interface busy: That is, a B-channel is not available for the call.
  - Call-reference busy: There are no idle call reference values available on the terminating DN/CT with which the call can be offered.

    In addition, if the terminating DN is associated with an Multi-Line Hunt Group, busy means that no hunt terminals within the group are available an the queue, if any, is full.
  - For conventional trunks, SS7-supported trunks, and private-facility trunks, busy is when all trunks within the selected trunk group are busy.

Information available: When the Facility_Selected and Available event is detected, it is assumed the following information is available and associated with the terminating portion of the call with restrictions as detailed noted:

– Information as per the T_Null PIC.

– Facility Group: see Recommendation Q.1290. For calls routed out of this SSF/CCF, this identifies the Trunk Group (private or public) that has been selected to route the call on. For calls terminating to a non-ISDN line or DSS 1 interface within the SSF/CCF, this may identify a particular Multi-line Hunt Group.

– Facility Group Member: see Recommendation Q.1290. For calls out of this SSF/CCF, this identifies the trunk (private or public) that has been selected to route the call on. For calls

terminating to a non-ISDN line DSS 1 interface on the SSF/CCF, this may identify the hunt-terminal within the Multi-line Hunt Group that has been selected for this call.

Exit events:

– Facility_Selected_and_Available event: This event occurs when the terminating access is not busy [i.e. an idle facility (e.g. B-channel, call appearance, or trunk) could be found]. (DP: Facility_Selected_and_Available.)

– A T_Busy event occurs when the terminating access is busy (as defined above). The T_Busy event may also be detected as a result of an analogue line being out of order, marked as busy by a customer make-busy key, or as a result of certain maintenance actions. (DP: T_Busy.)

  After detecting T_Busy, if IN service logic is not needed on the call and no switch-based features apply, an indication of the T_Busy event describing the type of busy (e.g. user or network) is passed to the originating call portion. If a terminating feature acts on the T_Busy event and changes the event (e.g. as in the call Waiting feature), the event is not passed to the Originating BCM.

– The T_Abandon event occurs when an indication of clearing is received from the originating portion of the call.

4.2.2.2.4 Present Call

Entry event: Facility_Selected_and_Available event. (DP: Facility_Selected_and_Available.)

Functions: Terminating resource informed of incoming call (e.g. line seizure, Q.931 Setup message, ISDN-UP IAM message). In the case of an analogue line, ringing is applied.

Information available: When the Call Accepted event is detected, it is assumed the following information is available and associated with the terminating portion of the call with restrictions as detailed noted:

– Information as per the T_Null PIC.

– Facility Group, Facility Group Member: See description in the Select Facility PIC.

– Information regarding the call connection: This information includes whether the call is end-to-end SS7 or not and whether the originating access is ISDN or non-ISDN.

Exit events:

– Terminating party is being alerted (e.g. ringing being applied, Q.931 ALERTING message, ISDN-UP ACM message). (DP: Call_Accepted.)

– Call is accepted and answered by terminating party (e.g. terminating party goes off-hook, Q.931 Connect message received, ISDN-UP answer message received). (DP: T_Answer.)

– Indication of originating party abandon received from originating half BCSM. (DP: T_Abandon.)

– The T_Abandon as described in T_Null PIC.

– A timer expiry upon sending the first Circuit Reservation Message (CRM) or a continuity check failure. (SS7 failure.) This event causes call processing to move to the Select Facility PIC.

– Presentation Failure: The call cannot be presented (e.g. ISDN user determined busy, ISDN-UP release message with busy cause). This event causes call processing to move to the T_Exception and is notified to the originating call portion (send Call PIC).

Corresponding Q.931 call state: 6. Call present

4.2.2.2.5 T_Alerting

Entry event: Terminating party is being alerted of incoming call. (DP: Call_Accepted.)

Function: An indication is sent to the originating half BCSM that if the terminating party is being alerted. Continued processing of call set-up (e.g. ringing, audible ring indication) is taking place. Waiting for the call to be answered by terminating party.

Information available: Once the terminating party is being alerted of the incoming call, it is assumed that information as per Present_Call PIC is available.

Exit events:

– Call is accepted and answered by terminating party (e.g. terminating party goes off-hook, Q.931 CONNect message received, ISDN-UP answer message received). (DP: T_Answer.)

– The following exception exit events are applicable to this PIC: call rejected, T_No_Answer and T_Abandon.

  - Call rejected exception event may happen when an ISDN user rejects a call while be alerted.
  - The T_No_Answer event occurs when the terminating party does not answer before the switch-based ringing timer expires. An indication of T_No_Answer event is passed to the originating half of the BCSM. This event leads to the T_No_Answer DP.
  - Indication of originating party abandon received from originating half BCSM. (DP: T_Abandon.)
  - The T_Abandon as described in the T_Null PIC.

Comment: For terminations to SS7 trunk groups, this PIC is entered upon the receipt of an address complete (ACM) message.

Corresponding Q.931 call states: 7. Call received and 8. Connect request

4.2.2.2.6 T_Active

Entry events: Call is accepted and answered by terminating party (e.g. terminating party goes off-hook, Q.931 Connect message received, ISDN-UP answer message received). (DP: T_Answer.)

Function: In this PIC several processes may be initiated:

- An indication is sent to the origination half BCSM that the terminating party has accepted and answered the call. Connection established between originating and terminating party. Call supervision is being provided.
- The calling party may be put on hold and returned to the active phase by a service logic.
- The calling party may be put on hold by a service logic and when the called party disconnects, the called party can be reconnected to the held call by the Reconnect information flow. The called user receives appropriate information (e.g. re-ring) and a reconnection timer is applied. The timer may have been started in the active phase.

Information available: Once the call is accepted and answered by the terminating party, it is assumed the following information is available and associated with the terminating portion of the call with restrictions as noted:

– Information as per T_Alerting.

– Feature Activation: see Recommendation Q.932. For this PIC, a service or feature request from the terminating party (e.g. DTMF, hook flash, ISDN feature activator, Q.932 HOLD or RETRIEVE message).

Exit events:

– A service/service feature request is received from the terminating party (e.g. DTMF, hook flash, ISDN feature activator, Q.931 HOLD or RETRIEVE message). (DP: T_Mid_Call.)

– A disconnect indication (e.g. on-hook, Q.931 disconnect message, SS7 release message) is received from the terminating party. (DP: T_Suspend.)

– A disconnect indication is received from the originating party via the originating half BCSM. (DP: T_Disconnect.)

– An indication of expiration of the reconnection timer is received. (T_Exception.)

– A connection failure occurs. (T_Exception.)

Comments:

– Disconnect indications and treatment are asymmetrical in the way disconnect timing is applied.

Corresponding Q.931 call state: 10. Active

Q.931 call states corresponding to T_Disconnect: 11. Disconnect request, 12. disconnect indication, and 19. Release request

4.2.2.2.7 T_Suspended

Entry event: An indication is received from the outgoing network that the terminating party has disconnected (e.g. on-hook). (DP: T_Suspend.)

Function: The physical resources associated with the call remain connected.

– According to the received indication the following applies:
  - A suspend indication is sent to the originating half BCSM.
  - For an SS7-supported trunk or an ISDN interface, in case that a disconnect indication (e.g. Q.931 disconnect message, SS7 release message) is received from the terminating party, this PIC is immediately exited to the T_Disconnect DP without any action.

– In the following cases, the timer is started and the call waits for re-answer request from the terminating party:
  1) For an SS7 supported trunk, in case of receiving network initiated suspend message.
  2) For an analogue interface, in case of detecting on-hook.

If re-answer request (e.g. off-hook, SS7 resume message) is received from the terminating party before the timer expires, the originating and terminating parties are reconnected.

NOTE – Both a Call Resume timer and a Call Retention timer may exist in this PIC. IN implementations may use a single timer for both conditions.

Information available: While in the T_Suspended PIC, it is assumed that the SSP has the same information available for the terminating call portion as it has in the T_Active PIC.

Exit event:

– The terminating party re-answers or a resume message is received before the timer expires; the T_BCSM returns to the T_Active PIC. (DP: T_Re-answer.)

– The timer expires. (DP: T_Disconnect.)

– A disconnection indication is received from the terminating party. (DP: T_Disconnect.)

– A disconnection indication is received from the originating party. (DP: T_Disconnect.)

– An exception event is encountered. (T_Exception.)

4.2.2.2.8 T_Exception

Entry event: An exception condition is encountered (as described above for each PIC).

Function: An indication of the exception condition is sent to the originating half BCSM. Default handling of the exception condition is being provided. This includes general actions necessary to ensure no resources remain inappropriately allocated, such as:

– If any relationships exist between the SSF and SCF(s), send an error information flow to the SCF(s) closing the relationships and indicating that any outstanding call handling instructions will not be run to completion (see Note).

– If an SCF previously requested that call parameters be provided at the end of the call, these should be included in the error information flow.

– The SSF/CCF should make use of vendor-specific procedures to ensure release of resources within the SSF/CCF so that line, trunk, and other resources are made available for new calls.

NOTE – This should be handled in the physical plane via an ABORT protocol procedure to close the relationship (i.e. close the TCAP transaction) and indicate that any outstanding operations will not be run to completion.

Information available: Once the SSF/CCF has determined an exception condition has occurred, it is assumed the SSF/CCF has information available as when the exception within the PIC occurred.

Exit event: Default handling of the exception condition by SSF/CCF completed (BCSM transition to T_Null PIC).

4.2.3 BCSM resume points and BCSM transitions in the IN CS-2 call model

Tables 4-1 and 4-2 together with Figures 4-6 and 4-7 describe the complete set of possible BCSM transitions for the IN CS-2 originating and terminating call models. The nature of the BCSM transitions is given in the third column.

Table 4-1/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Originating BCSM)

From	To	Nature of BCSM transitions
Origination_Attempt DP	Authorize_Origination_Attempt PIC	Basic
	Collect_Information PIC	Extended
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_Exception PIC	Extended
Orig._Attempt_Authorized DP	Collect_Information PIC	Basic
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_Exception PIC	Extended
	O_Abandon PIC	Extended

Table 4-1/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Originating BCSM) *(continued)*

From	To	Nature of BCSM transitions
Collected_Information DP	Collect_Information PIC (Collect_NDigits)	Extended
	Analyse_Information PIC	Basic
	Select_Route PIC	Extended
	O_Exception PIC	Extended
	O_Abandon PIC	Extended
Analyse_Information DP	Collect_Information PIC (Collect_NDigits)	Extended
	Analyse_Information PIC	Extended
	Select_Route PIC	Basic
	O_Exception PIC	Extended
	O_Abandon PIC	Extended
O_Term_Seized DP	O_Alerting PIC	Basic
	O_Active PIC (O_Disconnect)	Extended
	O_MidCall DP (O_Alerting PIC)	Extended
	O_Exception PIC	Extended
	O_Abandon PIC	Extended
Route_Select_Failure DP	Collect_Information PIC (Collect_NDigits)	Extended
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_Abandon PIC	Extended
	O_Exception PIC	Basic
O_Called_Party_Busy DP	ollect_Information PIC (Collect_NDigits)	Extended
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_Abandon PICO_Exception PIC	Extended
	O_Exception PIC	Basic
O_No_Answer DP	Collect_Information PIC (Collect_NDigits)	Extended
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_Abandon PIC	Extended
	O_Exception PIC	Basic
O_Answer DP	O_Active PIC (O_Answer)	Basic
	O_MidCall DP (O_Active)	Extended
	O_Exception PIC	Extended

Table 4-1/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Originating BCSM) *(continued)*

From	To	Nature of BCSM transitions
O_Suspend DP	O_Suspended PIC	Basic
	O_Disconnect DP	Extended
	O_Midcall DP (O_Suspended PIC)	Extended
	O_Exception PIC	Extended
	O_Active PIC (O_Disconnect)	Extended
O_Re-Answer DP	O_Active PIC	Basic
	O_Disconnect DP	Extended
	O_Midcall_Active DP	Extended
	O_Exception PIC	Extended
	O_Active PIC (O_Disconnect)	Extended
O_Mid_Call DP (Send Call PIC)	Send_Call PIC	Basic
	O_MidCall DP (Send_Call PIC)	Extended
	O_Active PIC (O_Disconnect)	Extended
	O_Exception PIC	Extended
O_Mid_Call DP (O_Alerting PIC)	O_Alerting PIC	Basic
	O_MidCall DP (O_Alerting PIC)	Extended
	O_Active PIC (O_Disconnect)	Extended
	O_Exception PIC	Extended
O_Mid_Call DP (O_Active PIC)	O_Active PIC	Basic
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_MidCall DP (O_Active PIC)	Extended
	O_Active PIC (O_Disconnect)	Extended
	O_Exception PIC	Extended
	O_Disconnect DP	Extended
O_Mid_Call DP (O_Suspended PIC)	O_Suspended PIC	Basic
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_MidCall DP (O_Suspended PIC)	Extended
	O_Active PIC (O_Disconnect)	Extended
	O_Exception PIC	Extended

Table 4-1/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Originating BCSM) *(continued)*

From	To	Nature of BCSM transitions
O_Disconnect DP	O_Null PIC	Basic
	Collect_Information PIC (Collect_NDigits)	Extended
	Analyse_Information PIC	Extended
	Select_Route PIC	Extended
	O_Exception PIC	Extended
O_Abandon DP	O_Null PIC	Basic
O_Null PIC	Origination_Attempt DP	Basic
Authorize_Origination_Attempt PIC	Origination_Attempt_Authorized DP	Basic
	O_Abandon DP	Basic
	O_Exception PIC	Basic
Collect_Information PIC	Collected_Information DP	Basic
	O_Abandon DP	Basic
	O_Exception PIC	Basic
Analyse_Information PIC	Analysed_Information DP	Basic
	O_Abandon DP	Basic
	O_Exception PIC	Basic
Select_Route PIC	Analyse_Information PIC	Basic
	Authorize_Call_Setup PIC	Basic
	Route_Select_Failure DP	Basic
	O_Abandon DP	Basic
	O_Exception PIC	Basic
Authorize_Call_Setup PIC	Send_Call PIC	Basic
	O_Abandon DP	Basic
	O_Exception PIC	Basic
Send_Call PIC	O_Term_Seized DP	Basic
	O_Mid_Call DP (Send_Call PIC)	Basic
	O_Called_Party_Busy DP	Basic
	O_Answer DP	Basic
	O_No_Answer DP	Basic
	Select_Route PIC	Basic
	O_Abandon DP	Basic
	O_Active PIC (O_Disconnect)	Extended
	O_Exception PIC	Basic
	Analyse_Information PIC	Extended

Table 4-1/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Originating BCSM) *(concluded)*

From	To	Nature of BCSM transitions
O_Alerting PIC	Select_Route PIC	Basic
	O_Mid_Call DP	Basic
	O_Answer DP	Basic
	O_No_Answer DP	Basic
	O_Called_Party_Busy DP	Basic
	O_Abandon DP	Basic
	O_Active PIC (O_Disconnect)	Extended
	O_Exception PIC	Basic
	Analyse_Information PIC	Extended
O_Active PIC	O_Midcall DP	Basic
	O_Disconnect DP	Basic
	O_Suspend DP	Basic
	O_Active PIC (O_Disconnect)	Extended
	O_Active PIC (O_Reconnect)	Extended
	O_Exception PIC	Basic
O_Suspended PIC	O_Re-Answer DP	Basic
	O_Mid_Call DP (O_Suspended PIC)	Basic
	O_Disconnect DP	Basic
	O_Exception PIC	Basic
O_Exception	O_Null PIC	Basic

Basic: The basic call process resumes in sequence of DP (e.g. Continue).

Extended: The basic call process progress due to SCF PIC instruction (e.g. Analysed_Information)

Table 4-2/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Terminating BCSM)

From	To	Nature of BCSM transition
Termination_Attempt DP	Authorize_Termination_Attempt PIC	Basic
	Select_Facility PIC	Extended
	T_Answer DP	Extended
	T_Abandon DP	Extended
	Call Accepted DP	Extended
	T_Exception PIC	Extended
Terminating_Attempt_Authorized DP	Select_Facility PIC	Basic
	T_Answer DP	Extended
	T_Abandon DP	Extended
	Call Accepted DP	Extended
	T_Exception PIC	Extended
Facility_Selected_and_Available DP	Present_Call PIC	Basic
	T_Answer DP	Extended
	T_Abandon DP	Extended
	Call Accepted DP	Extended
	T_Exception PIC	Extended
Call_Accepted DP	T_Alerting PIC	Basic
	T_Answer DP	Extended
	T_Abandon DP	Extended
	T_Active PIC (T_Disconnect)	Extended
	T_Exception PIC	Extended
	T_MidCall DP (T_Active PIC)	Extended
T_Busy DP	Select_Facility PIC	Extended
	Present_Call PIC	Extended
	T_Answer DP	Extended
	T_Abandon DP	Extended
	Call Accepted DP	Extended
	T_Exception PIC	Basic
T_No_Answer DP	Select_Facility PIC	Extended
	T_Answer DP	Extended
	Call Accepted DP	Extended
	T_Exception PIC	Basic

Table 4-2/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Terminating BCSM) *(continued)*

From	To	Nature of BCSM transition
T_Answer DP	T_Active PIC	Basic
	T_Active PIC (T_Disconnect)	Extended
	T_MidCall DP (T_Active PIC)	Extended
	T_Exception PIC	Extended
T_Suspend DP	T_Suspended PIC	Basic
	T_Disconnect DP	Extended
	T_Active PIC (T_Disconnect)	Extended
	T_MidCall DP (T_Active PIC)	Extended
	T_Exception PIC	Extended
T_Re-Answer DP	T_Active PIC	Basic
	T_Disconnect DP	Extended
	T_Active PIC (T_Disconnect)	Extended
	T_MidCall DP (T_Active PIC)	Extended
	T_Exception PIC	Extended
T_Midcall DP	T_Active PIC	Basic
	T_Disconnect DP	Extended
	T_Active PIC (T_Disconnect)	Extended
	T_Exception PIC	Extended
T_Disconnect DP	T_Null PIC	Basic
	T_Answer DP	Extended
	Call_Accepted DP	Extended
T_Abandon DP	T_Null PIC	Basic
T_Null PIC	Termination_Attempt DP	Basic
Authorize_Termination_Attempt PIC	Term_Attempt_Authorized DP	Basic
	T_Abandon DP	Basic
	T_Exception PIC	Basic
Select_Facility PIC	Facility_Selected_and_Available DP	Basic
	T_Busy DP	Basic
	T_Abandon DP	Basic

Table 4-2/Q.1224 – Complete Set of BCSM Transitions for the IN CS-2 Model (Terminating BCSM) *(concluded)*

From	To	Nature of BCSM transition
Present_Call PIC	T_Answer DP	Basic
	Call_Accepted DP	Basic
	T_Alerting PIC	Basic
	Select_Facility PIC	Basic
	T_Abandon DP	Basic
	T_MidCall DP (T_Active PIC)	Extended
	T_Busy DP	Basic
	T_Active PIC (T_Disconnect)	Extended
	T_Exception PIC	Basic
T_Alerting PIC	T_Answer DP	Basic
	T_No_Answer DP	Basic
	T_Abandon DP	Basic
	T_MidCall DP (T_Active PIC)	Extended
	T_Active PIC (T_Disconnect)	Extended
	T_Exception PIC	Basic
T_Active PIC	T_Midcall DP	Basic
	T_Disconnect DP	Basic
	T_Suspend DP	Basic
	T_Active PIC (T_Disconnect)	Extended
	T_Exception PIC	Basic
T_Suspended PIC	T_Re-Answer DP	Basic
	T_Disconnect DP	Basic
	T_MidCall DP (T_Active PIC)	Extended
	T_Active PIC (T_Disconnect)	Extended
	T_Exception PIC	Basic
T_Exception	T_Null PIC	Basic

Basic: The basic call process resumes in sequence of DP (e.g. Continue).

Extended: The basic call process progresses due to SCF instruction PIC (e.g. Analysed_Information).

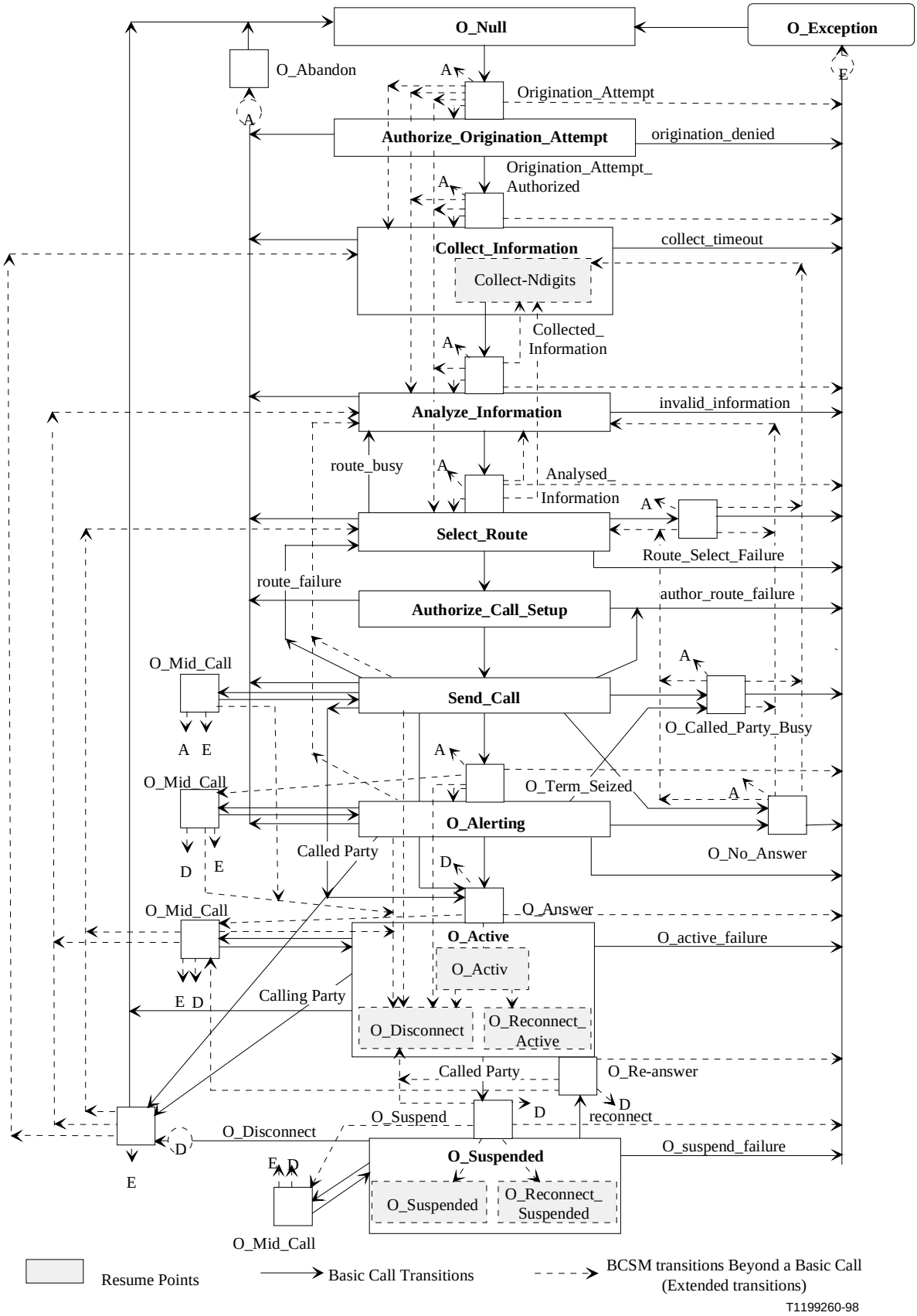


Figure 4-6/Q.1224 – Complete set of BCSM transitions for the IN CS-2 originating call model

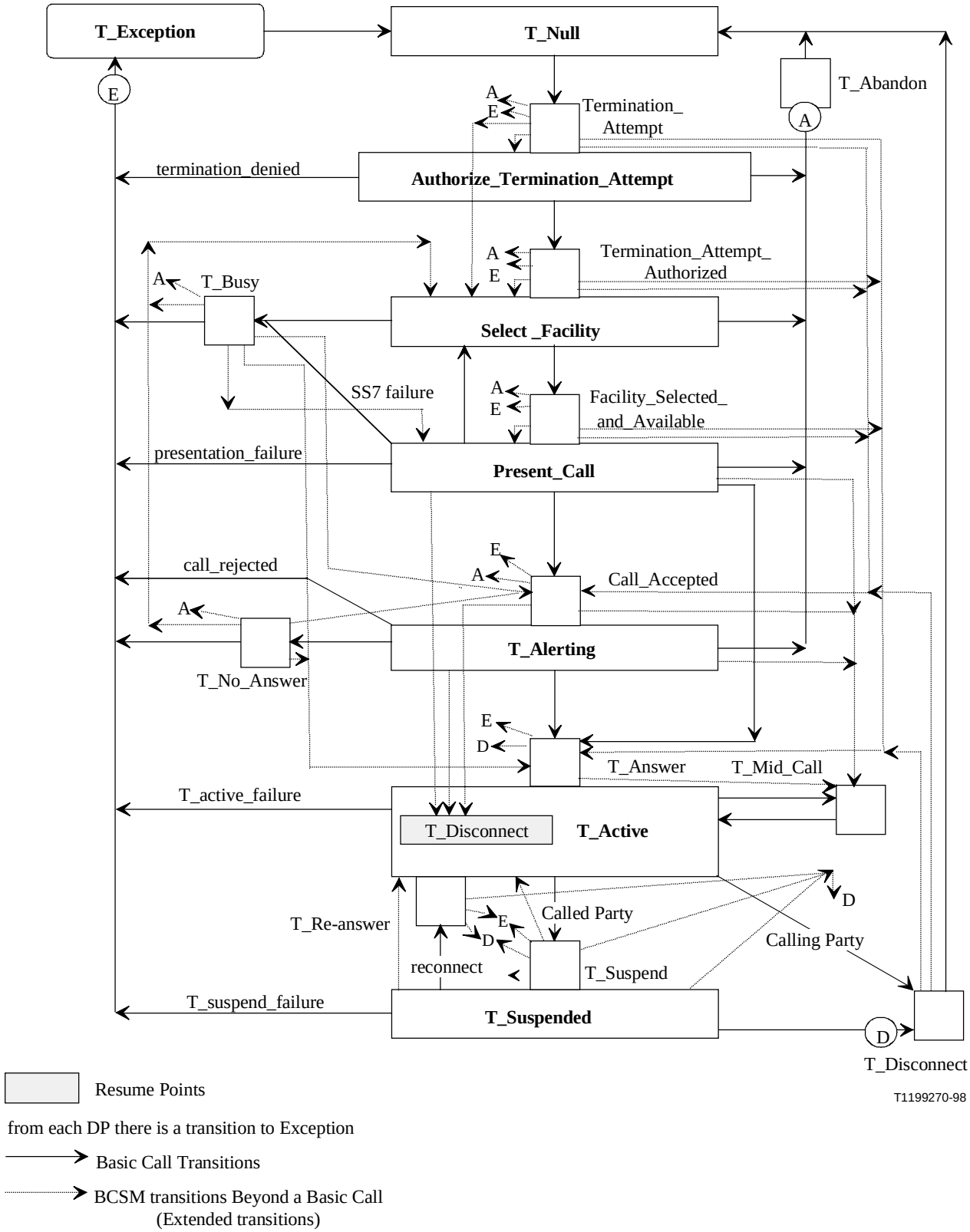


Figure 4-7/Q.1224 – Complete set of BCSM transitions for the IN CS-2 terminating call model

4.2.4 BCSM indications for the CS-2 call model

4.2.4.1 User – O_BCSM Access Signalling Indications (Category 1)

Definition

These Indications include the representation of the network's perception of possible actions taken by the calling party as well as the calling party's perception of actions taken by the network. The Indications are between a user (i.e. calling party) and a local exchange that is originating a call. They include the definition of how actions by the user (originating call model) affect the originating call model (user). These Indications are derived from Access Signalling (e.g. DSS 1, analogue) as well as any other information that is available. Figure 4-8 illustrates these indications.

Indications

(1) An Indication is sent from User to O_BCSM to initiate call establishment (e.g. SETUP).

(2) An Indication is sent from O_BCSM to User that network is unable to initiate call (e.g. RELEASE_COMPLETE).

(3) An Indication is sent from O_BCSM to User acknowledging the call initiation Indication (e.g. SETUP_ACKNOWLEDGE).

(4) The User sends call (dialling) information to the O_BCSM (e.g. INFORMATION).

(5) An Indication is sent from O_BCSM to the User to terminate the sending of call information (e.g. CALL_PROCEEDING).

(6) An Indication is sent from the User to the O_BCSM upon completion of call information.

(7) User is informed that call has been routed to another environment of network (e.g. PROGRESS).

(8) An Indication is sent from the O_BCSM to the User when the called party is being alerted (e.g. ALERTING, CONNECT).

(9) An Indication is sent from the O_BCSM to the User when the call is accepted.

(10) The User acknowledges that the call is accepted.

(11) The O_BCSM sends an Indication to the User that the called party is unable to accept the call, due to busy condition.

(12) The O_BCSM sends an Indication to the User since the called party is unable to accept the call, due to no answer condition.

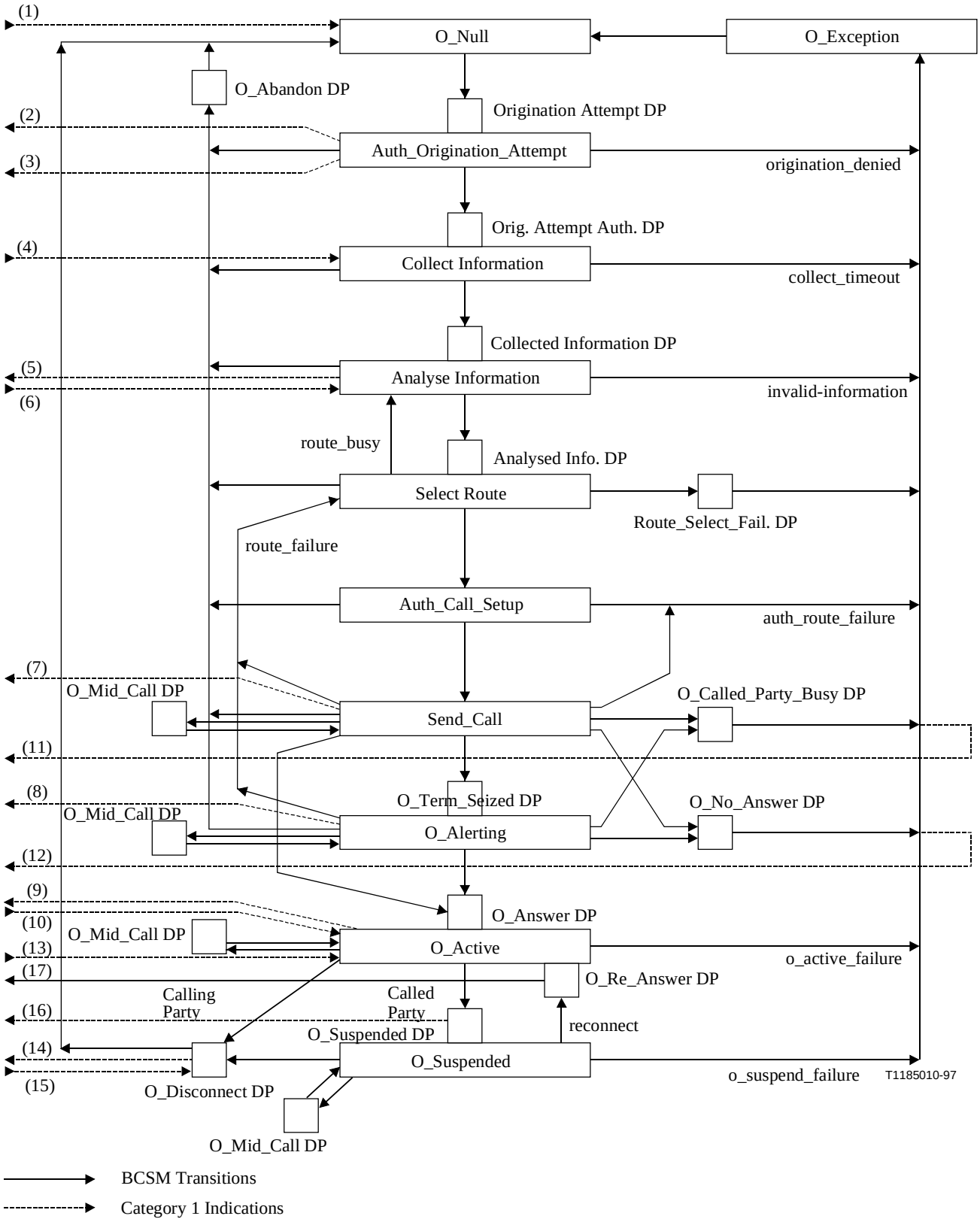
(13) An Indication is received by the O_BCSM from the User to end the call.

(14) The O_BCSM indicates to the User that the call is being disconnected.

(15) The User acknowledges to the O_BCSM that the call is being disconnected.

(16) An Indication is sent to the user when the connection towards the Called Party is suspended.

(17) An Indication is sent to the user when the connection towards the Called Party is reconnected.



NOTE – Indications which are shown as terminating on a DP in Figure 4-8 are received by the switch and are not part of the IN DP Processing.

Figure 4-8/Q.1224 – Access Signalling Indications for the CS-2 BCSM Category 1 (User – O_BCSM)

4.2.4.2 T_BCSM – User Access Signalling Indications (Category 2)

Definition

These Indications include the representation of the network's perception of possible actions taken by the called party as well as the called party's perception of actions taken by the network. The Indications are between a local exchange that is terminating a call and a user (i.e. called party). They include the definition of how actions by the terminating call model (user) affect the user (terminating call model). These Indications are derived from Access Signalling (e.g. DSS 1, analogue) as well as any other information that is available. Figure 4-9 illustrates these Indications.

Indications

(1) An Indication is sent from T_BCSM to the User to terminate the call to an idle facility (e.g. SETUP).

(2) An Indication is sent from User to T_BCSM indicating that the User cannot accept the call (e.g. RELEASE_COMPLETE).

(3) An Indication is sent from the User to the T_BCSM when the User determines compatibility with all call characteristics (e.g. SETUP_ACKNOWLEDGE).

(4) The T_BCSM sends any remaining call information to the User (e.g. INFORMATION).

(5) An Indication is sent from the T_BCSM to the User upon the sending of sufficient call information.

(6) An Indication is sent from the User to the T_BCSM upon receipt of sufficient call information (e.g. CALL_PROCEEDING).

(7) User sends an Indication to the T_BCSM that alerting is taking place (e.g. ALERTING).

(8) An Indication is sent from the User to the T_BCSM upon acceptance of the incoming call (e.g. CONNECT).

(9) An Indication is sent from the T_BCSM to the User acknowledging that the call can now be connected.

(10) An Indication is sent from the User to the T_BCSM that the User suspends the call.

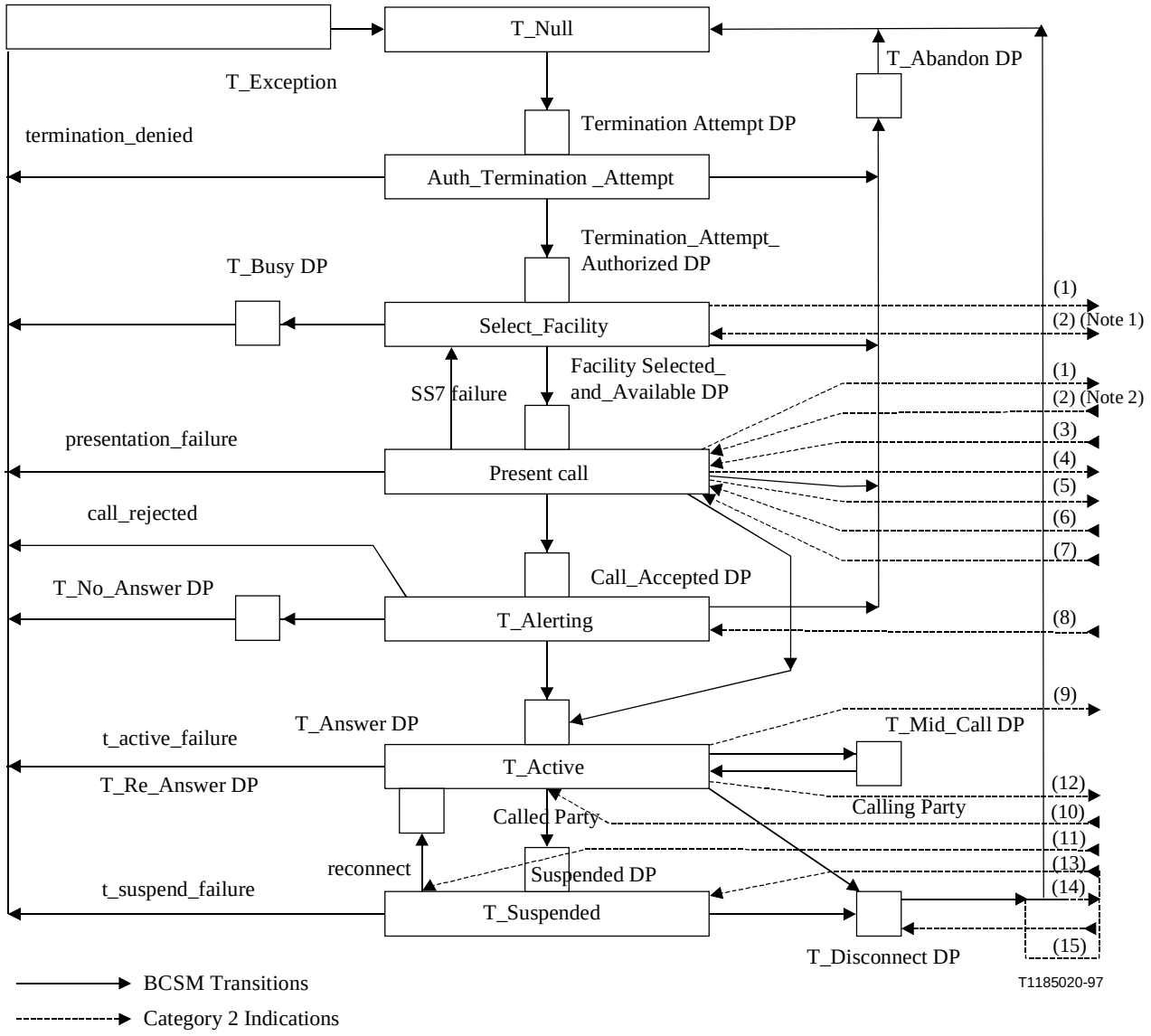
(11) An Indication is sent from the User to the T_BCSM that the User resumes the call.

(12) The T_BCSM sends an Indication to the User indicating that the calling party has gone on-hook.

(13) An Indication is received by the T_BCSM from the User to end the call.

(14) The T_BCSM indicates to the User that the call is being disconnected.

(15) The User acknowledges to the T_BCSM that the call is being disconnected.



NOTE 1 – Indications 1 and 2 at Select_Facility for case of switch-CPE B-channel negotiation.
NOTE 2 – Indications 1 and 2 at Present_Call for case of no B-channel negotiation.

NOTE 3 – Indications which are shown as terminating on a DP in Figure 4-9 are received by the switch and are not part of the IN DP processing.

Figure 4-9/Q.1224 – Access Signalling for the CS-2 BCSM Category 2 (T_BCSM-User)

4.2.4.3 Intra Local Exchange BCSM Indications (Category 3)

The following figure illustrates the communication between two call segments in the SSF/CCF for a basic two-party call, as described in 4.2.4.1 and 4.2.4.2. It shows the indications that flow between the originating and terminating BCSMs for CS-2, as described in 4.2.2.2. All possible indications are shown, except for any which may occur at the O-Exception and the T-Exception PICs. Note that these indications are not intended to be mapped to explicit information flows.

(1) Initiate T_BCSM after the authority to place the call has been verified and a usable route has been identified. The O_BCSM is currently in the Send_Call PIC. The originating Basic Call Manager has sent the call attempt to the terminating Basic Call Manager for further processing.

(2) For SS7-supported trunks, if the received IAM indicates a continuity check is required and the resultant continuity check is successful, then an Indication is sent from the O_BCSM to the T_BCSM (causes T_Null PIC to Termination_Attempt DP BCSM transition in the T_BCSM).

(3) An Indication is sent from the T_BCSM to O_BCSM that the terminating line or trunk is busy. (Causes Send_Call PIC to O_Called_Party_Busy BCSM transition in O_BCSM, if terminating to a line. If terminating to a trunk, then this Indication causes Send_Call PIC to Select_Route PIC BCSM transition in the O_BCSM.)

(4) An Indication is sent from the T_BCSM to O_BCSM that the terminating line or trunk is busy. (Causes O_Alerting PIC to O_Called_Party_Busy DP BCSM transition in O_BCSM.)

(5) An Indication is sent from the T_BCSM to O_BCSM that the call cannot be presented. (Causes Send_Call PIC to Select_Route PIC, O_Called_Party_Busy DP, or O_No_Answer DP.)

(6) An Indication is sent from the T_BCSM to the O_BCSM that an ISDN capable Called Party has signalled call acceptance with immediate BCSM transition to an answered (i.e. CONNect message) condition (causes Send_Call PIC to O_Answer DP BCSM transition in O_BCSM).

(7) An Indication is sent from T_BCSM to O_BCSM that Called Party is being alerted (causes O_BCSM to transit from Send_Call PIC O_Alerting PIC and prepare to send ring Indication to the Calling Party).

(8) An Indication is sent from T_BCSM to O_BCSM that Called Party has rejected the call (this is indicated to the O_BCSM with a busy cause and causes O_BCSM to transit from O_Alerting PIC to Select_Route PIC or O_Called_Party_Busy DP).

(9) An Indication is sent from T_BCSM to O_BCSM that Called Party has not answered within a specified time period (causes O_Alerting PIC to O_No_Answer DP BCSM transition in O_BCSM).

(10) An Indication is sent from the T_BCSM to the O_BCSM that called party has not answered within a specified time period. (Causes Send_Call PIC to O_No_Answer DP BCSM transition in O_BCSM.)

(11) An Indication is sent from T_BCSM to O_BCSM that Called Party has accepted and answered the call attempt (causes O_Alerting PIC to O_Answer DP BCSM transition in O_BCSM).

(12) An Indication is sent from the T_BCSM to the O_BCSM that the called party has accepted and answered the call attempt. (Causes Send_Call PIC to O_Answer DP BCSM transition in O_BCSM.)

(13) An Indication is sent from T_BCSM to O_BCSM that Called Party has disconnected (e.g. on-hook), (causes O_Active PIC to O_Suspended DP BCSM transition in O_BCSM).

(14) An Indication is sent from T_BCSM to O_BCSM that Called Party re-answers is received before the timer expires (causes O_Suspended PIC to O_Re_Answer DP BCSM transition in O_BCSM). Note that the name and function of this timer is for futhrt study.

(15) An Indication is sent from O_BCSM to T_BCSM that Calling Party has disconnected, while T_BCSM was in T_Active PIC (causes T_Active PIC to T_Disconnect DP BCSM transition in T_BCSM).

(16) An Indication is sent from O_BCSM to T_BCSM that Calling Party has disconnected, while T_BCSM was in T_Suspended PIC (causes T_Suspended PIC to T_Disconnect DP BCSM transition in T_BCSM).

(17) An Indication is sent from T_BCSM to O_BCSM that Called Party has disconnected (causes O_Suspended PIC to O_Disconnect DP BCSM transition in O_BCSM).

(18) An Indication is sent from the T_BCSM (T_Disconnect DP) to O_BCSM that the calling party has disconnected. (Causes O_Active PIC to O_Disconnect DP BCSM transition in O_BCSM.)

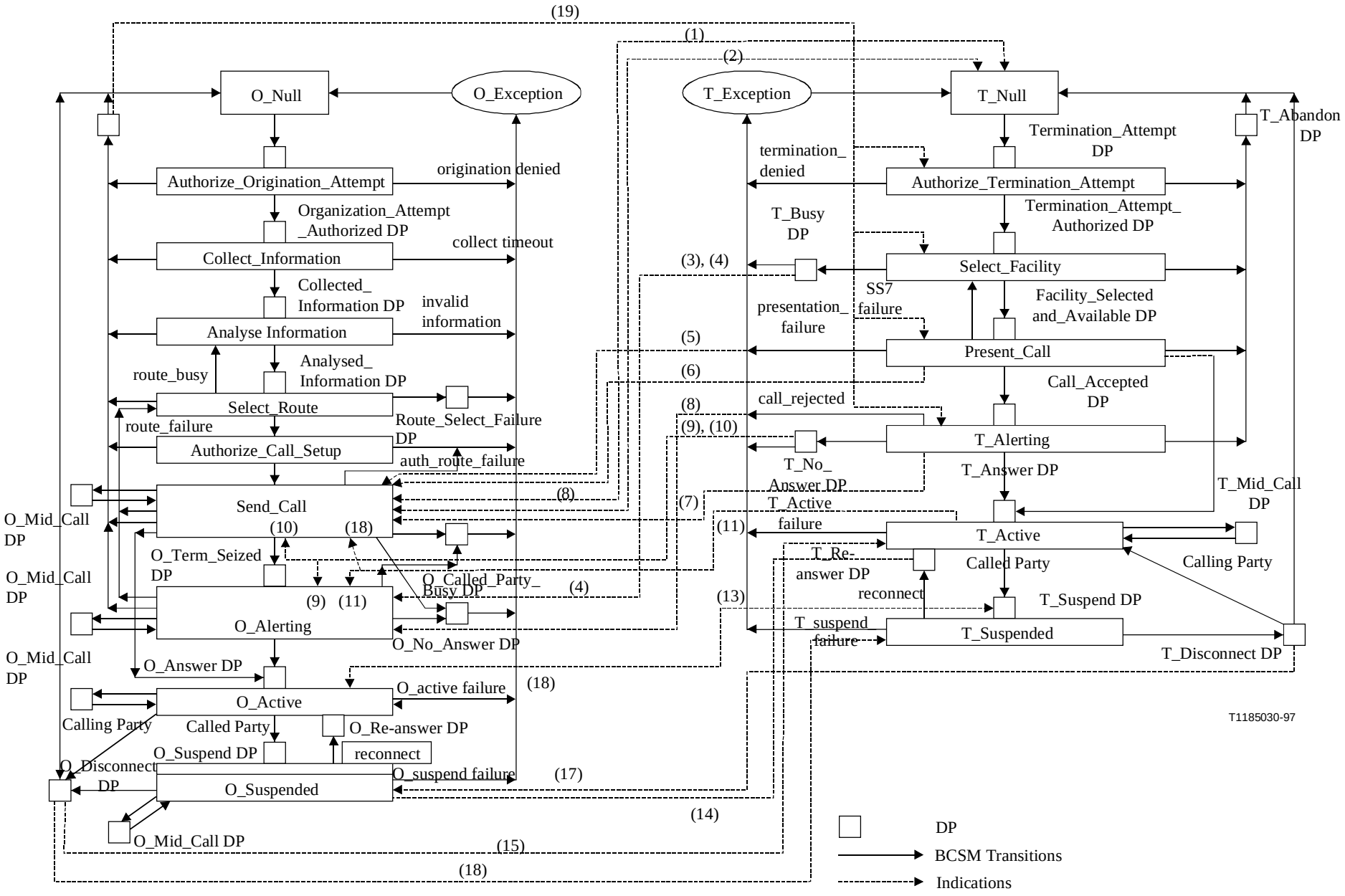
(19) An Indication is sent from O_BCSM to T_BCSM that Calling Party has abandoned (causes Authorize_Termination_Attempt PIC, Select_Facility PIC, Present_Call PIC or T_Alerting PIC to T_Abandoned DP BCSM transition in T_BCSM).

NOTE – Indications (15) and (17) are mutually exclusive:

– these indications are for intra-switch;

– the indications do not explicitly include the modelling of SRFs;

– indications which are preceded by a DP may be affected depending on whether the DP is active and the SCF response.


(19)
(1)
(2)
O_Null
O_Exception
T_Exception
T_Null
T_Abandon DP
Termination_Attempt DP
termination_ denied
Authorize_Termination_Attempt
Authorize_Origination_Attempt
origination denied
Organization_Attempt _Authorized DP
T_Busy DP
Termination_Attempt_ Authorized DP
Collect_Information
collect timeout
(3), (4)
Select_Facility
Collected_ Information DP
invalid information
SS7 failure
Facility_Selected and_Available DP
presentation_ failure
Analyse Information
(5)
Present_Call
Analysed_ Information DP
route_busy
(6)
Call_Accepted DP
Select_Route
(8)
call_rejected
Route_Select_Failure DP
route_failure
(9), (10)
T_Alerting
Authorize_Call_Setup
auth_route_failure
T_No_ Answer DP
T_Answer DP
T_Mid_Call DP
Send_Call
(8)
(7)
T_Active failure
O_Mid_Call DP
(10)
(18)
(11)
T_Active
O_Term_Seized DP
T_Re- answer DP
Called Party
Calling Party
O_Mid_Call DP
O_Called_Party_ Busy DP
(4)
(9)
(11)
reconnect
(13)
T_Suspend DP
O_Alerting
T_suspend failure
T_Suspended
O_No_Answer DP
T_Disconnect DP
O_Mid_Call DP
O_Answer DP
O_active failure
(18)
O_Active
Calling Party
Called Party
O_Re-answer DP
O_Suspend DP
reconnect
O_Disconnect DP
O_suspend failure
(17)
O_Suspended
(14)
O_Mid_Call DP
(15)
(18)
T1185030-97
DP
BCSM Transitions
Indications

4.2.5 BCSM detection points

Certain basic call and connection events may be visible to IN service logic instances. DPs are the points in call processing at which these events are detected. DPs for the BCSM are identified in 4.2.2.

A DP can be armed in order to notify an IN service logic instance that the DP was encountered, and potentially to allow the IN service logic instance to influence subsequent call processing. If a DP is not armed, the SSF/CCF continues call processing without SCF involvement. DPs are characterized by the following four attributes:

a) *Arming/disarming mechanism* – The mechanism by which the DP is armed. A DP may be statically armed or dynamically armed. A DP is statically armed through SMF service feature provisioning. A statically armed DP remains armed until explicitly disarmed by the SMF. The ability of an SCF to statically arm or disarm a DP is for further study. A DP is dynamically armed by the SCF within the context of a call-associated IN service control relationship.

   The following DP disarming rules apply:

   – If an armed EDP is met, then it is disarmed.
   – If an EDP is met that causes the release of the related leg, then all EDPs related to that leg are disarmed.
   – If a call is released, then all EDPs related to that call are disarmed.

b) *Criteria* – In addition to the condition that a DP be armed, conditions that must be met in order to notify the SCF that the DP was encountered (see 4.2.8).

c) *Relationship* – Given that an armed DP was encountered and DP criteria are met, the SSF may provide an information flow via a relationship:

   i) If this relationship is between the SSF/CCF and the SCF for the purpose of call/service logic processing, it is considered to be an IN service control relationship. This relationship may be of two types:

      – a control relationship if the SCF is able to influence call processing via the relationship;
      – a monitor relationship if the SCF is not able to influence call processing via the relationship.

      With respect to an IN service control relationship, the information flow provided by the SSF to the SCF on encountering a DP may initiate a control relationship, may be within the context of an existing control relationship, or may be within the context of an existing monitor relationship.

   ii) If this relationship is between the SSF/CCF and the SCF or SMF for management purposes, it is considered to be a service management control relationship. This relationship is for further study.

d) *Call processing suspension* – Given that an armed DP was encountered and DP criteria are met for an IN service control relationship, the SSF may suspend call processing to allow the SCF to influence subsequent call processing. When call processing is suspended, the SSF sends an information flow to the SCF requesting instructions, and waits for a response. When call processing is not suspended, the SSF sends an information flow notifying the SCF that a DP was encountered, and does not expect a response. This attribute is set by the same mechanism that arms the DP.

Based on these attributes, four types of DPs are identified for CS-1. The DP types are:

1) Trigger Detection Point-Request (TDP-R);

2) Trigger Detection Point-Notification (TDP-N);

3) Event Detection Point-Request (EDP-R);

4) Event Detection Point-Notification (EDP-N).

These DP types are defined by the DP attribute values in Table 4-3.

BCSM DPs may be any one of these DP types. DP processing for each DP type is illustrated in Figure 4-10 and described in 4.2.8.

Table 4-3/Q.1224 – BCSM DP Types

DP type	Arming mechanism	Criteria	IN service relationship	Suspension	Service feature examples
TDP-R	Static	Specific to DP	Initiates control relationship	Yes	All
TDP-N	Static	Specific to DP	Initiates and terminates monitor relationship	No	Televoting, call logging
EDP-R	Dynamic	None	Within context of existing control relationship	Yes	Call distribution, call rerouting distribution
EDP-N	Dynamic	None	Within context of existing control or monitor relationship	No	Charging for any service feature, call logging, call queueing

4.2.6 DP Criteria

As stated in 4.2.5, DP criteria are conditions that must be met in order to notify the SCF that the DP was encountered. These criteria can be assigned to a DP from the viewpoint of range of effectiveness, as identified below:

– *Individual line/trunk-based criteria*

  This type of criteria applies to each subscriber line or trunk line. For example, SCF processing is invoked when user A makes call origination. This criteria could be said to be specific for user A.

– *Group-based criteria*

  This type of criteria applies to a certain group of lines or users. For example, when a call origination from any user in a certain centrex group should invoke SCF processing the trigger should apply to that specific centrex group.

– *Office-based criteria*

  This type of criteria applies to the whole office. Any calls generated in the switching system will be subject to this criteria. For example, any call which makes access to the registered Freephone number is triggered and SCF processing is invoked.

The following criteria are DP criteria for IN CS-2, as applicable for a given DP:

– trigger assigned (unconditional/conditional on other criteria);
– class of service;
– specific B-channel identifier;
– specific digit strings;
– feature codes (e.g. *XX, #);
– prefixes (e.g. 0+, 00+, 0–, 00–, 011, 01, 1+);
– access codes (e.g. 8+) for customized numbering plan;
– specific abbreviated dialling strings for customized numbering plan;
– specific calling party number strings;
– specific called party number strings;
– nature of address (e.g. subscriber significant number, national significant number, international number);
– bearer capability;
– feature activation/indication (unconditional/conditional on specific feature patterns);
– facility information (unconditional/conditional on specific facility information patterns);
– cause (unconditional/conditional on specific cause patterns).

With respect to the DP criteria listed above, note that these DP criteria only apply to TDPs. DP criteria for Event Detection Points (EDPs) are addressed by the RequestReportBCSMEvent information flow. In addition, note that one or more DP criteria may be applicable at a given DP. The assignment of DP criteria to a TDP and the combinations of DP criteria applicable at a given DP continue to evolve. Further DP criteria and specific assignment of DP criteria to TDPs/EDPs may evolve through future capability sets. Note further that the assignment of DP criteria to a TDP on either a line/trunk, group or office basis may have an impact on the memory and real-time performance requirements of the SSF/CCF. The DP criteria for IN CS-2 are defined below, as applicable to a given TDP.

Note that the applicability of DP criteria at a given DP depends on when call processing information is available and how long it is retained. If network and service providers plan to implement IN CS-2 services in a multi-supplier environment, they should consider formulating such requirements to ensure consistent implementations across supplier equipment. Such requirements should be considered carefully so as not to adversely impact memory and real-time performance aspects of SSF/CCF processing.

1) *Trigger assigned*[2] (unconditional/conditional on other criteria) – An indicator of the armed/disarmed status of a TDP assigned on a line/trunk, group, or office basis.

   The trigger assigned criterion can be used by itself or in conjunction with other criteria at a TDP. If the trigger assigned criterion is unconditional at a TDP, then it is used by itself – no other DP criterion needs to be satisfied at the TDP before informing the SCF that the TDP was encountered. If the trigger assigned criterion is conditional at a TDP, then it is used in combination with other criteria at the TDP – all of the other DP criteria in the combination need to be satisfied before informing the SCF that the TDP was encountered.

   Applies at all DPs (all DPs can be provisioned as TDPs).

[2] It is possible that some DPs are always conditional. Further study may be required.

2) *Class of Service* – This is either a:

i) customer class of service,

ii) trunk class of service, or

iii) private facility class of service

Item i) is a code that identifies all attributes of a line that require distinctive call processing treatment (e.g. for party lines and coin lines), ii) is a code that identifies attributes of a trunk group such as type of signalling used, and iii) is a code that identifies attributes of a private trunk group such as type of signalling used and flash repeat capability.

Originating access (user/network) class of service is available at the Origination_Attempt DP and could be applicable at any of the originating DPs. Terminating access (user/network) class of service is available at the Termination_Attempt DP and could be applicable at any of the terminating DPs.

3) *Specific B-channel identifier* – An identifier of the specific B-channel on an ISDN interface from which a call attempt has originated or to which a call attempt is to be terminated.

A-party B-channel identifier is available at the Origination_Attempt DP for a party served by an ISDN interface only and could be applicable at all originating DPs. B-party B-channel identifier is available during the Select_Facility PIC after an idle terminating facility has been selected for a party served by an ISDN interface only and could be applicable at the Facility_Selected_and_Available, T_No_Answer, T_Answer, T_Mid_Call, T_Suspended, T_Re_Answer and T_Disconnect DPs and at the T_Abandon DP (only after an idle terminating facility has been selected).

4) *Specific digit strings* – A string of digits that must match collected digit strings for numbering plans in which a variable number of digits are to be collected. It could be zero or more digits (e.g. to trigger on "off-hook delay").

The string of digits should be consistent with the structure of the dialling plan and should be administrable. For example, the network provider may specify the first N digits where N is consistent with the structure of the E.164 numbering plan, or any other appropriate numbering plan.

Collected digit strings can be available at the Origination_Attempt DP for a party served by an ISDN interface using *en bloc* sending and at the Collected_Info DP for a party served by a non-ISDN line. Since collected digit strings are not analysed until the Analyse_Information PIC (except to determine if a sufficient number of digits have been collected), this criteria could be applicable at the Analysed_Info DP and beyond. The Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

- collected digit string can be available at the Origination_Attempt DP through ISUP signalling for an SS7 trunk;
- collected digit string can be available at the Collected_Info DP for a party served by a conventional trunk (e.g. non-SS7), ISDN interface using overlap sending, and private facilities.

5) *Feature codes* (e.g. *XX, #) – A vertical service code, such as a "#" or a two-digit or three-digit code preceded by "*" or "11," that precedes any subsequent digit collection (e.g. according to the "normal numbering plan").

Feature codes can be available at the Origination_Attempt DP for a party served by an ISDN interface using *en bloc* sending or through ISUP signalling for an SS7 trunk, and can be available at the Collected_Info DP for non-ISDN lines and private facilities. Since collected

digit strings are not analysed until the Analyse_Information PIC (except to determine if sufficient information has been collected), this criteria could be applicable at the Analysed_Info DP and beyond. The Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Feature codes can be available at the Collected_Info DP for a party served by an ISDN interface using overlap sending.

6) *Prefixes* (e.g. 0+, 00+, 011, 01, 1+) – A string of digits that are not feature codes or access codes and which precede any subsequent digit collection (e.g. according to the "normal numbering plan").

Prefixes can be available at the Origination_Attempt DP for a party served by an ISDN interface using *en bloc* sending, and can be available at the Collected_Info DP for non-ISDN lines, conventional trunks, and private facilities. Since collected prefix information is not analysed until the Analyse_Information PIC (except to determine if sufficient information has been collected), this criteria could be applicable at the Analysed_Info DP and beyond. The Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Prefixes can be available at the Collected_Info DP for a party served by an ISDN interface using overlap sending.

7) *Access codes* (e.g. 8+) for customized numbering plan – A string of digits in a customized numbering plan that matches access codes such as attendant access codes, access codes to escape to the public network, access codes to access a private facility, access codes to access a private network, and feature access codes.

Access codes can be available at the Origination_Attempt DP for a party served by an ISDN interface using *en bloc* sending, and can be available at the Collected_Info DP for non-ISDN lines and private facilities. Since collected access codes are not analysed until the Analyse_Information PIC (except to determine if sufficient information has been collected), this criteria could be applicable at the Analysed_Info DP and beyond. The Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Access codes can be available at the Collected_Info DP for a party served by an ISDN interface using overlap sending.

8) *Specific abbreviated dialling strings* for customized numbering plan – An abbreviated called party number in a customized numbering plan that must match collected address information.

Abbreviated address information can be available at the Origination_Attempt DP for a party served by an ISDN interface using *en bloc* sending, and at the Collected_Info DP for a party served by a non-ISDN line or private facilities. Since collected address information is not analysed until the Analyse_Information PIC (except to determine if sufficient information has been collected), this criteria could be applicable at the Analysed_Info DP and beyond. The Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

Specific abbreviated dialling strings can be available at the Collected_Info DP for a party served by an ISDN interface using overlap sending.

9) *Specific calling party number strings* – A string of digits that must match the calling party number, which is a local, national, or international E.164 number or a number in a customized numbering plan. If a call has been forwarded, the calling party number is the number of the original calling party.

The calling party number is available at the Origination_Attempt DP in the originating BCSM and the Termination_Attempt DP in the terminating BCSM for a call originating from a non-ISDN line, ISDN interface, and can be available at the Origination_Attempt DP and the Termination_Attempt DP for SS7 trunks. This criterion could be applicable at all DPs.

10) *Specific called party number strings* – A string of digits that must match the called party number, which is either a local, national, or international E.164 number, or a number in a customized numbering plan; the latter is not supported by SS7 or conventional trunks. If a call has been forwarded, the called party number is the number of the party that the call is forwarded to.

The called party number can be available at the Origination_Attempt DP for a party served by an ISDN interface using *en bloc* sending or for an SS7 trunk, and can be available at the Collected_Info DP otherwise. Since collected address information is not analysed until the Analyse_Information PIC (except to determine if sufficient information has been collected), this criteria could be applicable at the Analysed_Info DP and beyond, and at all terminating DPs. In the originating BCSM, the Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed. No specific proposals are made for the DPs in the terminating BCSM.

11) *Nature of address* (e.g. Subscriber Significant Number, National Significant Number, International Number) – An indicator of whether the called party number is a private, local (or subscriber), national, or international number.

The nature of address is available at the Analysed_Info DP. This criteria could be applicable at the Analysed_Info DP and beyond. The Analysed_Info DP (mandatory) and all those originating DPs that may be encountered after Analysed_Info (optional) are proposed since not all SSP suppliers may retain this information for the duration of the call/attempt.

12) *Bearer capability* – An indicator of the bearer capability as defined in Recommendation Q.1228

The bearer capability information is available at the Origination_Attempt DP. This criteria could be applicable at all DPs.

13) *Feature activation/indication* (unconditional/conditional on specific feature patterns) – In a local exchange only, a feature activation/indication on an ISDN interface or that is detected at the Mid_Call DP (e.g. "hook-flash," #, etc.) for ISDN and non-ISDN lines that can be sent in conjunction with or preceding other address/digit collection.

A feature activation/indication can be available at all DPs in the originating BCSM for a party served by an ISDN interface, and can be available at the O_Mid_Call DP in the originating BCSM for a party served by a non-ISDN line. A feature activation/indication can be available at the T_No_Answer, T_Answer, T_Mid_Call, T_Suspended, T_Re_Answer, T_Disconnect and T_Abandon DPs in the terminating BCSM for a party served by an ISDN interface, and can be available at the T_Mid_Call DP in the terminating BCSM for a party served by a non-ISDN line. Since collected feature activation information is not acted upon before the Analyse_Information PIC in the originating BCSM and before the T_Alerting PIC in the terminating BCSM, this criterion could be applicable at the Analysed_Info DP and beyond, and at the following terminating DPs: T_No_Answer, T_Answer, T_Mid_Call, T_Suspended, T_Re_Answer, T_Disconnect and T_Abandon.

14) *Facility information* (unconditional/conditional on specific facility information patterns) – A match on the Facility Information Element contained in a signalling message as defined in DSS 1 and ISUP.

Applicable DPs can be determined by mapping signalling messages to the BCSM (see 4.2.2 and Appendix II/Q.1229) and are for further study.

15) *Cause* (unconditional/conditional on specific cause patterns) – A match on the Cause IE contained in a signalling message as defined in DSS 1 and ISUP or an indicator of the cause of specific events of interest. Further study is required to identify the cause values needed as DP criteria for IN CS-2 services from the complete list of cause values specified in Recommendation Q.1228.

Route selection failure information is available at the Route_Select_Failure DP, busy cause information is available at the O_Called_Party_Busy and T_Busy DPs, and release cause information is available at the O_Disconnect, O_Abandon, T_Disconnect and T_Abandon DPs. This criteria is applicable at the identified DPs.

DP criteria assignment to a TDP is dependent on the information available at that TDP and the information available at a TDP is described in 4.2.2.

Tables 4-4 and 4-5 denote applicability of DP criteria to all the DPs.

The entries in the table can be:

- Customer Based;
- Trunk group Based;
- Private Facility Based;
- Office Based.

The DPs in the tables are abbreviated as follows:

OA Origination_Attempt

OAA Origination_Attempt_Authorized

CI Collected_Info

AI Analysed_Info

RSF Route_Select_Failure

OCPB O_Called_Party_Busy

ONA O_No_Answer

OAns O_Answer

OMC O_Mid_Call

OS O_Suspended

ORA O_Re_Answer

OD O_Disconnect

OAb O_Abandon

TA Termination_Attempt

TAA Termination_Attempt_Authorized

TB T_Busy

FSA Facility_Selected_and_Available

TNA T_No_Answer

TAns T_Answer

TMC T_Mid_Call

TS T_Suspended

TRA T_Re_Answer

TD T_Disconnect

Tab T_Abandon

Table 4-4/Q.1224 – Originating DP criteria

	Originating DP												
DP Criteria	**OA**	**OAA**	**CI**	**AI**	**RSF**	**OCPB**	**ONA**	**OAns**	**OMC**	**OS**	**ORA**	**OD**	**OAb**
Trigger Assigned	X	X	X	X	X	X	X	X	X	X	X	X	X
Class of Service	X	O	O	O	O	O	O	O	O	O	O	O	O
Specific Calling Party Number (Note 4)	X	O	O	O	O	O	O	O	O	O	O	O	O
Bearer Capability (Note 5)	X	O	O	O	O	O	O	O	O	O	O	O	O
Specific B-channel Identifier	O	O	O	O	O	O	O	O	O	O	O	O	O
Specific Digit String (Note 1)	–	–	X	X	O	O	O	O	O	O	O	O	O
Feature Code (Note 1)	–	–	X	X	O	O	O	O	O	O	O	O	O
Prefixes (Note 1)	–	–	X	X	O	O	O	O	O	O	O	O	O
Access Codes (Note 1)	–	–	X	X	O	O	O	O	O	O	O	O	O
Called Party Number (Note 1)	–	–	X	X	O	O	O	O	O	O	O	O	O
Specific abbreviated dialling string (Note 1)	–	–	–	X	O	O	O	O	O	O	O	O	O
Nature of Address	–	–	–	X	O	O	O	O	O	O	O	O	O
Feature Activation (Note 3)	–	–	–	X	X	X	X	X	X	X	X	X	X
Facility Information (Note 2)	–	–	–	X	–	–	–	X	X	–	–	–	–
Cause	–	–	–	–	X	X	–	–	–	–	–	X	X
X Aplicable – Not Aplicable O Optional													

Table 4-5/Q.1224 – Terminating DP criteria

	Terminating DP										
DP Criteria	**TA**	**TAA**	**TB**	**FSA**	**TNA**	**Tans**	**TMC**	**TS**	**TRA**	**TD**	**Tab**
Trigger Assigned	X	X	X	X	X	X	X	X	X	X	X
Class of Service	X	O	O	O	O	O	O	O	O	O	O
Specific Calling Party Number (Note 4)	X	O	O	O	O	O	O	O	O	O	O
Bearer Capability (Note 5)	O	O	O	O	O	O	O	O	O	O	O
Specific B-channel Identifier	–	–	–	O	O	O	O	O	O	O	O
Specific Digit String (Note 1)	–	–	–	–	–	–	–	–	–	–	–
Feature Code (Note 1)	–	–	–	–	–	–	–	–	–	–	–
Prefixes (Note 1)	–	–	–	–	–	–	–	–	–	–	–
Access Codes (Note 1)	–	–	–	–	–	–	–	–	–	–	–
Called Party Number (Note 1)	–	–	–	–	–	–	–	–	–	–	–
Specific abbreviated dialling string (Note 1)	–	–	–	–	–	–	–	–	–	–	–
Nature of Address	–	–	–	–	–	–	–	–	–	–	–
Feature Activation (Note 3)	–	–	–	–	X	X	X	X	X	X	X
Facility Information (Note 2)	–	–	–	–	–	X	X	–	–	–	–
Cause	–	–	X	–	–	–	–	–	–	X	X

X Applicable

– Not Aplicable

O Optional

NOTE 1 – Same type of trigger requiring analysis of a specific number of received digits. The analysis can be based on the complete number of received digits or can be based on a predefined number of digits starting from the most significant digit of the received information.

NOTE 2 – A match on the Facility Information Element contained in a signalling message as defined in DSS 1 and ISUP.

NOTE 3 – In a local exchange only. The BCSM has to analyse (if facility is allowed, stored as Class of Service attribute) the received information and has to initiate an IN trigger if required. A feature activation/indication can be available at all DPs in the originating BCSM for a party served by an ISDN interface and can be available at the O_Mid_Call DP in the originating BCSM for a party served by a non-ISDN line. A feature activation/indication can be available at the T_No_Answer, T_Answer, T_Mid_Call, T_Suspended, T_Re_Answer, T_Disconnect and T_Abandon DPs in the terminating BCSM for a party served by an ISDN interface and can be available at the T_Mid_Call DP in the terminating BCSM for a party served by a non-ISDN line.

NOTE 4 – The analysis should not be based on the complete calling party number, it shall be based on a predefined number of digits, starting from the most significant digit of the calling party number.

NOTE 5 – Interpretation of Bearer Capability as optional for all DPs other than Origination_Attempt needs further clarification (e.g. Origination_Attempt DP mandatory means Termination_Attempt DP mandatory). Further, B-channel selection does not appear as a DP-criteria in the table because specific selection of B-channel by the user is for further study: the network can override user selection of B-channel to be used.

If a criteria is marked with an "X" for a Detection Point, then this means that a conditional TDP which is armed at the Detection Point may require the criteria as listed in the table to be satisfied before informing the SCF that the TDP was encountered. For example, a conditional TDP at the Origination_Attempt DP may require the class of service criteria to be satisfied before the SCF is informed that the TDP was encountered.

If a criteria is marked with an "O" for a Detection Point, then this means that it is implementation dependent if the criteria specific information is still present at that DP because not all suppliers may retain this information for the duration of the call/attempt. If the information is still present, the treatment is the same as a criteria marked with an "X".

The trigger item is defined as a single set of DP criteria and the associated information that an SSF/CCF uses to determine if the criteria is met and how to process the trigger. The trigger item consists of trigger type, DP criteria, and the SCF routing information. The trigger items are assigned to users by management process. An SSF should use the SCF routing information to format and route the messages to the appropriate SCF application. The SCF may use existing MTP/SCCP capabilities to route to the SCF.

4.2.7 Trigger types and trigger precedence

Trigger types denote classes of events of interest. They are used to establish trigger precedence rules at TDPs and indicate to the SCF the service logic to be invoked. A non-exhaustive list of trigger types are defined. Implementation of this set of trigger types, or a subset of this set of trigger types, and further network provider defined trigger types are defined by the network operator. This subclause describes the current set of trigger types defined. Definition of further trigger types is for further study. Definition of further interfaces supporting the standard trigger types is for further study. The trigger types given are described in terms of:

1) TDP – The TDP at which the trigger can be detected.
2) DP Criteria – The conditions needed to trigger.
3) Category – Office, group, or subscribed (line-based).
4) Interface – Type of interface to which it can be assigned (e.g. ISDN line).
5) Trigger Type – The value that identifies the type of criteria that caused the SSF/CCF to detect a valid trigger condition at this TDP (i.e. the trigger type).
6) Fault handling – Defines fault handling procedures for the case when the SCF does not respond to the SSF/CCF message. Details on possibilities for fault handling is for further study.

Escape codes apply to the Off-Hook delay and Channel Setup PRI triggers. These codes provide an escape so that a subscriber to these triggers can still make certain calls (e.g. Emergency 911) when the SCF or the link to the SCF is down. Each SSF/CCF will have an administered list of escape codes. These codes are valid numbers as per the number plan, prefixed numbers (e.g. 0-, 00-), or an Emergency Service Code. The SSF/CCF provides the same escape treatment to calls in the following situations:

- A call from a line or trunk using the numbering plan in force corresponds to a number on the administered escape code list.
- A call from a line or trunk using the numbering plan in force corresponds to a prefix and number on the escape code list.
- A call from a line or trunk using the numbering plan in force corresponds to a feature code plus a number on the escape code list.

- A call from a line or trunk using a private numbering plan corresponds to an access code that results in the call being routed over the public network plus a number on the escape code list.

Network administrators should be aware that if a number is on the escape code list, then variations of that number that may occur because of different user dialling procedures may also need to be entered on the escape code list so that they also result in escape code treatment. If the SSF/CCF determines that an escape code applies, then the Off-Hook Delay trigger is not detected, a message is not sent to the SCF from the Collected_Information TDP, and the SSF/CCF continues with normal call processing at the Analyse_Information PIC. Triggers may be detected at subsequent TDPs.

After the BRI feature activator, public feature code, or customized numbering plan trigger is encountered, the SSF/CCF supports subsequent digit collection. Subsequent digit collection rules are the same for all subscribers to a particular numbering plan. This subsequent digit collection is specified by the numbering plan in force and, thus, is assumed to occur during the Collecting_Information PIC. Any subsequent digits are included in a CollectedInformation query if an Off_Hook_Delay trigger is hit.

The SSF/CCF supports the administration of the following rules for subsequent digit collection for a numbering plan, not an individual subscriber trigger. Each BRI feature activator, public feature code, or access code is administrable to do one of the following:

1) Do not collect subsequent digits.
2) Collect subsequent digits according to the normal numbering plan for that line or trunk. Second dial tone is applied after the feature code or indicator is dialled. A restart clears the digits dialled after the feature code, but does not clear the feature code itself. If the digits dialled are incomplete or invalid, the SSF/CCF shall not query the SCF, but will provide final treatment.
3) Collect a variable number of digits (0 to 32). Digit collection is complete when the caller enters a "#" to indicate end of dialling, or when the normal interdigit timing interval expires.

As the same DP may be armed multiple times as a TDP-R, precedence rules for trigger processing are specified as:

1) For ISDN BRI lines, the same trigger may be assigned to the ISDN line interface or the ISDN Service Profile. These may be provisioned with different SCFs as the destination. The sequence of processing shall be Service Profile first and ISDN line last.
2) Subscribed triggers have precedence over group triggers.
3) Group triggers have precedence over office triggers.

The sequence for processing IN CS-2 triggers should be as detailed in Table 4-6. Table 4-6 reflects the above trigger precedence rules, giving higher precedence triggers first within the TDPs. Precedence rules for network operator defined trigger types are for further study.

Table 4-6/Q.1224 – IN CS-2 trigger precedence

Trigger detection point	Trigger type
Collected_Information	Off_Hook_Delay
	Channel_Setup_PRI
	Shared_Interoffice_Trunk
Analysed_Information	BRI_Feature_Activation_Indicator
	Public_Feature_Code
	Specific_Feature_Code
	Customized_Dialling_Plan
	Specific_Digit_String
	Emergency Service

4.2.7.1 Off_Hook_Immediate

The SSF/CCF detects the offHookImmediate trigger when an origination indication from the interface is detected. The Origination_Attempt TDP has been reached.

1) TDP: Origination_Attempt.
2) DP Criteria: Trigger Assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type.
5) Trigger Type: offHookImmediate.
6) Fault Handling: Final treatment (other treatment such as Default Routing or Continue Call Processing are for further study).

4.2.7.2 Origination_Attempt_Authorized

The SSF/CCF detects the originationAttemptAuthorized trigger when the authority to place an outgoing call has been verified. The OriginationAttempt Authorized_TDP has been reached.

1) TDP: Origination_Attempt_Authorized.
2) DP Criteria: Trigger Assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type.
5) Trigger Type: originationAttemptAuthorized.
6) Fault Handling: Final treatment (other treatment such as Default Routing or Continue Call Processing are for further study).

4.2.7.3 Off-Hook_Delay

The Off-Hook_Delay trigger type denotes an event class on the originating interface such that the SSF/CCF receives enough information to process the call, information received does not violate the numbering plan in force, and escape codes/switch-based feature codes that prevent the Off-Hook_Delay trigger from being detected have not been entered. The Collected_Information TDP has been reached. This trigger occurs for all calls, but only on call origination. For example, the Off-Hook_Delay trigger type may be used in providing a feature to request user PIN prior to authorization of routing the call, after dialling is complete.

1) TDP: Collected_Information.

2) DP criteria: Trigger assigned (unconditional).

3) Category: Subscribed.

4) Trigger assigned to: non-ISDN line, BRI Service Profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type.

5) Trigger Type: offHookDelay.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.4 Channel_Setup_PRI

The Channel_Setup_PRI trigger type denotes a call attempt on a specific B-channel on a PRI interface. It is detected when SSF/CCF receives enough information to process the call, information received does not violate the numbering plan in force, and escape codes/switch-based feature codes that prevent the Channel_Setup_PRI trigger from being detected have not been entered. This trigger occurs for all calls that use identified B-channels. For example, if a B-channel is dedicated to a specific service (e.g. directory assistance), this trigger could be used in providing this service on the B-channel.

1) TDP: Collected_Information.

2) DP criteria: Trigger assigned (conditional), Specific B-channel identifier.

3) Category: Subscribed.

4) Trigger assigned to: Dedicated B-channel on a PRI interface.

5) Trigger Type: channelSetupPRI.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.5 Shared_Interoffice_Trunk

The Shared_Interoffice_Trunk trigger type denotes a class of events in which a SSF/CCF performs trigger processing for a CCF. It is detected for calls routed to an SSF/CCF from a CCF: an assist procedure is to be invoked. The SSF/CCF supports a shared interoffice trunk trigger for the interface to a CCF. On this interface, SS7 or conventional signalling used by the CCF to route the call is used by the SSF/CCF for triggering a query to the SCF. For example, this trigger may be used in providing a feature to supply specific routing of a call when an assist has been invoked.

1) TDP: Collected_Information.

2) DP criteria: Trigger assigned (unconditional).

3) Category: Subscribed (on trunk basis).

4) Trigger assigned to: Public trunk groups.

5) Trigger Type: sharedIOTrunk.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.6 BRI_Feature_Activation_Indicator

The BRI_Feature_Activation_Indicator trigger type is detected when the SSF/CCF detects a feature activation indicator. If the SSF/CCF receives a feature activator for a switch-based feature with or without dialled digits on the ISDN BRI line, it is desirable that the call escape the Off-Hook Delay trigger even if the other requirements for the trigger are met. This trigger is used for BRI features.

1) TDP: Analysed_Information, O_Mid_Call, T_Mid_Call.

2) DP criteria: Trigger assigned (conditional), BRI feature activation/indication.

3) Category: Subscribed.

4) Trigger assigned to: BRI Service Profile.

5) Trigger Type: featureActivator.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.7 Public_Feature_Code

The Public_Feature_Code trigger type is detected when any of the SSF/CCF supported feature codes (e.g. *XX) are dialled. Although several feature codes may cause the trigger to be detected, the SSF/CCF shall not differentiate between them for subscription purposes (e.g. *46, *53, *58 all cause the trigger to be detected). It is desirable that calls with a feature code that correspond to the activation, deactivation, or access to a switch-based feature with or without dialled digits for a call escape the Off-Hook Delay trigger even if the other requirements for the trigger are met. For example, this trigger may be used within a service providers network to advertise other capabilities to the user when an existing feature code is used.

1) TDP: Analysed_Information.

2) DP criteria: Trigger assigned (conditional), Feature Code (unconditional).

3) Category: Subscribed.

4) Trigger assigned to: non-ISDN line, BRI Service Profile, BRI interface, Directory Number and Call Type.

5) Trigger Type: verticalServiceCode.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.8 Specific_Feature_Code

The Specific_Feature_Code trigger type is detected when the SSF/CCF analyses a specific feature code which is administered as a DP criteria (e.g. only *46 causes the trigger to be detected). It is desirable that calls with a feature code that correspond to the activation, deactivation, or access to a switch-based feature with or without dialled digits for a call escape the Off-Hook Delay trigger even if the other requirements for the trigger are met. Invocation of call forward can be supported by this trigger type.

1) TDP: Analysed_Information.

2) DP criteria: Trigger assigned (conditional), feature code on specific feature pattern.

3) Category: Subscribed.

4) Trigger assigned to: non-ISDN line, BRI Service Profile, BRI interface, Directory Number and Call Type.

5) Trigger Type: specificFeatureCode.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.9 Customized_Dialling_Plan

The Customized_Dialling_Plan trigger type is detected when an access code is dialled within a customized numbering plan. For example, 8+ or a one to seven-digit intercom code can be defined as

a trigger to query the SCF. The customized numbering plan specifies, for each code, whether the SSF/CCF performs some manipulation of the dialled digits (e.g. digit insertion, deletion, or translation to public numbers), or queries the SCF with the digits as dialled. It is desirable that calls from users of a customized numbering plan with a feature access code that corresponds to the activation, deactivation, or access to a switch-based feature with or without dialled digits escapes the Off-Hook Delay trigger even if the other requirements for the trigger are met. For example, this trigger type could be used in providing 5-digit dialling through the public network for a virtual private network.

1) TDP: Analysed_Information.
2) DP criteria: Trigger assigned (conditional), Access Code or specific abbreviated dialling string for customized numbering plan.
3) Category: Group.
4) Trigger assigned to: All lines and trunks assigned to a customized numbering plan or business group.
5) Trigger Type: customizedAccess customizedIntercom.
6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.10 Specific_Digit_String

The Specific_Digit_String trigger type is detected when the appropriate sequence of digits is dialled according to the numbering plan in use. For example, a 3-, 6-, or 10-digit sequence of digits can be provisioned as the trigger. Trigger provisioning specifies whether the SSF/CCF performs some manipulation of the dialled digits (e.g. digit insertion, deletion, or translation to public numbers), or queries the SCF with the digits as dialled. The numbering plan in force should ensure that emergency service numbers are distinct from provisionable specific digit strings. Precedence should be specified (e.g. most to least specific): for further study. For example, this trigger could be used to provide for customized call routing on a specific directory number.

1) TDP: Analysed_Information.
2) DP criteria: Trigger assigned (conditional), Specific called party number string.
3) Category: Office.
4) Trigger assigned to: All lines and trunks assigned to a public office numbering plan or the entire SSF/CCF (i.e. not every number may have facilities associated with it).
5) Trigger Type: As per numbering plan (e.g. E.164, National, private numbering plans).
6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.11 Emergency Service

The Emergency_Service trigger type is detected when a digit string denoting emergency service is dialled. The SSF/CCF detects the Emergency Service trigger on any call with access to the public office numbering plan when a designated Emergency Service (e.g. 911) number is dialled. This trigger can provide for emergency call handling under control of the SCF.

1) TDP: Analysed_Information.
2) DP criteria: Trigger assigned (conditional), Specific called party number string.
3) Category: Office.
4) Trigger assigned to: All lines and trunks assigned to a public office numbering plan or the entire SSF/CCF.

5) Trigger Type: emergencyService.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.12 AFR

The AFR trigger type denotes exhaustion of a list of selected routes within the SSF/CCF. This is used to specify route choices for both public and private network calls. A specific list of routes is assigned to each user. The SSF/CCF detects the Automatic Flexible Routing (AFR) trigger when the list of routes is exhausted (i.e. all routes are busy or unavailable). For example, this trigger type could be used in providing a feature to play a customized announcement when all routes are busy.

1) TDP: Route_Select_Failure.

2) DP criteria: Trigger assigned (conditional), DP criteria for further study.

3) Category: Group.

4) Trigger assigned to: All lines and trunks with access to the AFR pattern.

5) Trigger Type: AFR.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.13 O_Called_Party_Busy

The SSF/CCF detects the O_Called_Party_Busy trigger when the originating call portion receives a report of user busy from the terminating call portion. For example, this trigger may be used to supply an automatic ring again if the called party is busy.

1) TDP: O_Called_Party_Busy.

2) DP criteria: Trigger assigned (conditional), DP criteria for further study.

3) Category: Subscribed Office.

4) Trigger assigned to: Subscribed-non-ISDN line, BRI Service Profile and Call Type, BRI/PRI interface; Office-non-ISDN line, ISDN interfaces, Directory Number and Call Type.

5) Trigger Type: oCalledPartyBusy.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.14 O_No_Answer

The SSF/CCF detects the O_No_Answer trigger when the application timer associated with the O_No_Answer event expires: a time-out associated with the originating portion of the call occurs. For example, this trigger type may be used to provide automatic alternate routing when the timeout occurs.

1) TDP: O_No_Answer.

2) DP criteria: Trigger assigned (conditional), DP criteria for further study.

3) Category: Subscribed, Office.

4) Trigger assigned to: Subscribed-non-ISDN line, BRI Service Profile, BRI/PRI interface; Office-non-ISDN line, ISDN interfaces, Directory Number and Call Type.

5) Trigger Type: oNoAnswer.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.15 O_Answer

The SSF/CCF detects the oAnswer trigger when answer indication from the terminating BCSM is received.

1) TDP: O_Answer.
2) DP Criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group.
5) Trigger Type: oAnswer.
6) Fault Handling: Call continued (other treatments such as final treatment or default routing are for further study).

4.2.7.16 O_Disconnect

The SSF/CCF detects the oDisconnect trigger when either the call is cleared from the terminating BCSM or the originating facility disconnects.

1) TDP: O_Disconnect.
2) DP Criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group.
5) Trigger Type: oDisconnect.
6) Fault Handling: Final treatment if clearing from the terminating BCSM (other treatments such as default routing or continue call processing are for further study), continue call processing if the originating facility disconnects.

4.2.7.17 O_Switch_Hook_Flash_Immediate

The O_Switch_Hook_Flash_Immediate trigger is detected in the originating BCSM when the SSF/CCF detects a switch hook flash indication from a non-ISDN line. This trigger may be used to initiate an IN-supported Call Party Handling feature. Note the unconditional nature of the O_Switch_Hook_Flash_Immediate trigger.

1) TDP: O_Mid_Call.
2) DP criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line.
5) Trigger type: oSwitchHookFlashImmediate.
6) Fault handling: No administrable fault handling (continue call processing).

4.2.7.18 O_Switch_Hook_Flash_Specified_Code

The SSF/CCF detects the O_Switch_Hook_Flash_Specified_Code trigger in the originating BCSM after it receives a switch hook flash indication from a non-ISDN line, applies second dial tone to the non-ISDN line, and collects a feature code (e.g. *XX) or an access code (e.g. 8+) that is designated as the criteria for this trigger at the SSF/CCF. This trigger may be used to initiate an IN-supported Call Party Handling feature. Note the conditional nature of the O_Switch_Hook_Flash_Specified_Code trigger.

1) TDP: O_Mid_Call.

2) DP criteria: Trigger assigned (conditional), feature code or access code.

3) Category: Subscribed.

4) Trigger assigned to: non-ISDN line.

5) Trigger type: oSwitchHookFlashSpecifiedCode.

6) Fault handling: No administrable fault handling (continue call processing).

4.2.7.19 Termination_Attempt

The SSF/CCF detects the Termination_Attempt trigger when the call directed to the DN/CT reaches the Termination_Attempt DP.

1) TDP: Termination_Attempt.

2) DP Criteria: Trigger Assigned (unconditional).

3) Category: Subscribed.

4) Trigger assigned to: Directory Number and Call Type (equipped or unequipped).

5) Trigger Type: terminationAttempt.

6) Fault Handling: Final treatment (other treatment such as Default Routing or Continue Call Processing are for further study).

4.2.7.20 Term_Attempt_Authorized

The SSF/CCF detects the termAttemptAuthorized trigger when the terminating authorization checks have been completed successfully.

1) TDP: Term_Attempt_Authorized.

2) DP Criteria: Trigger assigned (unconditional).

3) Category: Subscribed.

4) Trigger assigned to: non-ISDN line, Directory Number and Call Type, BRI/PRI interface.

5) Trigger Type: termAttemptAuthorized.

6) Fault Handling: Final treatment (other treatment such as Default Routing or Continue Call Processing are for further study).

4.2.7.21 T_Busy

The SSF/CCF detects the T_Busy trigger when it determines that the terminating access is busy (i.e. network determined user-busy). For example, this trigger may be used to forward the call to another number (based on time of day) if the terminating access is busy.

1) TDP: T_Busy.

2) DP criteria: Trigger assigned (unconditional).

3) Category: Subscribed.

4) Trigger assigned to: non-ISDN line, Directory Number and Call Type, BRI/PRI interface.

5) Trigger Type: tBusy.

6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.22 Term_Resource_Available

The SSF/CCF detects the Term_Resource_Available trigger when it determines that the terminating access is not busy [i.e. an idle facility (e.g. B-channel, call appearance, or trunk) could be found].

The conditions that result in the detection of terminating resource available depend on the type of terminating access and subscribed services.

1) TDP: Facility_Selected_and_Available.
2) DP Criteria: Trigger Assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group, Directory Number and Call Type.
5) Trigger Type: termResourceAvailable.
6) Fault Handling: Final treatment (other treatment such as Default Routing or Continue Call Processing are for further study).

4.2.7.23 T_No_Answer

The SSF/CCF detects the T_No_Answer trigger when the application_timer associated with the T_No_Answer event expires: a time-out associated with the terminating portion of the call occurs. For example, this trigger type can be used to automatically reroute the call to a voice mail system.

1) TDP: T_No_Answer.
2) DP criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, Directory Number and Call Type, BRI/PRI interface.
5) Trigger Type: tNoAnswer.
6) Fault handling: Final Treatment (other treatments such as Default Routing or Continue Call Processing are for further study).

4.2.7.24 T_Answer

The SSF/CCF detects the tAnswer trigger when it detects an answer indication from the terminating facility.

1) TDP: T_Answer.
2) DP Criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group.
5) Trigger Type: tAnswer.
6) Fault Handling: Call continued (other treatments such as final treatment or default routing are for further study).

4.2.7.25 T_Disconnect

The SSF/CCF detects the tDisconnect trigger when either the call is cleared from the originating BCSM or the terminating facility disconnects.

1) TDP: T_Disconnect.
2) DP Criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line, BRI service profile, BRI interface, PRI interface, private facility trunk group.
5) Trigger Type: tDisconnect.

6) Fault Handling: Final treatment if clearing from the terminating facility (other treatments such as default routing or continue call processing are for further study), continue call processing if the originating BCSM clears.

4.2.7.26 T_Switch_Hook_Flash_Immediate

The T_Switch_Hook_Flash_Immediate trigger is detected in the terminating BCSM when the SSF/CCF detects a switch hook flash indication from a non-ISDN line. This trigger may be used to initiate an IN-supported Call Party Handling feature. Note the unconditional nature of the T_Switch_Hook_Flash_Immediate trigger.

1) TDP: T_Mid_Call.
2) DP criteria: Trigger assigned (unconditional).
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line.
5) Trigger type: tSwitchHookFlashImmediate.
6) Fault handling: No administrable fault handling (continue call processing).

4.2.7.27 T_Switch_Hook_Flash_Specified_Code

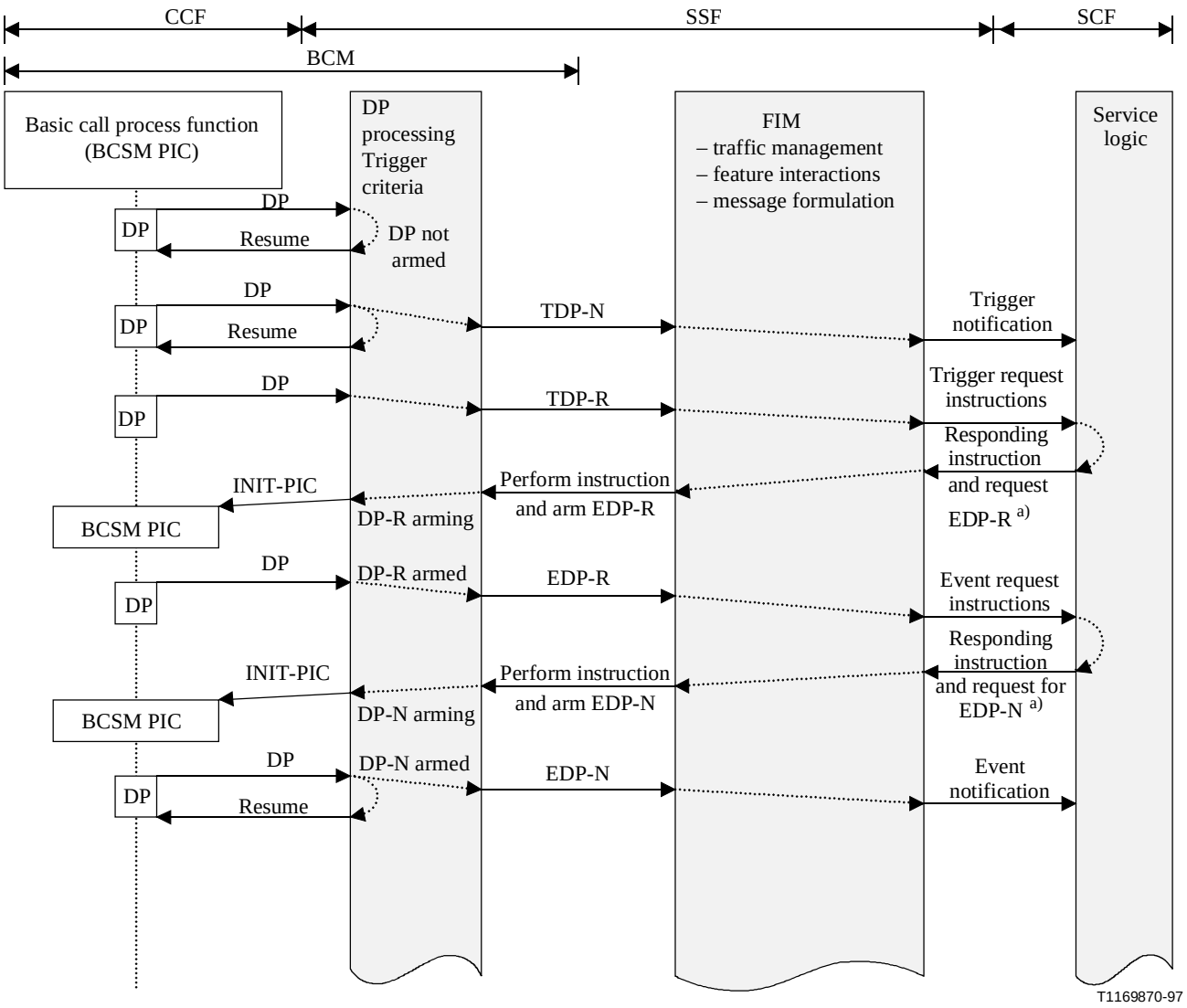
The SSF/CCF detects the T_Switch_Hook_Flash_Specified_Code trigger in the terminating BCSM after it receives a switch hook flash indication from a non-ISDN line, applies second dial tone to the non-ISDN line, and collects a feature code (e.g. *XX) or an access code (e.g. 8+) that is designated as the criteria for this trigger at the SSF/CCF. This trigger may be used to initiate an IN-supported Call Party Handling feature. Note the conditional nature of the T_Switch_Hook_Flash_Specified_Code trigger.

1) TDP: T_Mid_Call.
2) DP criteria: Trigger assigned (conditional), feature code or access code.
3) Category: Subscribed.
4) Trigger assigned to: non-ISDN line.
5) Trigger type: tSwitchHookFlashSpecifiedCode.
6) Fault handling: No administrable fault handling (continue call processing).

4.2.8 DP processing

DP processing involves:

– traffic management actions (see call gapping and service filtering information flows in clauses 11 and 12);
– determining if DP criteria are met (see 4.2.6 and this subclause);
– handling service logic instance interactions when invoking new instances of IN and non-IN service logic (see this subclause and 4.4.3);
– and formulating information flows to send to one or more SCFs (see this subclause and Initial DP and event report information flows in clauses 11 and 12).



DP Detection Point
TDP Trigger Detection Point
EDP Event Detection Point
R/N Request/notification
PIC Point in Call

a) In this example, the responding instruction and request for EDP are shown together. These are independent information flows and may not be sent together in all cases.

Figure 4-10/Q.1224 – DP processing for each DP type

Since a DP may be armed as a TDP and/or an EDP for the same call segment association, the BCM should apply the following set of rules during DP criteria processing to ensure single point of control:

Rule 1 – At any DP, a specific trigger condition can only trigger one service logic program instance (SLPI) at a time.

Rule 2 – At any DP, processing of notifications – EDP-N and TDP-N – has higher priority than processing of requests – EDP-R and TDP-R. If several notifications exist, EDP-R and TDP-R are processed when all notifications have been processed.

Rule 3 – If a DP is both armed as EDP and TDP, then the EDP processing has higher priority than the TDP processing since the EDP has been armed in an already existing SSF-SCF relationship.

Rule 4 – If a DP is both armed as EDP-R and TDP-R, the EDP-R is first processed and, if the control relationship is terminated as a result of the EDP-R processing, processing of the TDP-R is allowed.

The rules are listed in descending priority order. They are illustrated with the diagram in Figure 4-11.

A control relationship persists as long as there is ≥ 1 EDP-R armed for at least one call segment of a call segment association. A control relationship terminates if there are no more EDPs armed or the call clears. During a control relationship, EDPs may be dynamically disarmed by the SCF, or are disarmed by the SSF as they are encountered and reported to the SCF, or when the call clears.

Single point of control ensures that only one service logic exists within the control relationship.

Single point of control is only guaranteed within a Call Segment Association.

A control relationship changes to a monitor relationship if there are no more EDP-Rs armed and 1 EDP-N armed. A monitor relationship terminates if there are no more EDP-Ns armed or the call clears. During a monitor relationship, EDP-Ns are disarmed by the SSF as they are encountered and reported to the SCF, or when the call clears.

TDP-N criteria may be processed whether or not there is an existing control relationship for the same portion of the call, since a TDP-N does not open a control relationship. This procedure has no effect on the existing control relationship.

NOTE – It is possible to cause infinite retriggering during the processing of a call. Such an example is as follows:

1) User dials 555-1111.
2) Analysed_Information DP is armed as TDP-R, and the criteria is 555-1111.
3) SCF returns a Destination Routing Address of 555-1111 in the Connect operation.
4) The call will resume at the Analyse_Information PIC and then trigger at the Analysed_Information DP again. This effectively produces an infinite loop.

It should be noted that this loop can occur any time the response from a DP returns the call to a previous point in the BCSM. Recommendation Q.1229 provides details as to how retriggering can be avoided.

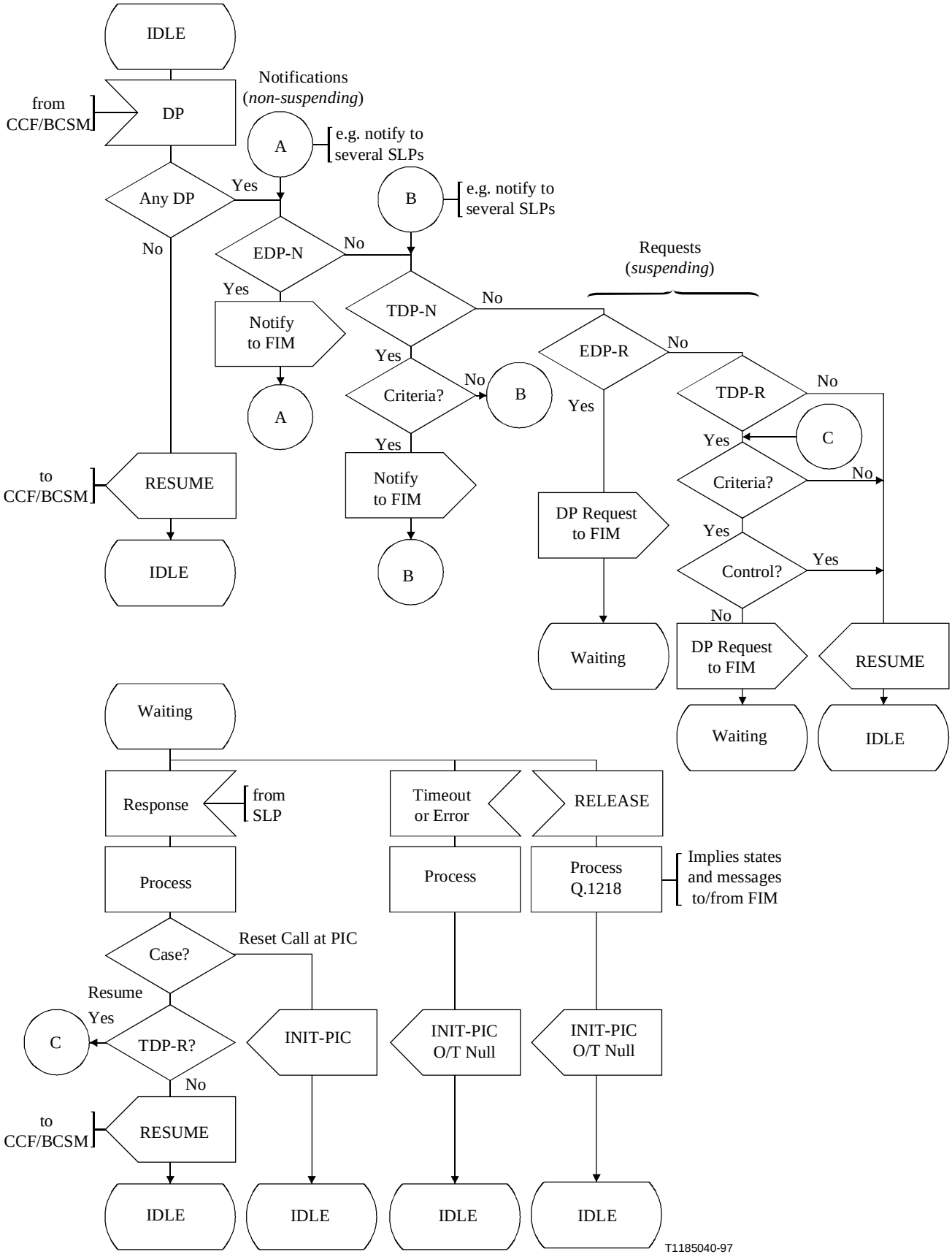


Figure 4-11/Q.1224 – Detection point processing

As a consequence of these rules, the BCM should support a number of TDP/EDP processing combinations to ensure single point of control. These combinations are identified in Table 4-7, along with three error combinations that should not occur.

These combinations applies only within one call segment association.

Table 4-7/Q.1224

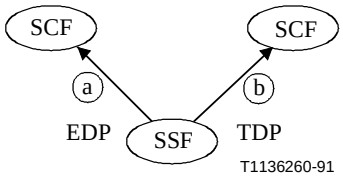
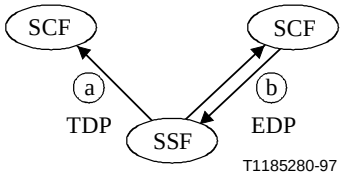
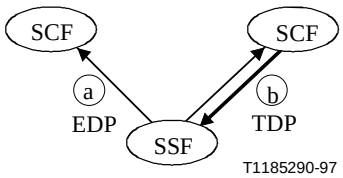
Scenario	TDP type	EDP type	Existing relationship	Processing
1	Not armed	Not armed	Don’t care	Continue
2	TDP-R	Not armed	No	Initiating DP request
3.a	TDP-R	Not armed	Control	Continue (ignore TDP)
3.b	TDP-R	Not armed	Monitor	Initiating DP request
4	TDP-N	Not armed	Don’t care	One-way DP notification Initiate Monitor relationship
5.a	Not armed	EDP-R	Control	Subsequent DP request, if armed EDPs remaining, or terminating DP request, if last armed EDP
5.b	Not armed	EDP-R	Monitor	Error case – Continue (ignore EDP)
6	Not armed	EDP-N	Control or Monitor	Subsequent DP notification, if armed EDPs remaining, or terminating DP notification, if last armed EDP
7	Not armed	EDP-R/N	No	Error case – Continue (ignore EDP)
8	TDP-N	EDP-N	Control or monitor	Process a and b: a) For EDP, process as scenario 6 b) For TDP, one-way DP notification, process as scenario 4 
9	TDP-N	EDP-R/N	No	Error error case – Ignore EDP and process TDP as scenario 4
10.a	TDP-N	EDP-R	Control	Process a and b: a) For TDP, one-way DP notification, process as scenario 4 b) For EDP, subsequent DP request, process as scenario 5.a 
10.b	TDP-N	EDP-R	Monitor	Error case – Ignore EDP and process TDP as scenario 4

Table 4-7/Q.1224 *(concluded)*

Scenario	TDP type	EDP type	Existing relationship	Processing
11.a	TDP-R	EDP-N	Control	Process a and b: a) For EDP, subsequent DP notification, process as scenario 6 b) Ignore TDP
11.b	TDP-R	EDP-N	Monitor	Process a and b: a) For EDP, process as scenario 6 b) For TDP, initiating DP request, process as scenario 3.b 
12	TDP-R	EDP-R/N	No	Error case – Ignore EDP and process TDP as scenario 2
13.a	TDP-R	EDP-R	Control	Process EDP as scenario 5.a. If this EDP was the last of the previously established control relationship, process the TDP afterwards. Otherwise the TDP is ignored
13.b	TDP-R	EDP-R	Monitor	Error case – Ignore EDP and process TDP as scenario 3.b

4.2.9 Out-Channel Call Related User Interaction (OCCRUI)

Description of the OCCRUI at the "Call Model" level

A generic transport mechanism (transparent at the SSF level) for the exchange of information between the User and the Service Logic is supported based on new IEs, respectively STUI in the "SCF-to-User" direction and UTSI in the "User to SCF" direction. These IEs are composed of two sub-IEs which are USIServiceIndicator and USIInformation. The first IE identifies the IN Service Logic invoked while the second carries useful information between the User and the Service Logic.

In the "SCF-to-User" direction, once the SSF receives an STUI IE from the SCF within the sendSTUI IF, it forwards it to the appropriate ISDN User. This User is clearly and easily defined by the Call Reference (indicated by the down-lower protocols) and the legID parameter (indicated by the INAP protocol).

In the "User-to-SCF" direction, the OCCRUI mechanism seems to be more complicated. In order to define how the SSF/CCF decides whether it forwards the OCCRUI information it receives from a User to the succeeding exchange or it passes it to a specific IN service, it seems necessary to distinguish two cases:

1) The OCCRUI information is considered as a "notification event" previously requested by the SCF. In this case, there is already a SSF-SCF relationship.

2) The OCCRUI information is an additional information. In this case, it is only an optional information.

The SSF addresses the appropriate SCF thanks to the *ServiceIndicator* parameter it receives within the UTSI IE. The *ServiceIndicator* value is indicated by the ISDN User or by the Service Logic:

– If the SCF has initiated the UTSI/STUI dialogue, the ISDN User sets the *ServiceIndicator* value of the UTSI IE to the *ServiceIndicator* value of the initial STUI IE. This scenario corresponds to the first case.

– If the ISDN user sends an UTSI without having previously received an STUI IE from the SCF, then it initializes the *ServiceIndicator* value of the UTSI IE with an predefined value. This scenario corresponds to the second case.

4.2.9.1 First case: the OCCRUI information is considered as a "notification previously requested by the SCF"

In this case, the SSF communicates with the SCF during an already existing SCF-SSF relationship; the SCF initiates the "Out-Channel" dialogue with the User sending an STUI IE, within the sendSTUI IF. Both SCF and SSF behaviour can be described as above:

– Thanks to an information flow, independently from the BCSM processing, one SCF tells the SSF with the requestReportUTSI IF to report to it all the UTSI Information Elements with a given *ServiceIndicator* value.

– Then, once it receives an UTSI IE, the CCF/SSF compares the *ServiceIndicator* value of this IE with the previously indicated *ServiceIndicator* value. If they coincide, then the SSF reports the UTSI IE to the SCF with the reportUTSI IF.

In this case, the required *ServiceIndicator* value is explicitly indicated by the SCF and stored at the SSF level; this data is in a table associated with the OCCRUI FSM.

NOTE – The STUI/UTSI IE is conveyed in the appropriate signalling message (Basic Call Control signalling message or Facility message) depending on the phase of the call on the dedicated leg (e.g. on the BCSM processing).

4.2.9.2 Second scenario: the OCCRUI information is an additional and optional information

The OCCRUI information does not impact the usual DP processing. The UTSI IE is only an optional parameter that the SSF introduces in the following information flows after analysing the corresponding *ServiceIndicator* value to make sure that the received UTSI IE is targeted to this particular Service Logic.

– InitialDP.

– DP Specific Information Flows.

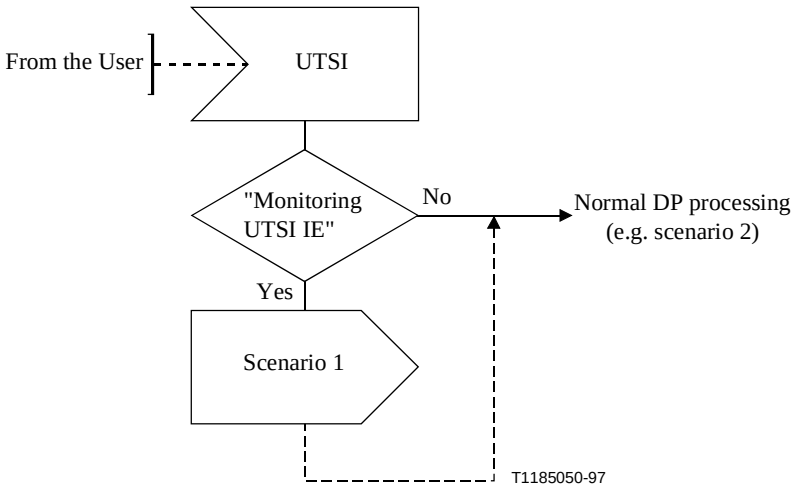
The SSF checks if the *ServiceIndicator* value of the UTSI coincides with the *ServiceIndicator* value contained within the Service related data table.

NOTE – The STUI/UTSI IE is conveyed in the appropriate signalling message (Basic Call Control signalling message or Facility message) depending on the phase of the call on the dedicated leg (e.g. on the BCSM processing).

4.2.9.3 Synthesis

The table and the SDL diagram beyond summarize the proposal:

Scenario	OCCRUI FSM	Service Indicator indicated by the ISDN User	Service Indicator value of reference	IF sent to the SCF
STUI/UTSI dialogue	Monitoring UTSI IE	Indicated in the STUI IE *Dynamic*	Stored in the data table associated with the OCCRUI FSM	Special IF to define
UTSI = additional information	Idle	Predefined	Explicitly indicated in the IN related data table of the IN service *Static in the SSF*	*Initial DP* IF *DP Specific* IF



4.3 IN-Switching Manager (IN-SM)

A brief description of the IN-SM is provided in 4.1. The IN-SM centres around the IN-Switching State Model (IN-SSM) which provides a description of SSF/CCF IN call/connection processing in terms of IN call/connection states. Object-oriented techniques are used to describe the IN-SSM, based on the concepts and principles outlined in Annex B/Q.1204.

The IN CS-2 IN-SM supports IN call party handling capabilities. Four core SSF/CCF capabilities are required to provide these capabilities. Once these four core capabilities are in place, further enhancements can be made to SSF/CCF processing to allow for call party handling within the SSF/CCF (e.g. the SSF/CCF Connection View State approach). Also, once these four capabilities are in place, then call party handling can be performed by other means than SSF/CCF processing (e.g. a bridging resource). Additionally, it is possible to combine aspects of internal SSF/CCF processing along with a bridging resource to provide a hybrid approach to call party handling. Therefore, for IN CS-2 the IN-SM provides flexibility in performing call party handling.

The IN-SM subjects described in the following subclauses include the IN-SM core call party handling capabilities, IN-SSM, IN-SSM events that can be reported to active IN service logic instances, and SSF resource control. A high-level description of these subjects is provided.

4.3.1 IN-Switching State Model (IN-SSM)

The IN-SSM provides an object-oriented finite state machine description of SSF/CCF IN call/connection processing in terms of IN call/connection states. It provides a framework for describing the scope of view and control of SSF/CCF activities offered to an SCF. The extent to which the IN-SSM is visible to the SCF is defined by the information flows identified for IN CS-2 between the SSF/CCF and SCF. See subclause 4.3.3.7, Connection View States (CVSs) and Possible Transitions.

IN call/connection states can be described in terms of the IN-SSM, which defines the set of SSF/CCF objects visible to the SCF. Each IN-SSM instance provides the SCF with a limited aperture of visibility and influence into SSF/CCF IN call/connection processing. This aperture of visibility and influence is defined by the objects that constitute the IN-SSM. These objects are abstractions of SSF/CCF resources accessible to the SCF.

There can be various types of IN-SSMs, each type defined by the objects that constitute it. For example, a Call Segment Association IN-SSM would contain objects that are abstractions of switching and transmission resources. This subclause focuses on such a Call Segment Association IN-SSM, though it is recognized that other types of IN-SSMs may exist for accessing other types of resources.

There can also be various subtypes of a particular IN-SSM type, each defined by a subset of, or restriction on the use of, the total set of objects in the IN-SSM type. It is anticipated that IN-SSM subtypes will be identified to align with specific IN capability sets as they are defined.

A Call Segment Association (CSA) instance is created when an IN service logic instance is invoked that requires IN connection control. It is either created as a result of encountering a TDP in a BCSM that satisfies DP criteria, or is initiated by the SCF independent of encountering TDPs. A CSA instance is destroyed when the SCF informs the SSF that the IN service logic instance is completed or the CSA should be destroyed. The SSF can also initiate CSA destruction (e.g. during error or abnormal conditions).

Within the SDL models (Annex A/Q.1228) the Call Segment Association instance is already created prior to the encountering of the TDP or SCF initiated call. This is part of the SDL modelling to enable proper receipt of the TDP events by the CSA object. This restriction is only for the purpose of SDL modelling and is not intended to be standardized.

Figure 4-12 provides an example of a CSA instance. It illustrates two classes of objects that have been identified: legs and connection points. A leg is a representation of a communication path towards an addressable network entity, as viewed from the CSA. A connection point is a representation of the interconnection of legs, as viewed from the CSA, that allows information to flow between legs. It should be noted that the fundamental processes that establish communication paths, and maintain connections between them, are the basic call processes modelled by one or more BCSMs. As such, the CSA objects reflect both connectivity information (e.g. the relation of legs and connection points to each other) and call processing information (e.g. BCSM events and basic call-related information), which can be used by an instance of IN service logic to influence the connectivity and call processing aspects of a call.

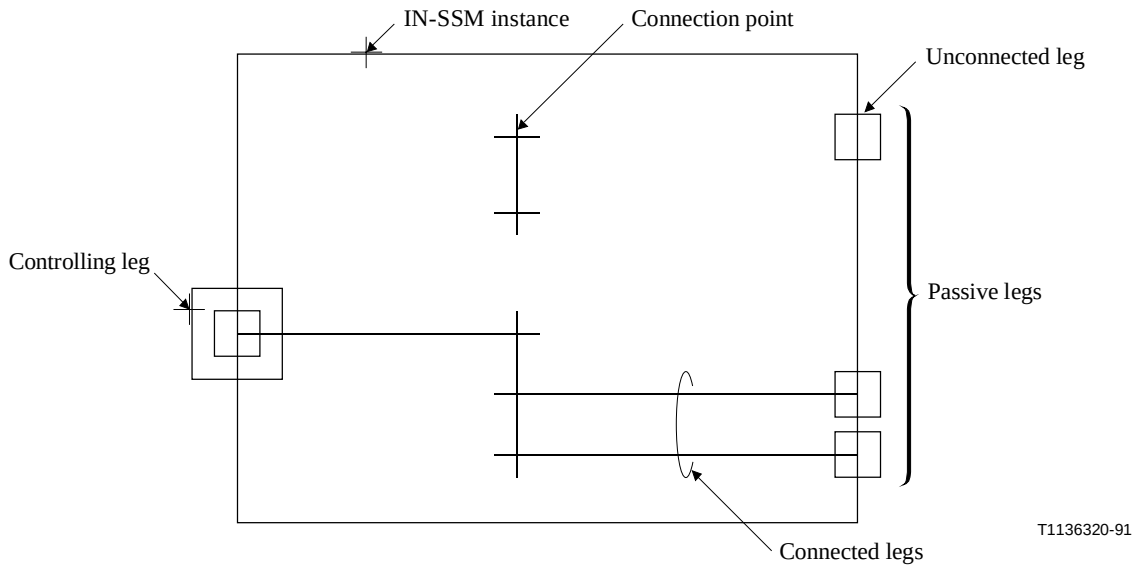


Figure 4-12/Q.1224 – Call Segment Association instance

The attributes of these objects and their relation to each other describe the state of the connections, and supporting basic call processes, represented by the CSA. The SCF can invoke SSF functions that manipulate these objects (e.g. changing their attributes or their relationship to each other, thereby changing the state of the connections and supporting basic call processes). This state information is provided to the SCF via information flows and information elements (e.g. EDP-Request information flows and related information elements). The allowable state changes for IN CS-2 are reflected in the semantic description of SCF-SSF information flows related to basic call processing (see BCP SIB stage 2 description and related information flows in clauses 11 and 12, respectively) and leg manipulation (see 4.3.3.7).

Objects in an IN-SSM are controlled within the context of an SCF-SSF interaction as defined by the IN-SSM type. As such, they are considered local to the IN-SSM. However, manipulating an object can have significance outside the IN-SSM and the SSF. In particular, leg manipulation has significance beyond the boundaries of the IN-SSM, even though it can only be controlled within the context of the IN-SSM. This is because a leg represents a path toward some addressable entity that may be supported by switching and transmission resources beyond the immediate control of the SSF/CCF. Thus, the relation between leg manipulation and signalling associated with switching and transmission resources should be identified as part of the semantic description of SCF-SSF information flows related to leg manipulation. This aspect is for further study (see subclause 4.3.3.7).

Other object classes related to the Call Segment Association IN-SSM are abstractions of specialized resources such as tones and announcements. These objects will not be explicitly shown in a Call Segment Association IN-SSM for IN CS-2 (though they may appear in the context of other IN-SSM types, such as "resource management" IN-SSMs). However, they may be implicitly used within a Call Segment Association IN-SSM via SSF functions that manipulate CSA objects (e.g. functions to send/receive information to/from users via legs). In addition, their use may be reflected in a CSA as an appearance of a leg representing a path to some external entity that provides specialized resources (such as may be supported by an SRF). Other object classes are not explicitly modelled for IN CS-2, though they are implied by other information flows/information elements defined for IN CS-2.

The characteristics of SSF/CCF call processing represented by CSA objects for IN CS-2 are described below. These characteristics imply the attributes and functions related to CSA objects, to be reflected in the call processing information flows/information elements defined for IN CS-2.

a) The IN CS-2 Call Segment Association provides the SCF with an abstract view of an isolated portion of a call managed by a functionally separate portion of the SSF/CCF. This isolated portion of a call is referred to as a "half-call" or *call segment* (see Figure 4-13).

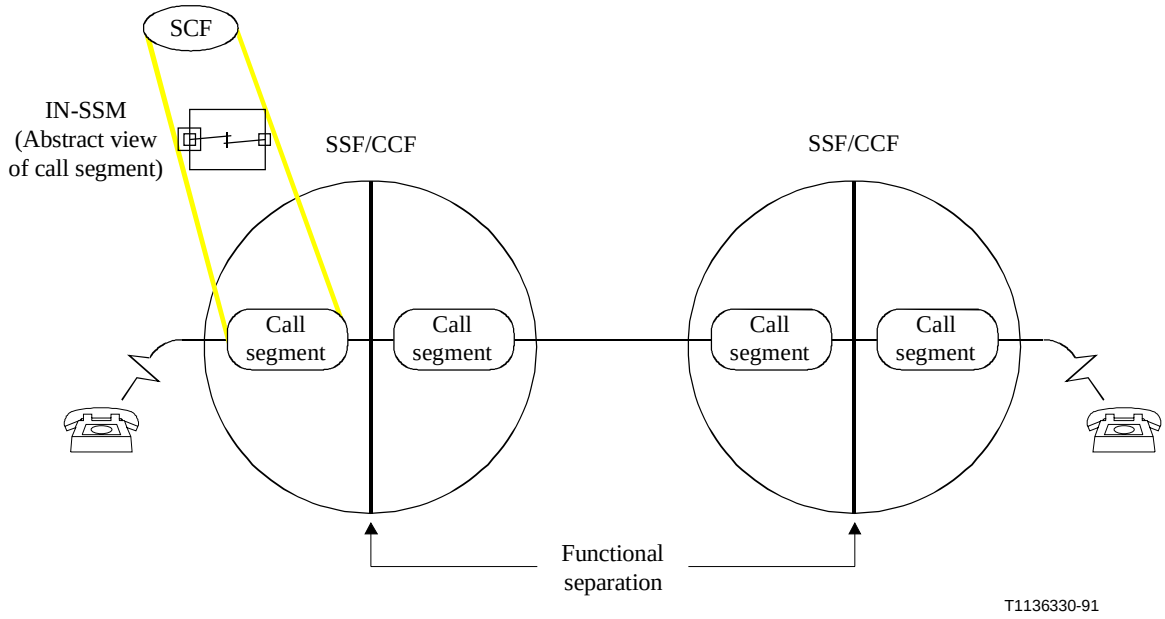


Figure 4-13/Q.1224 – Call segments in two-party inter-SSF/CCF call

This term is used to refer to the physical resources (e.g. connectivity and transmission resources represented by legs and connection points) and to the processes (e.g. basic call processes as modelled by BCSMs) that are involved in the isolated portion of a call (per Figure 4-1).

The SCF does not have direct access to a call segment, but rather has access to the abstract representation of the call segment provided by the IN CS-2 Call Segment Association. For IN CS-2, access via an IN CS-2 Call Segment Association is limited to a single two-party or multi-party[3] call segment, or to a pair of associated call segments (see Figure 4-14). A pair of associated call segments are two-call segments that can be related together by the SSF/CCF and manipulated as a pair (e.g. to merge them together into a single call segment). For IN CS-2, two call segments can only be associated if both call segments are for the same end user. For example, the SSF/CCF can associate two-call segments if the end user is involved in an existing call and would like to originate an additional call, or if the end user is involved in an existing call and there is a new call directed to that end user. This latter example is shown in Figure 4-14. The extent to which associated call segments are visible to the SCF via a CSA is for further study (see 4.3.3.7).

b) An IN CS-2 Call Segment Association provides an SCF with an abstract view of a single two-party or multi-party call segment, or of a pair of associated call segments. The CSA represents the properties of a call segment or pair of associated call segments of interest to the SCF (e.g. the connectivity and call processing aspects) and describes these properties in terms of objects (i.e. virtual resources) that can be manipulated by the SCF. For connection control, these objects include legs and connection points.

[3] Only single-ended, single point of control multi-party call segments are within the scope of CS-2.

Note that an IN CS-1 connection control IN-SSM is identical to a CSA (Call Segment Association) in terms of connectivity context for IN CS-2 as defined in 4.3.1 and 4.3.3.1, respectively.

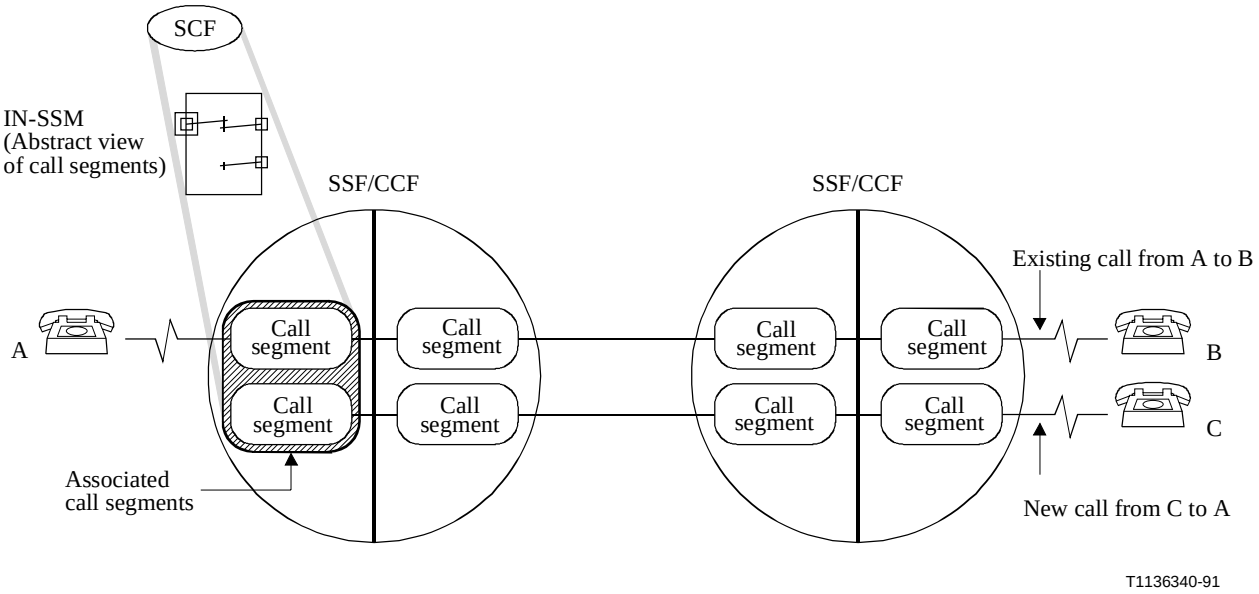


Figure 4-14/Q.1224 – Associated call segments

- A *leg* can be designated as a controlling leg or as a passive leg. For IN CS-2, the controlling leg is the leg that represents the local access interface at local exchange or the remote access interface at transit exchange (e.g. the incoming line or trunk in an originating call segment, or the outgoing line or trunk in a terminating call Segment). It is the leg for which IN service logic instances are invoked, either as a result of end user signalling (e.g. a mid-call event) or on behalf of an end user. There is no more than one controlling leg in a CSA. Transfer of ownership from an end user supported by a controlling leg to an end user supported by a passive leg is not feasible for IN CS-2.
- For IN CS-2, controlling legs represent line or trunk interfaces. The example of Connection View state transitions given for IN CS-2 presents limitations on manipulation of these two types of controlling legs. Especially, third-party control aspect which has capability of initiating call/connection set up between two parties from the third-party side is supported partially by defining controlling leg status as "surrogate". The leg status of "surrogate" represents only charging ownership for IN CS-2. Note that application of "surrogate" to passive leg is for further study.
- Legs are uniquely identifiable in a IN CS-2 Call Segment Association.
- It is possible to: influence the flow of basic call processing associated with a leg (e.g. generate a signalling event and continue basic call processing as appropriate for that event); add a passive leg to a IN CS-2 Call Segment Association by originating a call or terminating a call; to drop legs (one or more) by clearing calls; to make or break connections between legs (e.g. join or split); and to move legs from one connection point to another within the same IN CS-2 Call Segment Association (e.g. split a leg from one connection point then joining it to another). It is possible for IN CS-2 to move a leg from one call segment to another call segment within the same CSA, but is not feasible in IN CS-2 to move a leg from one CSA to another.

- A *connection point* represents a joint function between two legs, a conference function between three or more legs, replication function, merging function, or an information distribution function between two or more legs that specifies the directionality of information flow through the connection point (e.g. the connection point could receive information from multiple legs and distribute it to another leg). For IN CS-2, it interconnects legs supported by equivalent bearer services, and supports interworking between circuit mode/speech and circuit mode/3.1 kHz audio bearer services.
- There can be more than two connection points in a Call Segment Association. Each connection point is related to a call segment that is represented by the CSA. In an IN CS-2 Call Segment Association, it is possible to merge two connection points into a single connection point, thereby merging the corresponding call segments. Finally, it is possible to release a connection point and all of its legs all at once, thereby clearing the corresponding call segment.

The call segment concept can be used to describe how the definitions of "single-ended service feature" and "single point of control" apply to the distributed functional plane.

A *single-ended service feature*, as described in 7.2.1/Q.1221, is described in terms of:

- the scope of control of the service logic instance that realizes the service feature, with respect to the call; and
- the interaction of the service logic instance with respect to other single-ended service logic instances on the same call.

The scope of control of a single-ended service logic instance is restricted to the isolated "half-call(s)" in an SSF/CCF (i.e. the call segments) accessible to the SCF via a control relationship. This is illustrated in Figure 4-15 for a two-party call, which shows the BCSMs related to each call segment.

This is also extended in IN CS-2 for a pair of associated "half-calls", or a multi-party "half-call", though these scenarios are low priority for IN CS-2. These scenarios are illustrated in Figures 4-16 and 4-17.

All of these scenarios are based on the assumption that "half-calls" can be isolated from their complementary "half-calls" by the functional separation between an originating BCSM and its complementary terminating BCSM.

A single-ended service logic instance can only directly influence the processing of the isolated "half-call" (or associated "half-calls") in the SSF/CCF. The other "half-calls" can only be indirectly influenced via information propagating from one "half-call" to another (i.e. between originating and terminating BCSMs, or between BCSMs in different SSF/CCFs). As such, multiple single-ended service logic instances (one per "half-call") may be simultaneously active on a single call, each isolated from the other by the communication between "half-calls". The communication between originating and terminating BCSMs in the same SSF/CCF is described in 4.2.2 (IN CS-2 BCSM description), and is illustrated in Annex A. The communication between BCSMs in different SSF/CCFs is assumed to be the same as existing signalling between exchanges.

Single point of control, as it applies to the distributed functional plane, is as follows:

a) an isolated "half-call" in the SSF/CCF can only be influenced by one SCF at a time;

b) while one SCF is influencing an isolated "half-call" in the SSF/CCF, it shall be possible to:

- send DP report[4] information flows from the SSF/CCF to the same SCF or different SCFs;

[4] See BCP SIB stage 2 description in clause 5.

- end the control relationship between the controlling SCF and the SSF/CCF, or change the control relationship to a monitor relationship, then initiate a control relationship between the SSF/CCF and a different SCF (see 4.2.9).

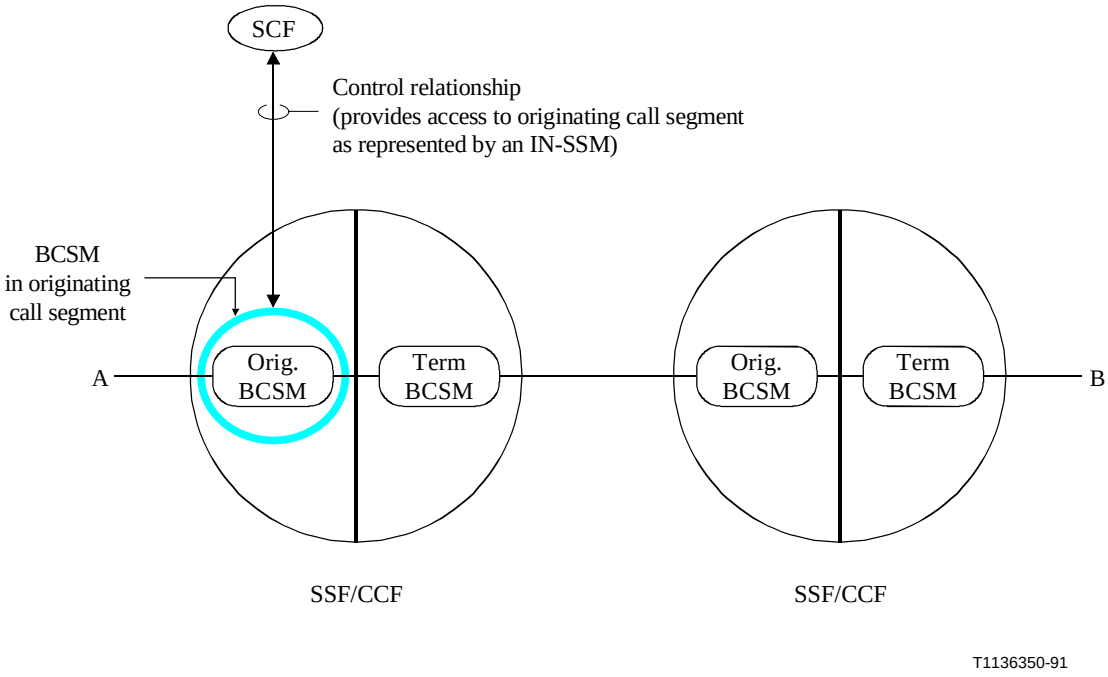


Figure 4-15/Q.1224 – Single-ended control of a two-party call

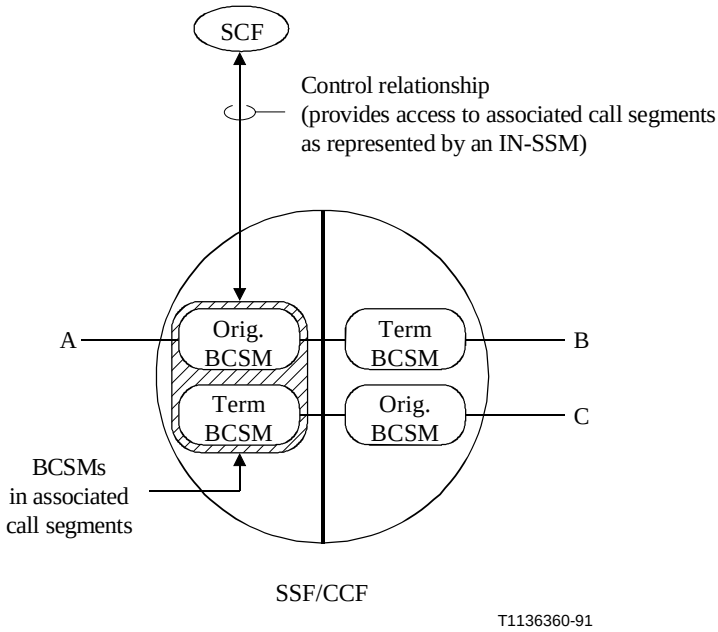


Figure 4-16/Q.1224 – Single-ended control of associated calls

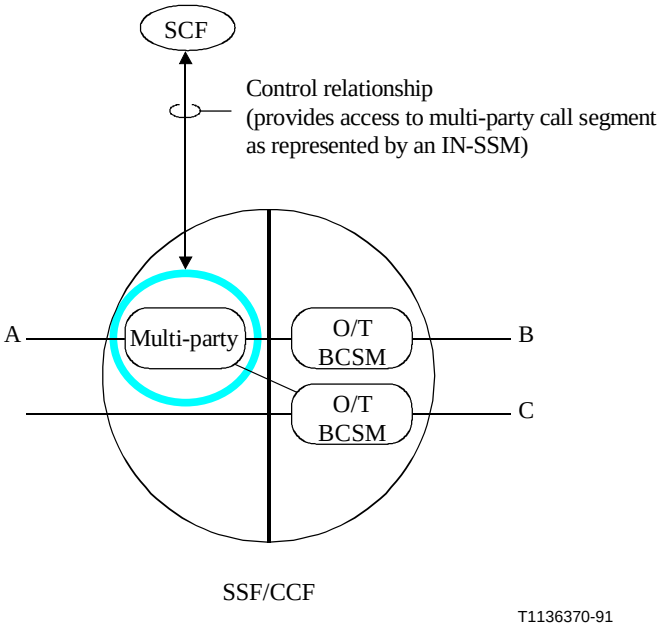


Figure 4-17/Q.1224 – Single-ended control of a multi-party call

4.3.2 IN-SM Core Capabilities for Call Party Handling

4.3.2.1 IN-SM Core Capability 1

Core Capability 1 allows for a user to enter information during a midcall event. Specifically, it is the ability for user to initiate midcall interactions and provide additional input. In order for IN to influence the call and perform call party handling, there must be a method for IN to detect the user hookflash or feature activation/indication (i.e. Midcall Detection Point), during an active call, from the controlling party. The SSF/CCF must provide this information to the SCF. Since this action can be a prelude to a large number of potential IN services/features dial tone is offered to the user for the input of service (feature) codes. This mechanism exists to convey information from the user to the SCF in order to determine the appropriate subsequent steps to be taken by IN. Both analogue as well as ISDN BRI and PRI access are supported. (Analogue trunks are for further study) Actions by the SSF/CCF include the following:

1) respond to flash or feature activation by the controlling party;
2) place passive party in an SSF/CCF-based hold condition;
3) offer dial tone to the controlling party and collect digits from the controlling party;
4) interpret the digits from the controlling party.

4.3.2.1.1 Respond to flash or feature activation by the controlling party

The initial hookflash or feature code activation by the user is captured for processing by the SSF/CCF. Since this is prior to IN involvement in the call, the feature interaction mechanisms that are in place can be used. Therefore, the SSF/CCF processes the flash or feature code according to its internal priority and precedence rules, which determine if and when IN involvement is necessary.

The specific trigger types to be recognized by the SSF/CCF include the following:

- O_Switch_Hook_Flash_Immediate (armed as an EDP or TDP);
- O_Switch_Hook_Flash_Specified_Code;
- T_Switch_Hook_Flash_Immediate;

- T_Switch_Hook_Flash_Specified_Code;
- BRI_Feature_Activation_Indicator.

4.3.2.1.2 Hold condition for passive party

SSF/CCF-based processing is required to place the passive party in a non-IN hold condition (e.g. switch-based call hold). Since this is prior to IN involvement, this condition is not to be reported to the SCF. Announcements may or may not be played to the passive party at this time (e.g. music). Depending on the switch implementation, a three-way port may be necessary at this time if music or similar announcement is to be played to the passive party. But, since availability of a three-way port is generally a function of the (switched-based) services to which a party has subscribed, use of the three-way port should not be mandatory at this time.

4.3.2.1.3 Offer dial tone to the controlling party and collect digits from the controlling party

Non-IN (e.g. switch-based) processing is also required to offer dial tone to the controlling party in order to obtain additional input from the user to process the hookflash or feature indicator. This additional information will be used to determine what actions or service features the user desires to be performed.

The method of providing the dial tone and collecting the digits can be implementation specific. For example, an external resource can be used to play announcements and collect digits or this task can be handled by internal SSF/CCF processing.

SSF/CCF processing must be sufficient to determine when the digits entered correspond to a valid digit pattern (e.g. Directory Number or feature code). Therefore, this mechanism requires SSF/CCF processing capabilities equivalent to those identified for use during the Collect_Information PIC.

4.3.2.1.4 Interpret the digits from the controlling party

The information entered by the user can be used by the SSF/CCF for further processing of SSF/CCF-based features or interpreted by the SSF/CCF that IN interactions are required. Note that this mechanism is required for subsequent IN interactions due to the potential large number of services/features that could be offered. This mechanism could also prove useful by the SSF/CCF to determine subsequent actions when only switch-based features are considered.

The SSF/CCF must be able to interpret the digits provided by the controlling party. The digits could be a Directory Number, in which case normal call processing would take place for an outgoing call from the controlling party and no IN transactions would result. Alternatively, the controlling party can enter a feature code (e.g. *003) for which the SSF/CCF would determine if IN interactions are to be involved. It should be possible to administer the valid list of feature codes on a per switch, per group of users, or per line basis.

There are two alternatives for feature code processing:

1) The Midcall Detection Point processing can include processing to determine specific feature codes. That is, the feature codes used would be specific to midcall processing and the specific codes could be reused during other stages of call processing to represent other features.

2) The SSF/CCF can have a generic ability to respond to feature codes. In this case, the particular feature code has a single meaning. Information associated with the feature code would be required to determine at what stages of call processing that the feature code is valid.

4.3.2.2 IN-SM Core Capability 2

Core Capability 2 is the ability of the SSF/CCF to connect a call party to an external resource to perform a transfer. Specifically, it is the ability of the SSF/CCF to connect to a resource/transfer each call party. The SSF/CCF is able to independently route each party participating in a given call to a point external to the SSP, under the direction of an SCF. The SSF/CCF is able to maintain the transaction with the SCF in an open state throughout the period in which the call party is routed to the external point.

Other parties involved in the call are not affected by the transfer and it is also possible to subsequently transfer these other parties to points which are external to the SSP. Actions by the SSF/CCF include the following:

1) accept an E.164 number from the SCF and establish the connection to the external point;
2) disassociate the party from the original call, thereby creating two calls;
3) attach application level correlators, as directed by the SCF, to the resultant calls;
4) perform processing on the original call as directed by the SCF.

4.3.2.2.1 Accept a DN from the SCF and establish the connection to the external point

The SCF determines the external point to which the party is to be routed and passes the information to the SSF/CCF as an E.164 number. The SSF/CCF must be able to act upon this information and establish a connection with the specified location. Normal connection processing can be used (e.g. DSS 1, ISUP) as defined by IN CS-2 for connecting to a resource.

4.3.2.2.2 Disassociate the party from the original call, thereby creating two calls

This capability calls for the SSF/CCF to create two calls from a single original call, effectively separating a party from that call. The original call could possibly be a multi-party call, and may consist of any combination of originating and terminating BCSMs. The second call that is generated will only be for a single party and will maintain the BCSM status that was associated with that party prior to the split.

There will be no SSF/CCF association between the resultant calls. Therefore, the SSF/CCF will not maintain knowledge relating one call to another. Such knowledge must be maintained external to the SSF/CCF such as the SCF.

4.3.2.2.3 Attach Application Level Correlators, as Directed by the SCF, to the Resultant Calls

The SSF/CCF must attach separate application level correlators that were generated by the SCF to each of the resultant calls. The service logic in the SCF will use these correlators in performing call party handling. The correlators will be included within any subsequent operations between the SSF/CCF and the SCF.

4.3.2.2.4 Perform processing on the original call as directed by the SCF

After creating the second call, processing must resume for the original call. The SSF/CCF actions are to be directed by the operations from the SCF. For example, the following options exist as to the nature of this processing:

1) offer dial tone to the controlling party and allow for another call origination (i.e. cause a BCSM transition from the Midcall DP at the Active PIC, to the Collect_Information PIC);
2) offer dial tone to the controlling party and allow for the collection of additional feature codes to be sent to the SCF (i.e. remain at the Midcall DP of the Active PIC);

3) proceed with the original active call (i.e. cause a BCSM transition from the Midcall DP of the Active PIC, to the Active PIC);

4) perform any other processing that is applicable.

4.3.2.3 IN-SM Core Capability 3

Core Capability 3 is the ability of the SSF/CCF to present the current call view to the SCF. Specifically, it is the ability to present the current half call view to the SCF. The SCF should be able to observe the present status of the call at a given SSF/CCF-based on the originating or terminating half call model. This view is sufficient for service logic within the SCF to be able to determine the status of current connections to the call party on whose behalf IN transactions have been invoked. This view is also able to provide information for each call party associated with this user in terms of the Basic Call State Model (BCSM). Also, current signalling conditions, events and trigger processing information are presented to the SCF. Actions by the SSF/CCF include the following:

1) determine if a control or monitoring relationship exists;

2) associate the appropriate correlator (e.g. correlation ID or Session ID, etc.) as previously instructed by the SCF;

3) generate a view of half call;

4) halt call processing for the half call and send query to SCF.

4.3.2.3.1 Determine if control or monitoring relationship exists

The SSF/CCF must determine if a previous (IN) control or (IN) monitoring relationship exists and apply the precedence and priority rules to determine if an IN transaction can be initiated and any associated limitations.

4.3.2.3.2 Associate a correlator

If the SCF has provided an application level correlator for use with this transaction, then that correlator must be associated with this transaction. Note that the correlator could have been assigned during previous IN interactions within this transaction or it could have been assigned during an earlier transaction. Service logic within the SCF will use this correlator as appropriate.

4.3.2.3.3 Generate a connection view of the half call

Information relating to the half call segment (originating or terminating) within this SSF/CCF must be packaged in a form that will be recognizable to service logic within the SCF (e.g. Connection View states or state of any switch-based features that may apply). The particular form of the package will depend on the call party handle technique that is being used. The information to be packaged should include the following:

1) status of any previous relationships (e.g. control, monitoring) that may exist between the SSF/CCF and any SCF, for this half call (DP processing rule 5 is for further study);

2) status of any call parties associated with this half call (e.g. joined, shared, pending);

3) status of any existing half calls associated with this half call;

4) status of the BCSM related to this half call;

5) status of the BCSM related to any associated half calls;

6) information entered by the user related to the midcall event (e.g. feature code);

7) cause of the midcall event (e.g. hookflash, feature activator).

4.3.2.3.4 Halt call processing for the half call and send query to SCF

Processing as defined for IN CS-2.

4.3.2.4 IN-SM Core Capability 4

Core Capability 4 is the ability of the SSF/CCF to combine separate calls into a single call. Specifically, it is the ability for the SSF/CCF to combine selected transferred paths within the SSP into a call. The SSF/CCF must be able to regain control of calls that had previously been externally routed. Upon notification from the SCF, the SSF/CCF is to combine selected calls, as identified by correlator(s), into a single call with the controlling party as identified by the SCF. The form of the resultant call is limited by the connection capabilities of the SSF/CCF (e.g. two-way, three-way, etc.). Actions by the SSF/CCF include the following:

1) disconnect connections to the external point;

2) associate the parties involved based on application level correlators;

3) bring resultant call to the active state.

4.3.2.4.1 Disconnect connections to the external point

The form of this disconnect is equivalent to that identified by IN CS-2 for disconnect from an SRF, therefore normal disconnect (e.g. ISUP, DSS 1) can be used.

4.3.2.4.2 Associate the parties involved based on application level correlators

Upon receiving an operation from the SCF, the SSF/CCF selects the calls identified by the correlators within the operation. The SSF/CCF is then to merge these calls into a single call. The controlling party is determined by which transaction is used to convey the operation to the SSF/CCF from the SCF. The result is a transient state in which the calls are associated.

4.3.2.4.3 Bring resultant call to the active state

The SSF/CCF continues processing of the associated calls from the above description and brings the call to an active state. The BCSMs involved maintain the status (originating, terminating) as before the merge. A clear talk path is generated between all parties involved. Only a single transaction remains after the merge.

4.3.3 The Connection View State (CVS) approach

This subclause describes the Connection View State approach in the context of the Connection View model, defines an inventory of Connection View states, and lists the allowable transitions between CVSs.

4.3.3.1 The Connection View (CV) model

The Connection View State approach is based on the Connection View (CV) model. CV processing provides an SCF with the ability to influence existing call and connection processing capabilities. It does so by providing a generic representation of call and connection processing resources that support the processing capabilities of interest.

As Figure 4-18 illustrates, the CV processing within the SSF/CCF can be viewed as translating SCF instructions into operations that are understood by internal SSF/CCF call processing, as well as translating internal call processing events and the state of internal call processing resources into information which is understood by the SCF.

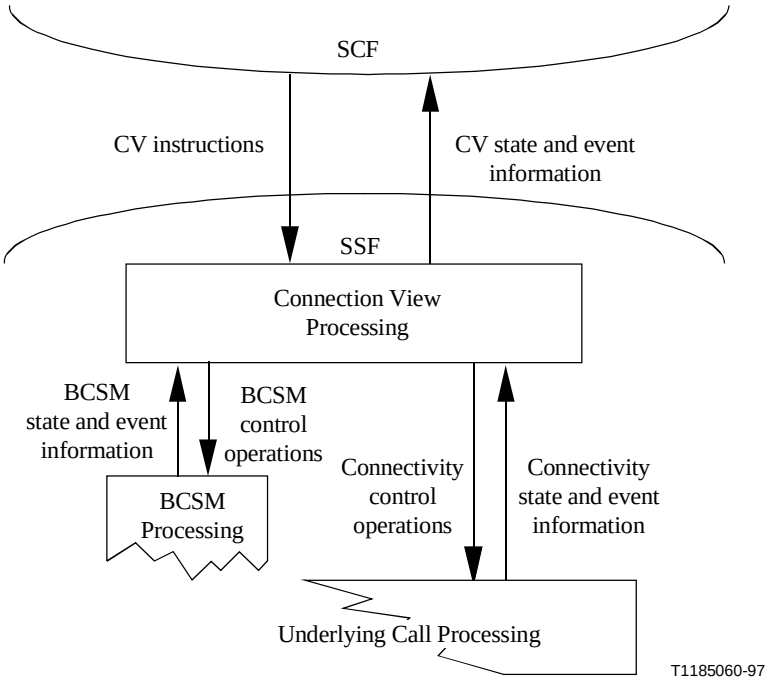


Figure 4-18/Q.1224 – Connection View (CV) processing model

The SSF/CCF call processing and CV processing resources are described by a set of CV objects which include the following:

- Call Segment Association (CSA),
- Call Segment (CS),
- Legs,
- Connection Point (CP),
- BCSM.

See 4.3.3.5, Figure 4-20, that illustrates the relationship of these objects to one another.

The CV provides a view or abstraction of call and connection processing resources that is independent of supplier implementation and that only represents the essential characteristics of the resources needed by service logic, hiding the physical details and technical complexity of these resources. The CV reflects these properties in terms of *connectivity* context and a *call processing* context. The objects in the *call processing* context manage the relationship of a party to a basic two-party call and the objects in the *connectivity* context manage multiple two-party calls.

The *call processing* context reflects the state of the basic call processing required to set up and maintain the legs in a CS. Only one type of object is explicitly defined in the call processing context and that is BCSM. The BCSM represents the basic call processing required to establish and maintain a communication path from the CS toward an originating party and from the CS toward a terminating party.

The *connectivity* context reflects the state of a CS or a pair of associated CSs and includes the set of legs in the CSs and the relationship of each leg to a Connection Point (CP). The types of objects in the connectivity context correspond to legs, CPs, Call Segments (CSs), and pairs of associated CSs (CSA). Figure 4-19 provides the graphical examples of the object relationship in the connectivity context.

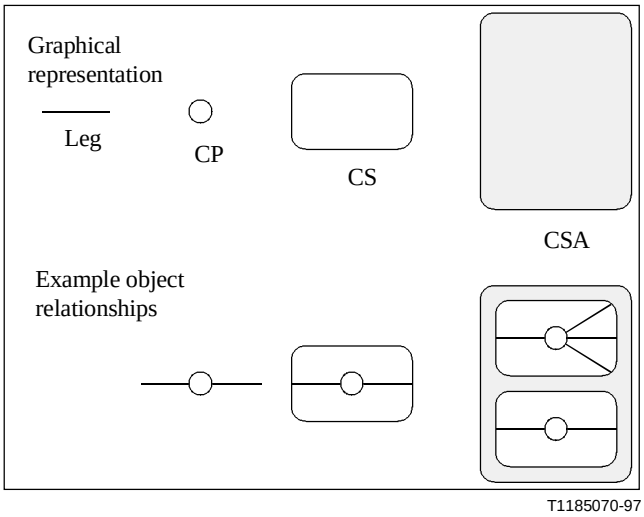


Figure 4-19/Q.1224 – Connection View objects

4.3.3.2 Definition of Connection View State

The connection configuration of a CV may be described by a Connection View State (CVS). Each CVS represents the connectivity objects of a CV. The CVSs refine the connection view to identify specific states and transitions that are accessible to the SCF.

Subclause 4.3.3.5 describes the notation used to illustrate CVSs. Note that a CVS represents a **"half-call"** view (i.e. a single two-party CS, a pair of associated CSs, or a single multi-party CS).

4.3.3.3 CVS Model Philosophy

The CVS model unambiguously defines the CVSs so that minimal information is to be sent in the messages to the SCF. The CVSs are further used to define the allowed state transitions that shall be supported by CV processing in the SSF/CCF. The CV processing manages (i.e. creates, updates, and destroys) the CVSs to provide current CV information to the SCF. The CV processing creates a CVS when the SSF/CCF detects an event in a given CS, or an associated CS if one already exists, or no CVS exists for the CS. The CV processing updates the CVS to represent the current view due to the detection of a call processing event, or as a result of processing a new SCF message. SCF messages may be sent in response to call processing events received from the SSF. SCF messages may also be sent in cases the SSF/CCF does not explicitly expect these messages, e.g. as a result of charging notifications (e.g. operations ApplyChargingReport, EventNotificationCharging), or of a timer supervision in the SCF (e.g. Call Connection Elapsed Time Supervision), or of a user to service information (USI), or of DP events sent from the SSF in notification mode. The CV processing destroys the CVS due to certain call processing events that can result from the user actions, other call processing events, or the SCF messages.

4.3.3.4 Relationship of the Connection View States to the SDL diagrams

The text in Q.1224 provides a high level overview of CPH and the Connections View State (CVS) diagrams. Connection View States provided in the text are examples of how Call Segments can be combined to form a CVS. Further, the text provides examples of some common transitions among the CVSs.

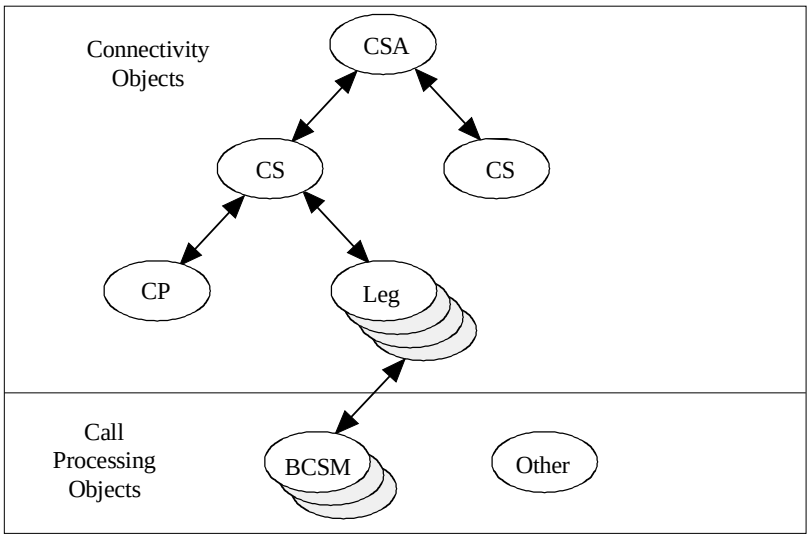
Connection View States are informative, only the Call Segments within a CVS have defined states. Definitions of the Call Segments states can be found in Annex A/Q.1228. Annex A/Q.1228 provides a diagram to show a complete set of transitions among the Call Segment states, as well as a set of

SDLs to show individual details. These should be used when designing services in order to ensure a correct implementation.

4.3.3.5 Relationship of BCSM to Connection View States

A Connection View State is a model used to categorize the status of one or more Call Segments, each comprising a set of one or more Legs. The purpose of defining Connection View State is to catalogue a set of examples used to describe leg manipulation services.

Figure 4-20 shows, at a high level, the relationship of the Connectivity Objects that make up the Connection View States to the Call Processing Objects (BCSMs).



CSA CS Association
CS Call Segment
CP Connection Point

Figure 4-20/Q.1224 – Connection View State objects and their relationship with BCSM

A Connection View State shows the connectivity between a Controlling Leg and one or more Passive Legs (see below for definitions of terms).

A single BCSM models the Originating or Terminating call processing to set up and maintain a two-party call. **There is a single instance of a BCSM for each Passive Leg of a Connection View State (CVS).**

In case no passive leg is present within a CS in the CVS, the BCSM instance belongs to the controlling leg in that CS.

A Call Segment with only a controlling leg (status "joined") left is assigned a BCSM instance (Call processing Context) to supervise the leg. The "BCSM type" attribute set on the controlling leg is used to reflect the type of BCSM (originating or terminating) to be assigned. The corresponding FSM for the Call Segment is put into the "Waiting_For_Instruction" state when a leg is left in a CS on request from the SCF due to a CPH operation (e.g. DisconnectLeg). The call processing is suspended at the O/T_MidCall DP in the associated BCSM.

When the controlling leg becomes connected again to another passive leg (e.g. MergeCallSegment, Connect or AnalyseInformation operation) and the BCSM instances are the same type no BCSM instance will be connected to the controlling leg anymore. The DP armed for the controlling BCSM

instance will be transferred to the passive leg BCSM instance. In case of conflicting BCSM types this will lead to an error condition.

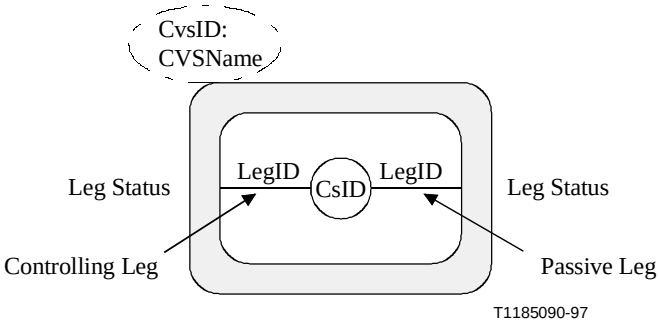


Figure 4-21/Q.1224 – Representation used in a CVS

Figure 4-21 illustrates the notation used in describing the CVSs.

1) Each CVS identifies the Call Segments (CSs) in that CVS and lists the call segment identifier (CsID) in the connection point.

2) A leg represents a communication path towards an end user.

   a) In a CS, only one leg is directed towards the originating or a terminating access and is called the controlling leg. The access signalling on the controlling leg can influence the progress of the call or the indication of access signalling (e.g. RELease COMplete message from the called ISDN terminal) received by the passive legs can also influence the progress of a call. As part of the convention for illustrating CVSs, the controlling leg is always drawn on the left side of the connection point and is identified with a LegID of "c".

   b) As part of the convention for illustrating CVSs, the legs connected to the right side of the connection point are always called the passive legs and are identified with a legID of "p1" or "p2".

   c) There is one controlling leg in a CSA for IN CS-2.

   d) Leg status of "pending" or "joined" for controlling leg is assigned leg "c". Leg status of "null", shared", or "surrogate" for controlling leg does not have leg notations.

   e) When a controlling leg status in a CSA is returned to null or shared after CVS transition, the controlling leg in the CS is transferred to the associated CS where controlling leg is located as leg status of "joined" or "pending".

   f) Leg notation of passive legs is unique in a CSA and is identified with legIDs of "p1’, "p2", "p3" etc.

      Leg notation of "p1", "p2" etc. represents sequential values in incremental order such as "p1", "p2", "p3", starting from the passive legs in CS#1 to those in maximum CS#n in the CSA.

      Hence, number of passive legs are shown by maximum number of suffix for leg notation of passive legs.

   g) Leg status of "pending", "joined" for passive legs are assigned leg "p1", "p2" etc.

      Leg status of "null" for passive leg does not have leg notation. Note that a passive leg status of "shared" is not supported for IN CS-2 so that notation rule of passive legs including leg status of "shared", "joined" or "pending" is for further study .

h) When a passive leg status in CSA is returned to null after CVS transition, remaining passive legs in CSA are re-notated sequentially in incremental order like "p1", "p2", etc., starting from the passive legs in CS#1 to those in maximum CS#n in the CSA. Hence, number of passive legs are shown by maximum number of suffix for leg notation of passive legs.

3) A leg in a call segment is represented by either a solid or dashed line. The leg status reflects the status of the leg in a CS and is shown at the end of the leg.

a) A solid line for the leg indicates that the leg status is "joined". The leg status of "joined" in a CS indicates that a path is joined to the connection point which enables the user to communicate with other users in the CS.

b) A dashed line indicates that the leg status is either "shared", "pending", or "surrogate".

– The leg status of "shared" indicates that a controlling leg is absent from a CS and is present in the associated CS.

– The leg status of "pending" in a CS indicates that a path is in the process of being set up.

A controlling leg status is "pending" during the call setup PICs of a T_BCSM. A passive leg status is pending during the call setup PICs of an O_BCSM only.

– A controlling leg with the status "surrogate" indicates that a leg supports a communication path towards a virtual party in the network, rather than towards an external end party. A virtual party behaves either on behalf of a party not participating in the call (e.g. call forwarding) or on behalf of the network itself (e.g. wake-up calls). A "surrogate" leg allows a virtual party to invoke features in the course of providing its service although the party is not active (e.g. billing). The definition of "surrogate" legStatus for the "Originating Setup M-Party" CVS and "Active M-Party" CVS is for futher study.

For a two-party CS, there is a controlling leg and a passive leg connected to a connection point. There is either an originating or a terminating BCSM associated with the passive leg of the two-party CS. For a multi-party CS, there is one controlling leg and two or more passive legs. Each passive leg is associated with an O_BCSM or a T_BCSM.

Examples

CVS: Originating Setup

See Figure 4-22.

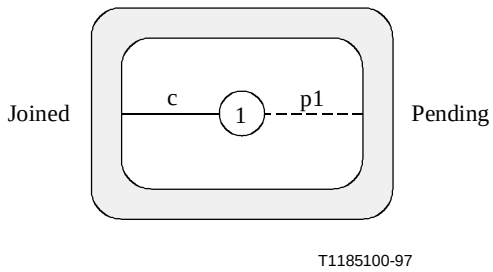


Figure 4-22/Q.1224 – CVS: Originating Setup

In CVS "Originating Setup" there is an Originating BCSM associated with the Pending passive leg. That BCSM is in the call setup phase, since the passive leg is still Pending. When the BCSM reaches the stable phase, this CVS will transition to CVS: "Stable 2-Party".

CVS: Terminating Setup

See Figure 4-23.

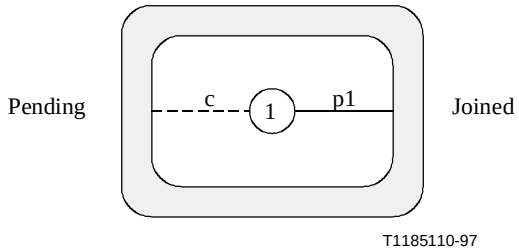


Figure 4-23/Q.1224 – CVS: Terminating Setup

In CVS: "Terminating Setup" there is a Terminating BCSM associated with the Joined passive leg. That BCSM is in the call setup phase, since the Controlling Leg towards the called party is still Pending. When the BCSM reaches the stable phase, this CVS will transition to CVS: "Stable 2-Party".

CVS: M-Party Setup

See Figure 4-24.

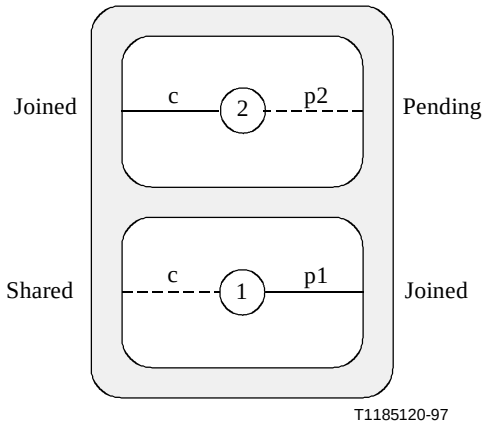


Figure 4-24/Q.1224 – CVS: M-Party Setup

In CVS "M-Party Setup," there is one Originating BCSM associated with the Pending passive leg p2 in the top CS (CS 2), and one stable (Originating or Terminating) BCSM associated with the Joined passive leg p1 in the bottom CS (CS 1). The BCSM for CS 2 is in the call setup phase, since the passive leg is still Pending. When the BCSM for CS 2 reaches the stable phase, CVS "M-Party Setup," will transition to CVS "Call on Hold." In CVS "M-Party Setup" the controlling leg, Leg c is shared between the two calls. The controlling leg participates in both BCSMs that lie under the CVS view. This is why we say there is one BCSM per passive leg, to avoid confusion with the role of the controlling leg. Call processing events that occur during this CVS are reported in the context of the BCSM where the event occurs.

CVS: Stable M-Party

See Figure 4-25.

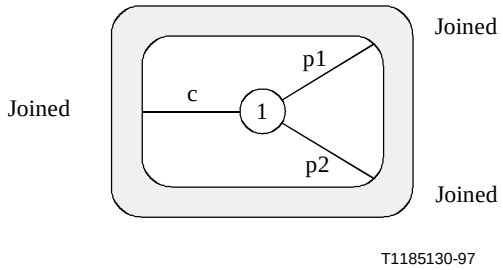


Figure 4-25/Q.1224 – CVS: Stable M-Party

In CVS "Stable M-Party," there is one Originating or Terminating BCSM associated with each Joined passive leg. These BCSMs are in the stable or clearing phase. The controlling leg, Leg c, participates in 2 BCSMs simultaneously in this CVS. It is possible that one of these is an Originating BCSM and one is a Terminating BCSM. When an event happens due to an action on Leg c (e.g. a switch-hook flash), it is not obvious which BCSM context should be used to report the event. See the following subclause for details on how to report events detected on more than one BCSM.

4.3.3.6 Rules for inter-BCSM precedence

For Call Party Handling, it is possible to detect the same event (e.g. mid-call or user abandon) on multiple BCSMs. However, it is necessary to provide only one report of the given event to SCF.

Event detection rules

The inter-BCSM precedence rules for the *detection* of events signalled on the controlling leg follow:

1) If the event signalled on the controlling leg does *not* indicate the particular passive leg to which it applies (e.g. analog hook-flash), then the following rules apply:
   a) if the CSA contains multiple Call Segments (CSs), the mid-call event should be detected only on the CS with a controlling *legStatus* of "joined";
   b) if the CS to which the event applies is a multi-party CS, then the mid-call event should be detected only on all BCSMs in the multi-party CS.
2) If the event signalled on the controlling leg indicates the particular passive leg to which it applies (e.g. party identifier indication), then the mid-call event should be detected on the BCSM associated with the indicated passive leg.

Event reporting rules

The inter-BCSM precedence rules for the *reporting* (to SCF) of events signalled on the controlling leg follow:

1) An event is reported only once to SCF.
2) A "BCSM type" attribute on the controlling leg indicates if the controlling belongs to an originating or a Terminating BCSM. The "BCSM type" attribute is set to reflect the type of BCSM (originating or terminating) in which the trigger or event occurred at the moment the CS was created.
3) If the controlling leg belongs to an originating BCSM, it is reported once as an originating event.

4) If the controlling leg belongs to a terminating BCSM, it is reported once as a terminating event.

Example

To illustrate these rules with an example, events are reported for the "Stable M-Party" CVS as follows:

a) If all of the underlying BCSMs are O_BCSMs, the call processing events detected on Leg c (e.g. at the O_Mid_Call or O_Disconnect DPs) are reported from the appropriate O_BCSM DPs.

b) If all of the underlying BCSMs are T_BCSMs, the call processing events detected on Leg c (at the T_Mid_Call or T_Disconnect DPs) are reported from the appropriate T_BCSM DPs.

c) If the underlying BCSMs are of different types, the call processing events detected on Leg c (e.g. at the Mid_Call or Disconnect DPs) are reported from the appropriate O_BCSM DPs.

d) If an event is detected in more than one BCSM, it is reported only once. This can happen for example when there are two T_BCSMs or two O_BCSMs and the event occurs because of an action on the controlling leg (Leg c). If the controlling leg makes a switch-hook flash, it is detected in both BCSMs, but the event should be reported only once according to rules a) to c).

Rule c) above indicates that the events on Leg c are reported from the O_BCSM since the list of possible events for Leg c from the O_BCSM is a superset of the possible events for Leg c from the T_BCSM.

4.3.3.7 Connection View States (CVSs) and possible transitions

This subclause describes the Connection View (CV) processing for each CVS, and gives examples of the events or the SCF messages that result in creating a new CVS, destroying a CVS, and transitioning from the one CVS into another CVS. The transitions between CVSs occur due to end user actions (e.g. off-hook or disconnect), switch processing (e.g. switch-based features), or as a result of processing SCF messages. Note that the transition events related to user interactions (e.g. ConnectToResource) are left for further study. *Figure 4-27 and Table 4-8 provide a concise summary of these transition events*.

Note that *this subclause concentrates on those IN-related events that support the Connection View State approach to Call Party Handling*. IN-initiated transitions may be categorized according to the manner in which they affect the CVS:

1) *BCSM-only transitions*: Occur when the BCSM changes state (or continues normal processing) with no explicit change to the CVS. For example, if SSF sends an OMidCall message from the "M-Party Setup" CVS, and SCF responds with AnalyseInformation, the BCSM changes state, but no explicit change to the initial CVS occurs. Note that, in processing the BCSM request, there *may* be an implicit change to the CVS through non-IN processing (e.g. a switch-based screening feature). *BCSM-only transitions will* ***not*** *be addressed in this subclause*.

2) *SCF-requested connectivity changes:* Occur when SCF explicitly requests a change in connectivity. This request may or may not cause a CVS transition.

Only category 2) will be addressed in this subclause.

Within this subclause, each CVS is described with respect to the following:

a) *Relationship with the BCSM*: a list of the Points in Call (PICs) that are associated with this CVS. Note that the PICs are associated with the passive legs (i.e. *Leg* p1 or *Leg* p2) only, to avoid the ambiguity that would otherwise result for the "Stable M-Party," "Transfer", and

"Forward" CVSs. Additional detail on the mapping of DPs and events to the CVSs is for further study.

For ease of reference, the PICs have been categorized as follows:

- Originating *Call Setup*:
  Authorize_Origination_Attempt, Collect_Information, Analyse_Information, Select_Route and Authorize_Call_Setup.
- Originating *Stable Call*: Send_Call, O_Alerting and O_Active.
- Originating *Call Clearing*: O_Suspended.
- Terminating *Call Setup*:
  Authorize_Termination_Attempt, Select_Facility, Present_Call and T_Alerting.
- Terminating *Stable Call*: T_Active.
- Terminating *Call Clearing*: T_Suspended.

b) *Entry Events*: the events that cause the call processing and CV processing associated with this CVS to be initiated.

c) *Exit Events*: the events that signify normal completion of the call processing and CV processing associated with this CVS. This item begins with a summary of the SSF events (i.e. TDPs, EDPs, and BCSM events) relevant to the CVS approach to Call Party Handling. A list of SCF responses that cause CVS transitions follow.

Figure 4-26 depicts the CVSs to be supported by the CV processing. The notation used in this figure is described in 4.2.5 (Relationship of BCSM to Connection View States).

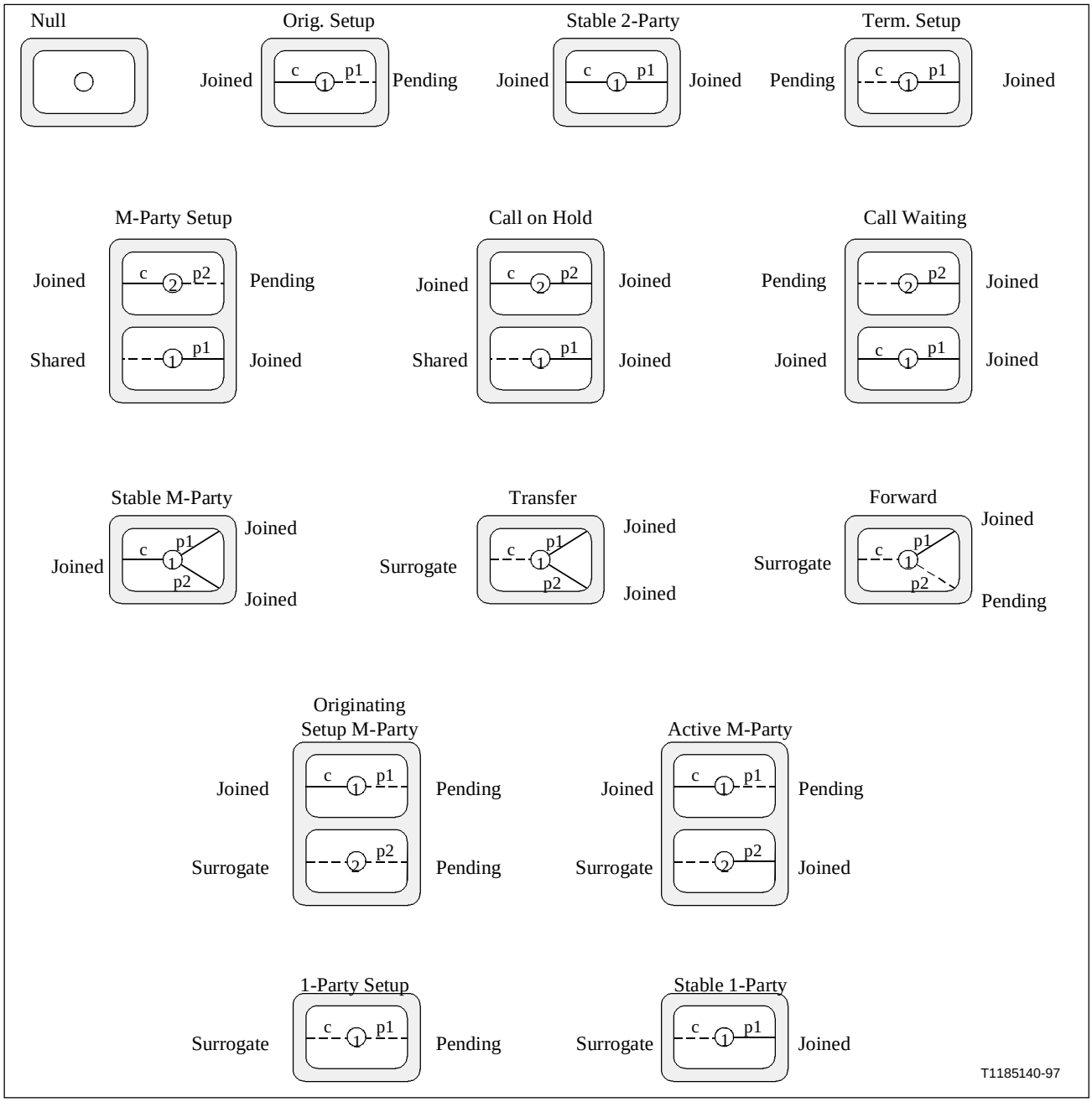


Figure 4-26/Q.1224 – Connection View States

The following subclauses provide detailed descriptions of these Connection View States (CVSs), including their relationship with the BCSM and the transition events between CVSs:

4.3.3.7.1	"Null"
4.3.3.7.2	"Orig. Setup"
4.3.3.7.3	"Stable 2-Party"
4.3.3.7.4	"Term. Setup"
4.3.3.7.5	"M-Party Setup"
4.3.3.7.6	"Call on Hold"
4.3.3.7.7	"Call Waiting"
4.3.3.7.8	"Stable M-Party"
4.3.3.7.9	"Transfer"
4.3.3.7.10	"Forward"
4.3.3.7.11	"Originating Setup M-Party"
4.3.3.7.12	"Active M-Party"
4.3.3.7.13	"1-Party Setup"
4.3.3.7.14	"Stable 1-Party"

4.3.3.7.1 "Null" Connection View State

The "Null" CVS represents a condition where call processing is not active. There is no controlling leg or passive leg connected to the connection point.

a) *Relationship with the BCSM*: The "Null" CVS is associated with the call processing in O_Null or T_Null.

b) *Entry Events*: Disconnecting of a previous call, after exception handling, or system (re)initialization. Details are for further study.

c) *Exit Events*:

Summary of CVS-related SSF events: Origination_Attempt and Termination_Attempt.

Transitions from the "Null" CVS:

- To the "Originating Setup" CVS
  - Detection of the Origination_Attempt event as described in O_Null.
- To the "Terminating Setup" CVS
  - Detection of the Termination_Attempt event as described in T_Null.
- To the "1-Party Setup" CVS
  - Processing of a valid InitiateCallAttempt message received from SCF.

4.3.3.7.2 "Originating Setup" Connection View State

The "Originating Setup" CVS represents an originating two-party call in the setup phase. The characteristics of the "Originating Setup" CVS follow:

a) *Relationship with the* BCSM:

controlling *leg* = c:
legStatus = joined

passive *leg* = p1:
legStatus = pending
bcsmPic = originating call setup

b) *Entry Events*: See the exit events for the "Null", "Stable 2-Party," and "M-Party Setup" CVSs.

c) *Exit Events*:

Summary of CVS-related SSF events: Call Setup Authorized, Collected_Information or Analysed_Information.

Transitions from the "Originating Setup" CVS:

- To the "Stable 2-Party" CVS
  - Detection of the Call Setup Authorized event as described in Authorize_Call_Setup.
- To the "Originating Setup M-Party" CVS
  - Processing of a valid InitiateCallAttempt message received from SCF, in response to a CollectedInformation or AnalysedInformation message for *leg* c.

4.3.3.7.3 "Stable 2-Party" Connection View State

The "Stable 2-Party" CVS represents a stable or clearing two-party call, and is either an originating or a terminating call from the perspective of the controlling user. The characteristics of the "Stable 2-Party" CVS follow:

a) *Relationship with the* BCSM:

controlling *leg* = c:
legStatus = joined
passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating stable call; terminating call clearing

b) *Entry Events*: See the exit events for the "Originating Setup," "Terminating Setup," "M-Party Setup," "Call On Hold," "Call Waiting," "Stable M-Party," "Transfer", and "Active M-Party" CVSs.

c) *Exit Events*:

Summary of CVS-related SSF events:

Switch_Hook_Flash_Immediate trigger, BRI_Feature_Activation_Indicator trigger, second dial tone associated with the Switch_Hook_Flash_Specified_Code trigger, O_Mid_Call, T_Mid_Call, O_Abandon, O_Suspended, O_Disconnect, and T_Disconnect.

Transitions from the "Stable 2-Party" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF, in response to an OMidCall or TMidCall message (reporting the detection of a Switch_Hook_Flash_Immediate or BRI_Feature_Activation_Indicator trigger, or a mid-call EDP).
  - Processing of a valid ReleaseCall message received from SCF, in response to an OAbandon, ODisconnect or TDisconnect message for *leg* c.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1.
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or an user to service information).
- To the "Originating Setup" CVS:
  - Processing of a valid message containing DisconnectLeg for *leg* p1, plus AnalyseInformation, SelectRoute, or Connect received from SCF, in response to an

OMidCall message (reporting the detection of a Switch_Hook_Flash_Immediate or BRI_Feature_Activation_Indicator trigger, or a mid-call EDP).

– In response to an OMidCall or TMidCall message, processing of a valid SplitLeg message, followed by a CreateCallSegmentAssocation message (which creates a null CSA), followed by a MoveCallSegments message (to move CS 2 from its current location in the source CSA into a new target CSA) and a CollectInformation message on the CS which remains in the source CSA (to transition the unmoved CS into the "Originating Setup" CVS).

NOTE 1 – For TMidCall DP, this transition is valid at a local exchange and not at a transit exchange.

– Processing of a valid message containing DisconnectLeg for *leg* p1, plus AnalyseInformation or SelectRoute or Connect received from SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).

– Processing of a valid message containing DisconnectLeg for *leg* p1, plus AnalyseInformation or SelectRoute or Connect received from SCF, in response to an OSuspended message for *leg* p1.

– Processing of a valid AnalyseInformation, SelectRoute or Connect message received from SCF, in response to an ODisconnect message for *leg* p1.

– Processing of a valid Connect message received from SCF, in response to a TDisconnect message for *leg* p1.

NOTE 2 – AnalyseInformation and SelectRoute are allowed only in response to originating events.

- To the "M-Party Setup" CVS:
  - Processing of a valid SplitLeg message received from SCF, followed by user interaction, and then a valid AnalyseInformation, SelectRoute or Connect message received from SCF, in response to an OMidCall message (reporting the detection of a Switch_Hook_Flash_Immediate or BRI_Feature_Activation_Indicator trigger, or a mid-call EDP).
  - Processing of a valid SplitLeg message received from SCF, followed by user interaction, and then a valid Connect message received from SCF, in response to a TMidCall message (reporting the detection of a Switch_Hook_Flash_Immediate or BRI_Feature_Activation_Indicator trigger, or a mid-call EDP).
  - Processing of a valid SplitLeg message received from SCF, which was initiated by the SCF (caused by, e.g. a user to service information), followed by user interaction, and then a valid AnalyseInformation, SelectRoute, or Connect message received from SCF.
  - The detection of a "flash", followed by the application of second dial tone for a Switch_Hook_Flash_Specified_Code trigger.
- To the "Forward" CVS:
  - In response to an OMidCall or TMidCall message, processing of a valid SplitLeg message, followed by a CreateCallSegmentAssociation message (which creates a null CSA), followed by a MoveCallSegments message (to move CS 2 from its current location in the source CSA into a new target CSA) and a Connect message on the moved CS in the target CSA (to forward the call).

- To the "Stable 2-party" CVS:
  - In response to an OMidCall or TMidCall message, processing of a valid SplitLeg message, followed by a CreateCallSegmentAssocation message (which creates a null CSA), followed by a MoveCallSegments message (to move CS 2 from its current location in the source CSA into a new target CSA) and a Connect message on the CS which remains in the source CSA (to transition the unmoved CS back into the Stable 2-party CVS).

    NOTE 3 – For TMidCall DP, this transition is valid at a local exchange and not at a transit exchange.

4.3.3.7.4 "Terminating Setup" Connection View State

The "Terminating Setup" CVS represents a terminating two-party call in the set-up phase. The characteristics of the "Terminating Setup" CVS follow:

a) *Relationship with the* BCSM:

controlling *leg* = c:
legStatus = pending

passive *leg* = p1:
legStatus = joined
bcsmPic = terminating call setup

b) *Entry Events*: See the exit events for the "Null" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events: T_Busy and T_Answer.

Transitions from the "Terminating Setup" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF, in response to a TBusy message.
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or an user to service information).
- To the "Stable 2-Party" CVS:
  - Detection of the T_Answer event as described in T_Alerting.
- To the "Call Waiting" CVS:
  - Processing of a valid message containing MoveCallSegments, plus user interaction received from SCF, in response to a TBusy message.

    NOTE – Interaction is optional.
- To the "Forward" CVS
  - Processing of a valid Connect message received from SCF, in response to a TBusy message.

4.3.3.7.5 "M-Party Setup" Connection View State

The "M-Party Setup" CVS represents two associated call segments. The controlling user only has put one party on hold, and has originated a new call which has not yet reached the stable phase. Note that how the passive user can put one party on hold for some services is for futher study in order to receive the request from remote party side at transit exchange.

CS 2 represents an originating two-party call in the setup phase.

Call processing in *CS* 1 is in the stable or clearing phase, and represents either an originating or terminating call segment. The passive leg (*leg* p1) is on hold. Note that the direction of the call on hold is irrelevant.

a) *Relationship with the* BCSM:

CS = 2:
controlling *leg* = c:
legStatus = joined
passive *leg* = p2:
legStatus = pending
bcsmPic = originating call setup

CS = 1:
controlling leg:
legStatus = shared
passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating stable call; terminating call clearing

b) *Entry Events*: See the exit events for the "Stable 2-Party" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events:
Switch_Hook_Flash_Specified_Code trigger, O_Abandon, O_Suspended, O_Disconnect, T_Disconnect, and Call Setup Authorized.

Transitions from the "M-Party Setup" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF, in response to an OMidCall or TMidCall message (reporting the detection of a Switch_Hook_Flash_Specified_Code trigger).
  - Processing of a valid ReleaseCall message received from SCF, in response to an OAbandon message (for *leg* c).
  - Processing of a valid ReleaseCall message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1.
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Originating Setup" CVS:
  - Processing of a valid DisconnectLeg message for *leg* p1 received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1.
  - Processing of a valid DisconnectLeg message for *leg* p1 received from the SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Stable 2-Party" CVS:
  - Processing of a valid message containing DisconnectLeg for *leg* p2, plus MergeCallSegments, plus Reconnect received from SCF, in response to an OAbandon message (for *leg* c).

- To the "Call on Hold" CVS:
  - Detection of the Call Setup Authorized event for *CS* 2, as described in Authorize_Call_Setup.

4.3.3.7.6 "Call on Hold" Connection View State

The "Call on Hold" CVS represents two associated call segments. The controlling user only has put one party on hold, and is participating in another call which is in the stable or clearing phase. Note that how the passive user can put one party on hold for some services is for futher study in order to receive the request from remote party side at transit exchange.

CS 2 represents a two-party call in the stable or clearing phase.

Call processing in *CS* 1 is in the stable or clearing phase, and represents either an originating or terminating call segment. The passive leg (*leg* p1) is on hold. Note that the direction of the call on hold is irrelevant.

a) *Relationship with the* BCSM:

CS = 2:
controlling *leg* = c:
legStatus = joined

passive *leg* = p2:
legStatus = joined
bcsmPic = originating stable call; originating call clearing; terminating stable call; terminating call clearing

CS = 1:
controlling *leg*:
legStatus = shared

passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable call; originating call clearing; terminating stable call; terminating call clearing

b) *Entry Events*: See the exit events for the "M-Party Setup," "Call on Hold," "Call Waiting," and "Stable M-Party" CVSs.

c) *Exit Events*:

Summary of CVS-related SSF events:
O_Mid_Call, T_Mid_Call, O_Abandon, O_Suspended, O_Disconnect, and T_Disconnect.

Transitions from the "Call on Hold" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OAbandon, ODisconnect or TDisconnect message for *leg* c.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1 or *leg* p2.
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).

- To the "Stable 2-Party" CVS:
  - Processing of a valid DisconnectLeg message for *leg* p1 received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid DisconnectLeg message for *leg* p1 received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1.
  - Processing of a valid DisconnectLeg message for *leg* p1 received from the SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
  - Processing of a valid message containing DisconnectLeg for *leg* p2, plus MergeCallSegments, plus Reconnect received from SCF, in response to an OAbandon, ODisconnect, TSuspended or TDisconnect message for *leg* c.
  - Processing of a valid MergeCallSegments message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p2.
- To the "Call on Hold" CVS:
  - Processing of a valid MoveLeg message for *leg* c received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid MoveLeg message for *leg* c received from SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or an user to service information).
- To the "Stable M-Party" CVS:
  - Processing of a valid MergeCallSegments message received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid MergeCallSegments message received from SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Transfer" CVS:
  - Processing of a valid MergeCallSegments message received from SCF, in response to an OAbandon, ODisconnect or TDisconnect message for *leg* c.
  - Processing of a valid message containing DisconnectLeg for *leg* c, plus MergeCallSegments received from SCF, in response to an OAnswer message for *leg* p2.

4.3.3.7.7 "Call Waiting" Connection View State

The "Call Waiting" CVS represents two associated call segments. The controlling user is participating in a call that is in the stable or clearing phase, and another call is terminating to the controlling user.

Call processing in *CS* 2 is in the terminating setup phase in call waiting.

CS 1 represents an originating or terminating two-party call in the stable or clearing phase.

a) *Relationship with the* BCSM:

CS = 2:

controlling *leg*:

legStatus = pending

passive *leg* = p2:

legStatus = joined

bcsmPic = terminating call setup (Present_Call and T_Alerting only)

CS = 1:

controlling *leg* = c:

legStatus = joined

passive *leg* = p1:

legStatus = joined

bcsmPic = originating stable call; originating call clearing; terminating stable call; terminating call clearing

b) *Entry Events*: See the exit events for the "Terminating Setup" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events:
O_Mid_Call, T_Mid_Call, O_Abandon, O_Suspended, O_Disconnect, and T_Disconnect.

Transitions from the "Call Waiting" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OAbandon, ODisconnect or TDisconnect message for *leg* c.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1.
  - Processing of a valid ReleaseCall message received from SCF, in response to a TDisconnect message for *leg* p2.
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Stable 2-Party" CVS:
  - Processing of a valid DisconnectLeg message for *leg* p2 received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid DisconnectLeg message for *leg* p2 received from the SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
  - Processing of a valid message containing DisconnectLeg for *leg* p1, plus MergeCallSegments, plus Reconnect received from SCF, in response to an OAbandon, ODisconnect, TSuspended or TDisconnect message for *leg* c.
  - Processing of a valid DisconnectLeg message for *leg* p2 received from SCF, in response to a TDisconnect message for *leg* p2.
- To the "Call on Hold" CVS:
  - Processing of a valid MoveLeg message for *leg* c received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid MoveLeg message for *leg* c received from SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Forward" CVS:
  - Processing of a valid Connect message received from SCF, in response to an OMidCall or TMidCall message.

– Processing of a valid Connect message received from SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).

NOTE – AnalyseInformation and SelectRoute are allowed only in response to originating events.

Note that the Connect message causes the disassociation of the two CSs: the transaction remains associated with *CS* 2 (which is then considered to be in the "Forward" CVS), while *CS* 1 becomes a two-party call outside the context of IN (i.e. is no longer associated with the transaction).

4.3.3.7.8 "Stable M-Party" Connection View State

The "Stable M-Party" CVS represents a stable or clearing multi-party call in one call segment. The characteristics of the "Stable M-Party" CVS follow:

a) *Relationship with the* BCSM:

controlling *leg* = c:
legStatus = joined

passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating stable call; terminating call clearing

passive *leg* = p2:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating stable call; terminating call clearing

b) *Entry Events*: See the exit events for the "Call on Hold" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events:
O_Mid_Call, T_Mid_Call, O_Abandon, O_Suspended, O_Disconnect, and T_Disconnect.

Transitions from the "Stable M-Party" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OAbandon, ODisconnect or TDisconnect message for *leg* c.
  - Processing of a valid ReleaseCall message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1 or *leg* p2.
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Stable 2-Party" CVS:
  - Processing of a valid DisconnectLeg message for *leg* p1 or *leg* p2 received from SCF, in response to an OMidCall or TMidCall message.
  - Processing of a valid DisconnectLeg message for *leg* p1 received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1.

– Processing of a valid DisconnectLeg message for *leg* p2 received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p2.

– Processing of a valid DisconnectLeg message for *leg* p1 or *leg* p2 received from the SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).

- To the "Call on Hold" CVS:

  – Processing of a valid SplitLeg message for *leg* p1 or *leg* p2 received from SCF, in response to an OMidCall or TMidCall message.

  – Processing of a valid SplitLeg message for *leg* p1 or *leg* p2 received from the SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).

- To the "Transfer" CVS:

  – Processing of a valid DisconnectLeg message for *leg* c received from SCF, in response to an OMidCall or TMidCall message.

  – Processing of a valid DisconnectLeg message for *leg* c received from SCF, in response to an OAbandon, ODisconnect or TDisconnect message for *leg* c.

  – Processing of a valid DisconnectLeg message for *leg* c received from the SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).

4.3.3.7.9 "Transfer" Connection View State

The "Transfer" CVS represents a transferred call. The call between the two passive legs is in the stable or clearing phase. Note that the "surrogate" *legStatus* for the controlling leg indicates the charging relationship between the two passive legs after the call has been transferred. The characteristics of the "Transfer" CVS follow:

a) *Relationship with the* BCSM:

controlling *leg*:
legStatus = surrogate

passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating stable call; terminating call clearing

passive *leg* = p2:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating stable call; terminating call clearing

b) *Entry Events*: See the exit events for the "Call on Hold,", "Stable M-Party," and "Forward" CVSs.

c) *Exit Events*:

Summary of CVS-related SSF events: O_Suspended, O_Disconnect, and T_Disconnect.

Transitions from the "Transfer" CVS:

- To the "Null" CVS:

  – Processing of a valid ReleaseCall message received from SCF, in response to an OSuspended, ODisconnect or TDisconnect message for *leg* p1 or *leg* p2.

– Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or an user to service information).

- To the "Stable 2-Party" CVS:
  – In response to an O_Disconnect or T_Disconnect message for *leg* p1 or *leg* p2, processing of a valid MoveCallSegments message (to move the CS in a "Transfer" CVS from a source CSA to a target CSA), followed by a valid MergeCallSegments message on the target CSA (to merge two CSs in the target CSA into a "Stable 2-Party" CVS).
  – Processing of a valid DisconnectLeg message for *leg* p1 or *leg* p2 for the CS in a "Transfer" CVS, followed by a valid MoveCallSegments message (to move the CS in a "Transfer" CVS from a source CSA to a target CSA), followed by a valid MergeCallSegments message on the target CSA (to merge two CSs in the target CSA into a "Stable 2-Party" CVS), initiated by the SCF.
  – Processing of a valid DisconnectLeg message for *leg* p1 or *leg* p2 for the CS in a "Transfer" CVS, followed by a valid MoveCallSegments message (to move the CS in a "Transfer" CVS from a source CSA to a target CSA), followed by a valid MergeCallSegments message on the target CSA (to merge two CSs in the target CSA), followed by a Reconnect message (to reestablish a "Stable 2-Party" CVS with a control party that has gone on-hook).
  – Processing of a valid DisconnectLeg message for *leg* p1 or *leg* p2 for the CS in a "Transfer" CVS, followed by a Reconnect message (to re-establish a "Stable 2-Party" CVS with a control party that has gone on-hook).
- To the "Forward" CVS:
  – Processing of a valid message containing DisconnectLeg for *leg* p2, plus AnalyseInformation or SelectRoute or Connect received from SCF, in response to an OSuspended message for *leg* p2.
  – Processing of a valid AnalyseInformation, SelectRoute or Connect message received from SCF, in response to an ODisconnect message for *leg* p2.
  – Processing of a valid Connect message received from SCF, in response to a TDisconnect message for *leg* p2.
  – Processing of a valid message containing DisconnectLeg for *leg* p2, plus AnalyseInformation or SelectRoute or Connect received from SCF, initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or an user to service information).

    NOTE – AnalyseInformation and SelectRoute are allowed only in response to originating events.

4.3.3.7.10 "Forward" Connection View State

The "Forward" CVS represents a forwarded call. Call processing for the first passive leg (*leg* p1) is in a stable or clearing phase, or a terminating call setup phase, whereas call processing for the second passive leg (*leg* p2) is in an originating call setup phase. Note that the "surrogate" *legStatus* for the controlling leg indicates the charging relationship for the forwarded passive leg (*leg* p2). The characteristics of the "Forward" CVS follow:

a) *Relationship with the* BCSM:

controlling *leg*:
legStatus = surrogate

passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable call; originating call clearing;
terminating call setup,
terminating stable call, terminating call clearing

passive *leg* = p2:
legStatus = pending
bcsmPic = originating call setup (Analyse_Information, Select_Route and Authorize_Call_Setup only)

b) *Entry Events*: See the exit events for the "Stable 2-Party," "Terminating Setup," "Call Waiting," "Transfer," and "Stable 1-Party" CVSs.

c) *Exit Events*:

Summary of CVS-related SSF events: Call Setup Authorized.

Transitions from the "Forward" CVS:

- To the "Null" CVS:
  - Processing of a valid ReleaseCall message received from SCF which was initiated by the SCF (caused by, e.g. a charging notification from the SSF, an SCF time supervision or a user to service information).
- To the "Transfer" CVS:
  - Detection of the Call Setup Authorized event for *leg* p2, as described in Authorize_Call_Setup.

4.3.3.7.11 "Originating Setup M-Party" Connection View State

While in the "Originating Setup M-Party" CVS the connection view will associate the relevant call segments. The controlling user is in the setup phase (e.g. connected to an SRF) and the SRF has originated a new call which has not yet reached the stable state.

CS 2 represents an originating two-party in the setup phase.

CS 1 represents an originating two-party in the setup phase.

a) *Relationship with the* BCSM:

CS = 1:
controlling *leg* = c:
legStatus = joined
passive *leg* = p1:
legStatus = pending
bcsmPic = originating call setup

CS = 2:
controlling *leg*:
legStatus = surrogate
passive *leg* = p2:
legStatus = pending
bcsmPic = originating call setup

b) *Entry Events*: See the exit events for the "Originating Setup" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events: Call Setup Authorized.

Transitions from the "Originating Setup M-Party" CVS:

- To the "Active M-Party" CVS:
  - Detection of the Call Setup Authorized event for *leg* p2, as described in Authorize_Call_Setup.

4.3.3.7.12 "Active M-Party" Connection View State

By detection of an Call Setup Authorized event (e.g. when ISUP sends an IAM message) the "Originating Setup M-Party" state will transit to the "Active M-Party" state.

CS 1 represents an originating two-party in the setup phase.

CS 2 represents a stable two-party call.

a) *Relationship with the* BCSM:

CS = 1:
controlling *leg* = c:
legStatus = joined
passive *leg* = p1:
legStatus = pending
bcsmPic = originating call setup

CS = 2:
controlling *leg*:
legStatus = surrogate
passive *leg* = p2:
legStatus = joined
bcsmPic = originating stable call

b) *Entry Events*: See the exit events for the "Originating Setup M-Party" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events: OAnswer.

Transitions from the "Active M-Party" CVS:

- To the "Stable 2-Party" CVS:
  - Processing of a valid message containing DisconnectLeg for *leg* p1, plus MergeCallSegments received from SCF, in response to an OAnswer message.

4.3.3.7.13 "1-Party Setup" Connection View State

"1-Party Setup CVS represents a 1-party call being originated on behalf of the network (i.e. the controlling leg has a *legStatus* = surrogate).

a) *Relationship with the* BCSM:

controlling *leg*:
legStatus = surrogate
passive *leg* = p1:
legStatus = pending
bcsmPic = originating call setup

b) *Entry Events*: See the exit events for the "Null" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events: Call Setup Authorized.

Transitions from "1-Party Setup" CVS:

- To the "Stable 1-Party" CVS:
  - Detection of the Call Setup Authorized event as described in Authorize_Call_Setup.

4.3.3.7.14 "Stable 1-Party" Connection View State

"Stable 1-Party" CVS represents a 1-party call originated on behalf of the network (i.e. the controlling leg has a *legStatus* = surrogate), that is in a stable or clearing phase.

a) Relationship with the BCSM:

controlling *leg*:
legStatus = surrogate
passive *leg* = p1:
legStatus = joined
bcsmPic = originating stable; originating call clearing

b) *Entry Events*: See the exit events for "1-Party Setup" CVS.

c) *Exit Events*:

Summary of CVS-related SSF events: O_Answer

Transitions from the "Stable 1-Party" CVS:

- To the "Forward" CVS
  - Processing of a valid Connect message received from SCF, in response to an OAnswer message.

4.3.3.8 Summary of Transition Events

Figure 4-27 summarizes the entry and exit events. This diagram does **not** include the exit events to the "Null" CVS, or the events that result in no transition. See the detailed descriptions of the transition events under each individual CVS for more details.

Table 4-8 contains a complete list of the transition events defined for each CVS, including entry and exit events, and "no transition" events.

The following abbreviations are used in Figure 4-27 and Table 4-28:

- Messages received by SSF from SCF:

AI	AnalyseInformation
CI	CollectInformation
CCSA	CreateCallSegmentAssociation
Conn	Connect
DL	DisconnectLeg
ICA	InitiateCallAttempt
MgC	MergeCallSegments
MvC	MoveCallSegments
ML	MoveLeg
Rec	Reconnect
RC	ReleaseCall
SL	SplitLeg
SR	SelectRoute

- Trigger-related notation:

"2dt"	second dial tone (SSF processing required for the Switch_Hook_Flash_Specified_Code triggers)
{text}	processing required for indicated IN trigger
SHFSC	Switch_Hook_Flash_Specified_Code trigger

- Other notation:

ui	user interaction (i.e. ConnectToResource, etc.)
√	switch-based transition
|	logical OR
<>	conditional (based on service logic or event to which SCF responds)
(c)	(logical label for controlling *leg*)
(p1), (p2)	(logical labels for passive *legs*)

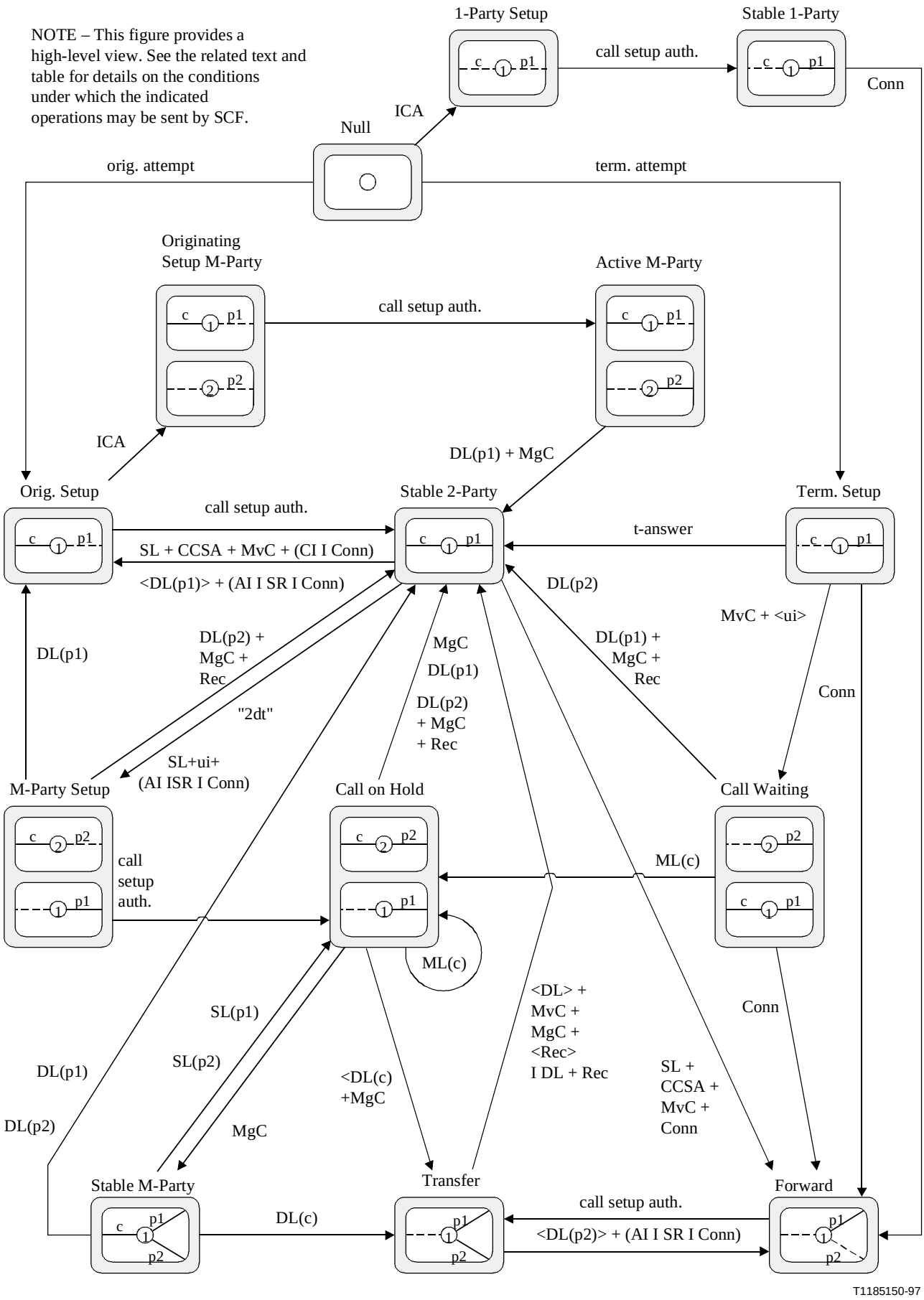


Figure 4-27/Q.1224 – Summary of entry and exit events

Table 4-8/Q.1224 – Transition events

From	To SSF event	Null	Orig. Setup	Stable 2-Pty	Term. Setup	M-Pty Setup	Call Hold	CW	Stable M-Pty	Trn.	Fwd.	Orig. Setup M-Pty	Active M-Pty	1-Pty Setup	Stable 1-Pty
Null	"orig. attempt"		√												
	"term. attempt"				√										
	[SCF initiated]													ICA	
Orig. Setup	"call setup auth."			√											
	CollectedInfo(c) AnalysedInfo(c)											ICA			
Stable 2-Pty	OMidCall	RC	DL(p1) + (AI | SR | Conn) SL + CCSA + MvCS + CI (Note 1)	SL + CCSA + MvCS + Conn (Note 1)		SL + ui + (AI | SR | Conn)					SL + CCSA + MvCS + Conn (Note 1)				
	TMidCall	RC	SL + CCSA + MvCS + CI (Notes 1, 2)	SL + CCSA + MvCS + Conn (Notes 1, 2)		SL + ui + Conn					SL + CCSA + MvCS + Conn (Note 1)				
	OAbandon(c) ODisconnect(c) TDisconnect(c)	RC													
	OSuspended(p1)	RC	DL(p1) + (AI | SR | Conn)												
	ODisconnect(p1)	RC	(AI | SR | Conn)												
	TDisconnect(p1)	RC	Conn												
	{flash+2dt: SHFSC}					√									
	[SCF initiated]	RC	DL(p1) + (AI | SR | Conn)			SL + ui + (AI | SR | Conn)									

Table 4-8/Q.1224 – Transition events *(continued)*

From	To SSF event	Null	Orig. Setup	Stable 2-Pty	Term. Setup	M-Pty Setup	Call Hold	CW	Stable M-Pty	Trn.	Fwd.	Orig. Setup M-Pty	Active M-Pty	1-Pty Setup	Stable 1-Pty
Term. Setup	TBusy	RC						MvC + <ui>			Conn				
	"t-answer"			√											
	[SCF initiated]	RC													
M-Pty Setup	OMidCall TMidCall	RC													
	OAbandon(c)	RC		DL(p2) + MgC + Rec											
	OSuspended(p1) ODisconnect(p1) TDisconnect(p1)	RC	DL(p1)												
	"call setup auth.(p2)"						√								
	[SCF initiated]	RC	DL(p1)												
Call Hold	OMidCall TMidCall	RC		DL(p1)			ML(c)		MgC						
	OAnswer(p2)									DL(c) + MgC					
	OAbandon(c) ODisconnect(c) TDisconnect(c)	RC		DL(p2) + MgC + Rec						MgC					
	TSuspended(c)			DL(p2) + MgC + Rec											
	OSuspended(p1) ODisconnect(p1) TDisconnect(p1)	RC		DL(p1)											
	OSuspended(p2) ODisconnect(p2) TDisconnect(p2)	RC		MgC											
	[SCF initiated]	RC		DL(p1)			ML(c)		MgC						

Table 4-8/Q.1224 – Transition events *(continued)*

From	To SSF event	Null	Orig. Setup	Stable 2-Pty	Term. Setup	M-Pty Setup	Call Hold	CW	Stable M-Pty	Trn.	Fwd.	Orig. Setup M-Pty	Active M-Pty	1-Pty Setup	Stable 1-Pty
CW	OMidCall TMidCall	RC		DL(p2)			ML(c)				Conn				
	OAbandon(c) ODisconnect(c) TDisconnect(c)	RC		DL(p1) + MgC + Rec											
	TSuspended(c)			DL(p1) + MgC + Rec											
	OSuspended(p1) ODisconnect(p1) TDisconnect(p1)	RC													
	TDisconnect(p2)	RC		DL(p2)											
	[SCF initiated]	RC		DL(p2)			ML(c)				Conn				
Stable M-Pty	OMidCall TMidCall	RC		DL(p1) DL(p2)			SL(p1) SL(p2)			DL(c)					
	OAbandon(c) ODisconnect(c) TDisconnect(c)	RC								DL(c)					
	OSuspended(p1) ODisconnect(p1) TDisconnect(p1)	RC		DL(p1)											
	OSuspended(p2) ODisconnect(p2) TDisconnect(p2)	RC		DL(p2)											
	[SCF initiated]	RC		DL(p1) DL(p2)			SL(p1) SL(p2)			DL(c)					
Trn.	OSuspended(p1)	RC													
	ODisconnect(p1) TDisconnect(p1)	RC		MvCS + MgCS (Note 1)											
	OSuspended(p2)	RC									DL(p2) + (AI | SR | Conn)				

Table 4-8/Q.1224 – Transition events *(concluded)*

From	To SSF event	Null	Orig. Setup	Stable 2-Pty	Term. Setup	M-Pty Setup	Call Hold	CW	Stable M-Pty	Trn.	Fwd.	Orig. Setup M-Pty	Active M-Pty	1-Pty Setup	Stable 1-Pty
	ODisconnect(p2)	RC		MvCS + MgCS (Note 1)							(AI | SR | Conn)				
	TDisconnect(p2)	RC		MvCS + MgCS (Note 1)							Conn				
	[SCF initiated]	RC		DL + MvCS + MgCS + <Rec> (Note 1) DL + Rec (Note 1)							DL(p2) + (AI | SR | Conn)				
Fwd.	"call setup auth.(p2)"									√					
	[SCF initiated]	RC													
Orig. Setup M-Pty	"call setup auth.(p2)"												√		
Active M-Pty	OAnswer			DL(p1) + MgC											
1-Pty Setup	"call setup auth."														√
Stable 1-Pty	OAnswer										Conn				

NOTE 1 – Supports the hybrid approach.

NOTE 2 – This transition is valid at a local exchange only and not at transit exchange.

4.3.4 The Hybrid Approach

4.3.4.1 General

The Hybrid Approach is complementary to the Connection View State Approach as described in the previous subclause. As with the Connection View State Approach, the Hybrid approach relies on implementations of the four core call party handling capabilities. But, the Hybrid Approach combines these four core capabilities with only a minimal set of Connection View states within the SSF/CCF. In this case an external resource (e.g. SRF bridging resource) is used to provide additional call party handling capability.

Since this Hybrid Approach only requires the four core SSF/CCF capabilities, only minimal enhancements are required at the SSF/CCF beyond IN CS-1. In cases where certain Connection View states cannot be supported by the SSF/CCF, the Hybrid Approach provides for call party handling where a strictly SSF/CCF-oriented approach leaves off.

Also, the Hybrid Approach provides a flexible system by working with Connection View states within the SSF/CCF. For example, Connection View states within the SSF/CCF can be used for simple situations such as two-party call party handling up to establishing a three way call. But for slightly more complex scenarios such as a three-way call with a fourth party in call-waiting, the Hybrid Approach can add to these capabilities and provide for access to a bridging resource in an external platform.

For simple two-party scenarios the efficiency costs of connecting the call parties to the external resource may outweigh the benefits gained. Therefore, this approach makes use of the internal call party handling capabilities of the SSF/CCF. Not all calls need to be routed to the resources, only the ones that require call party handling capabilities above and beyond those provided by the SSF/CCF. Also, the fourth core capability, as described in a previous subclause, provides for the call to be returned to the SSF/CCF when the services of external resources are no longer required. The Hybrid Approach includes the following advantages:

1) minimal enhancements to the SSF/CCF;
2) efficient use of specialized resources, resource only used on an "as needed" basis;
3) support for any call party handling feature;
4) provides a clean slate for call party handling techniques within the resource;
5) minimizes interaction problems with embedded base of switch-based features;
6) compatible with Connection View states in the SSF/CCF;
7) extendible to more complex scenarios;
8) extendible to a hybrid narrow-band-broadband environment;
9) makes use of current capabilities and implementations.

The following assumptions hold for the Hybrid Approach in general:

1) the SRF assigns and maintains a pool of DNs to which calls can be routed;
2) the SCF maintains correlators consisting of the CSAid and CSid at the application level on a per call basis. The SCF can determine the CSAid from the specific instance of the FE relationship for subsequent information flows. Service logic within the SCF can make use of these correlators for features which span a single call instance;
3) the SSF/CCF is able to match an application level correlator to a particular call and maintain that association;

4) the SSF/CCF is able to separate call parties involved in a single call, thereby separating complex calls (e.g. call hold) into multiple calls that are smaller and more manageable;

5) the SSF/CCF is able to merge multiple calls into a single call;

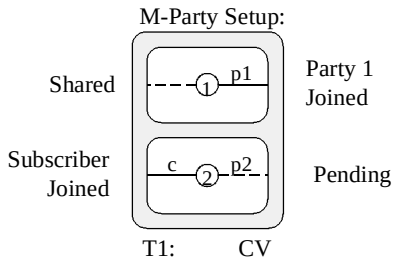
6) at the midcall, a user may enter a DN to place an outgoing call or enter a feature code which can be sent to the SCF.

4.3.4.2 SSF/CCF and SRF Connection Scenarios for the Hybrid Approach

The Hybrid Approach allows for connections to be supported between users, the SSF/CCF and the resource in a deterministic manner. It is possible to identify the Connection View states that enable particular connections, or as a result of connections that have been made.

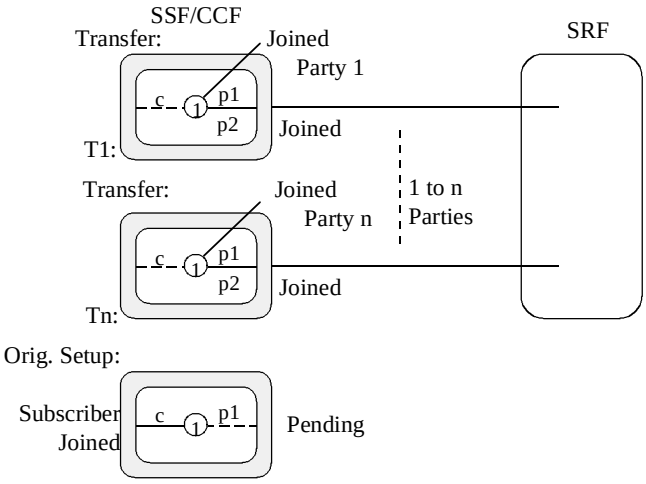
The following figure contains the set of connections that are used for the Hybrid Approach. In this figure, "Tx" refers to an open transaction between the SSF/CCF and the SCF. Also, "CV" refers to those components that illustrate the connection view which the SSF/CCF will provide to the SCF. The connection scenarios provide illustration for up to four call parties. Connections for more than four parties are provided for in a similar manner.

Connection Scenario 1



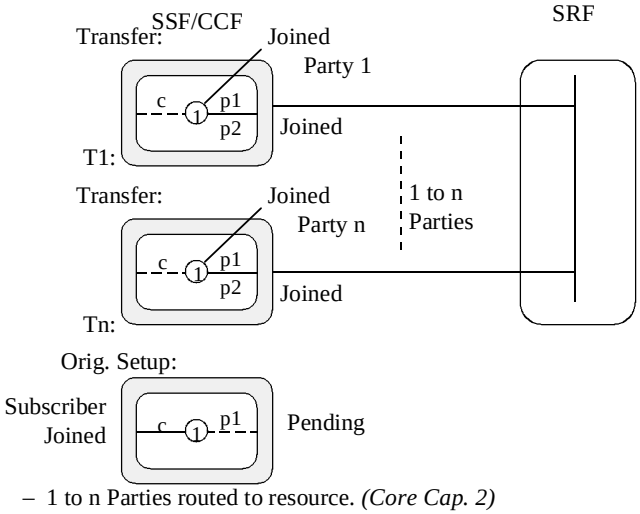
– Party 1 placed on SSF/CCF based hold. *(Core Cap. 1)*
– SSF/CCF provides dial tone to Subscriber. *(Core Cap. 1)*
– Digit collector is connected to Subscriber. *(Core Cap. 1)*
– Single Transaction T1 exist.

Connection Scenario 2



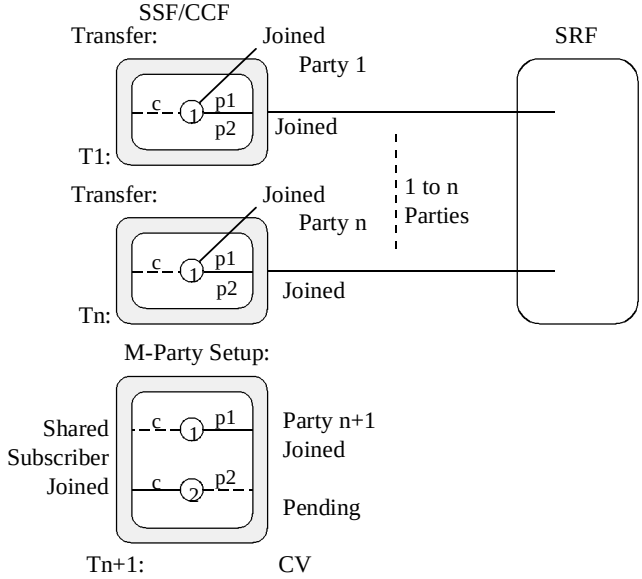
– 1 to n Parties routed to resource. *(Core Cap. 2)*
– Subscriber is given dial tone, call setup.

Connection Scenario 3



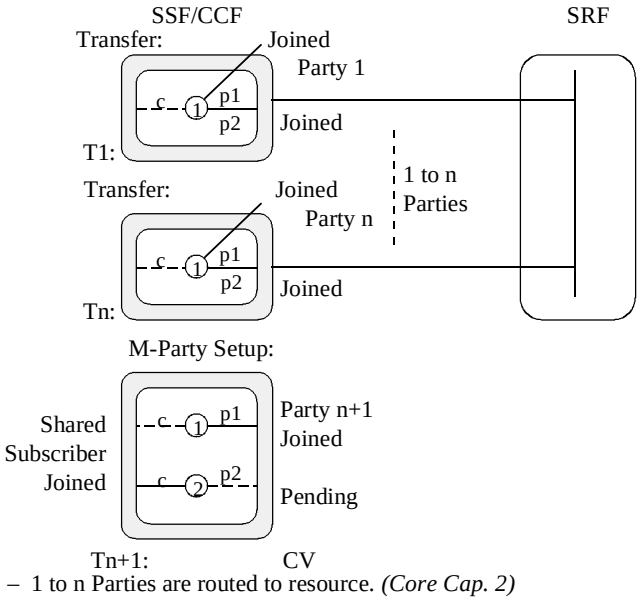
– 1 to n Parties routed to resource. *(Core Cap. 2)*
– 1 to n Parties connected at resource.
– Subscriber is given dial tone, call setup.

Connection Scenario 4



– 1 to n Parties are routed to resource. *(Core Cap. 2)*
– Party n+1 placed on SSF/CCF based hold. *(Core Cape. 1)*
– Subscriber given dial tone and digit collector at midcall trigger.

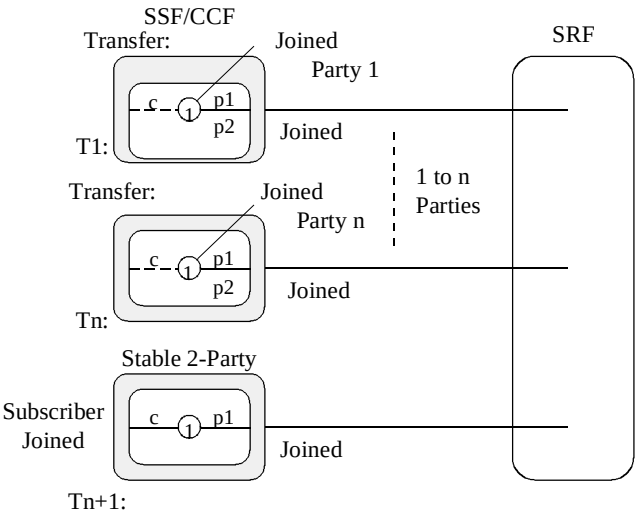
Connection Scenario 5



– 1 to n Parties are routed to resource. *(Core Cap. 2)*
– 1 to n Parties connected at resource.
– Party n+1 placed on SSF/CCF based hold. *(Core Cap. 1)*
– Subscriber given dial tone and digit collector at midcall trigger.

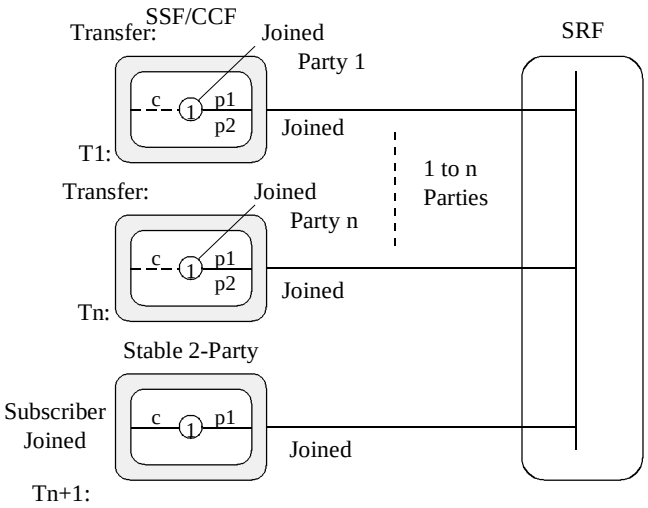
T1185160-97

Connection Scenario 6



– 1 to n Parties and subscriber are routed to resource.
(Core Cap. 2)

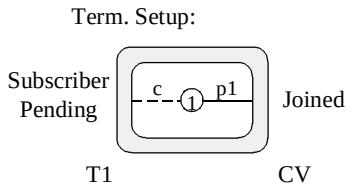
Connection Scenario 7



– 1 to n Parties and subscriber are routed to resource.
(Core Cap. 2)
– 1 to n Parties or subscriber connected at resource.

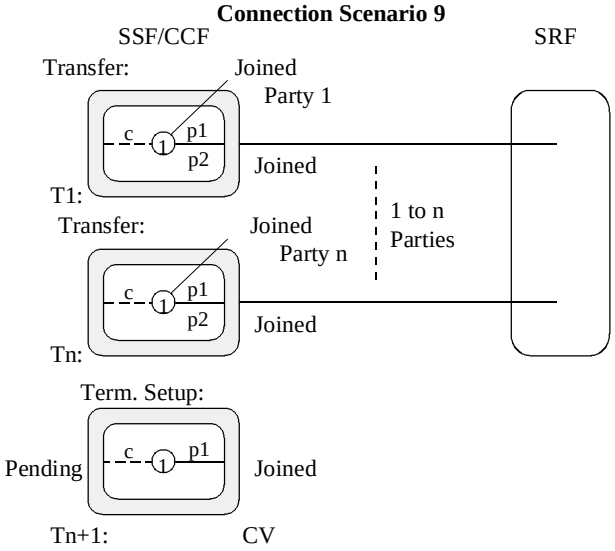
T1185170-97

Connection Scenario 8



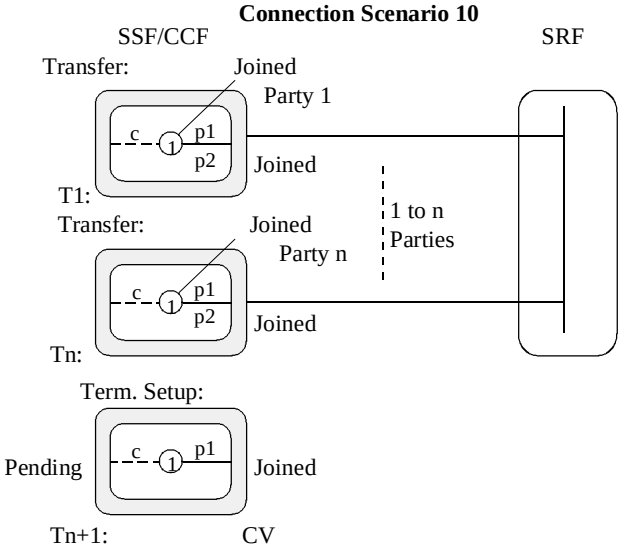
– Incoming call for subscriber

Connection Scenario 9



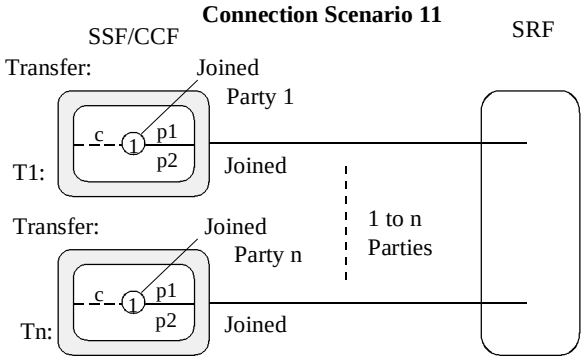
– 1 to n Parties routed to resource. *(Core Cap 2)*
– Incoming call for subscriber.

Connection Scenario 10



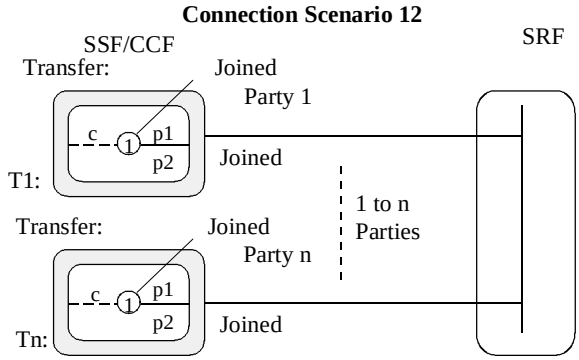
– 1 to n Parties routed to resource. *(Core Cap 2)*
– 1 to n Parties connected at resource.
– Incoming call for subscriber.

Connection Scenario 11



– 1 to n Parties routed to resource. *(Core Cap 2)*

Connection Scenario 12



– 1 to n Parties routed to resource. *(Core Cap 2)*
– 1 to n Parties connected at resource.

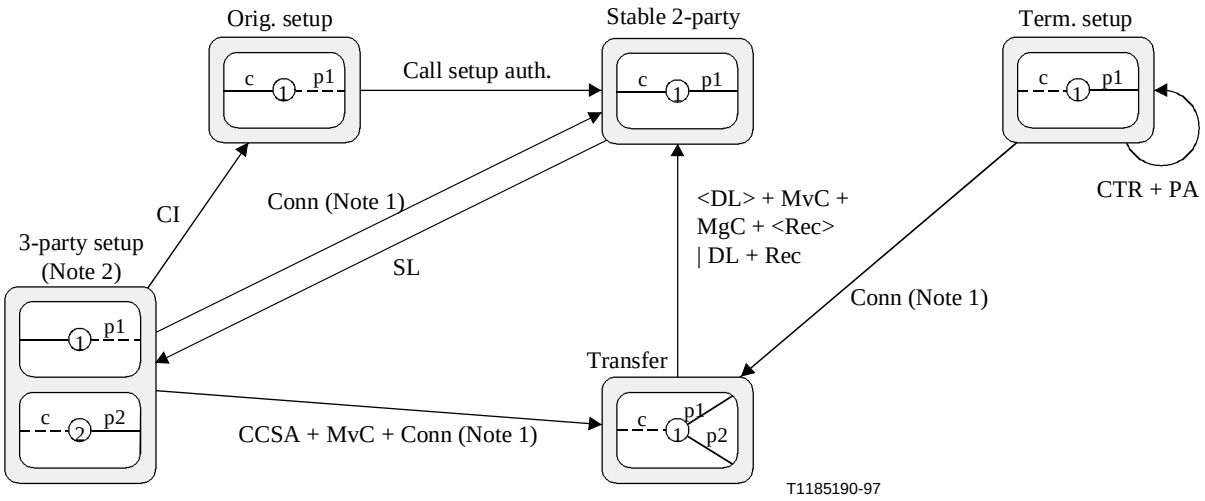
T1185180-97

4.3.4.3 The catalogue of Connection View States and transitions for the Hybrid Approach

The Hybrid Approach does not rely on all of the Connection View states and the transitions as described in the previous subclause, some of the call party handling capability will be realized in a platform that is external to the SSP. Figure 4-28 illustrates the catalogue of SSF/CCF Connection View states required for the Hybrid Approach, along with the required transitions. Transitory configurations (e.g. Forward) are not illustrated.

The following abbreviations are used for the operations in Figure 4-28:

CCSA	CreateCSA
CI	CollectInformation
Conn	Connect
CTR	ConnectToResource
DL	DisconnectLeg
MgC	MergeCallSegments
MvC	MoveCallSegments
PA	Play Announcement
Rec	Reconnect
SL	SplitLeg



<> Conditional (depends on event to which SCF responds)
| Logical OR

NOTE 1 – An EDP should be set before each Connect to ensure each call sent to an SRF is monitored for a disconnect by the SSF.
NOTE 2 – CCSA + MvC sent once to separate CS-1 and CS-2 from same CSA prior to CI or Conn operations.

Figure 4-28/Q.1224 – Catalogue of Connection View States for the Hybrid Approach

4.3.5 IN-SSM EDPs

Certain IN-SSM events can be reported to active IN service logic instances that have already been invoked. These events are referred to as IN-SSM EDPs. For example, events such as the successful completion or failure of a particular IN-SSM function may need to be reported. Detection of IN-SSM EDPs does not lead to the invocation of additional IN service logic instances. IN-SSM EDPs are handled implicitly for IN CS-2 for those information flows from the SCF that require confirmation by the SSF/CCF.

4.3.6 SSF resource control

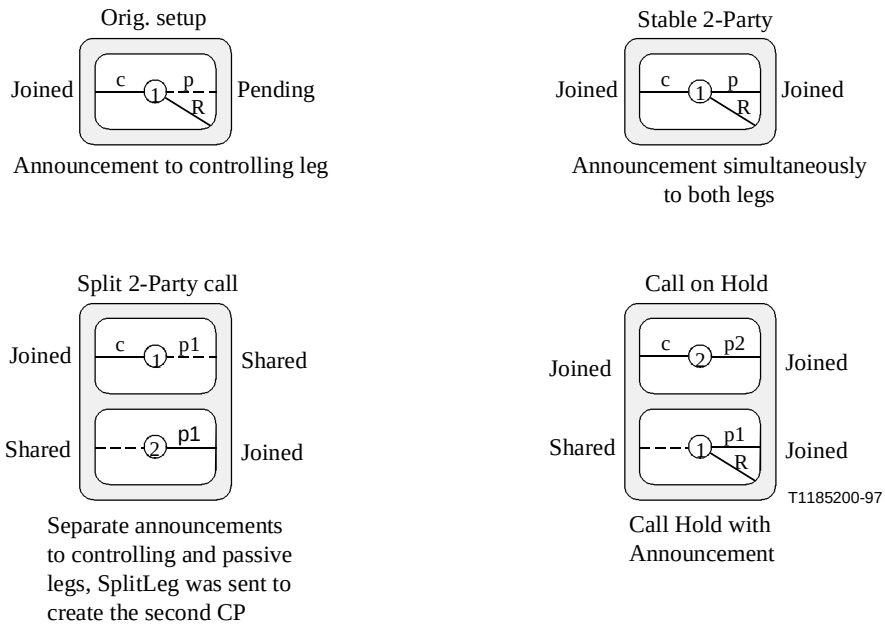
Local and remote specialized resources needed to perform IN call/service processing are accessible to the SSF/CCF. The treatment of specialized resources with respect to the Call Segment Association was described above. Objects that explicitly represent specialized resources are not explicitly

modelled for IN CS-2, though they are implied by other information flows/information elements defined for IN CS-2.

4.3.6.1 SSF Resource Control for announcements and digit collection

In the Connection View model, SRF connections for announcements and digit collection (relay as well as assisting) are always established from a Connection Point, as shown in the figures below.

Examples



An SRF can be connected to any CP which has at least one joined leg.

If the CP has more than one leg joined to it, information can be sent/received from all legs that have a bearer connection with the leg where the SRF is connected. Within a CS, bearer connections exist between all legs where the corresponding BCSM is in the O/T_Active PIC or at the O/T_Mid_Call DP. If the User Interaction is intended for one of these parties only, the concerned leg can be separated from the CP by means of a SplitLeg IF. This will break the bearer connection between this leg and the other legs left at the CP. The split leg will now be connected to its own CP. Note that in the split two-party call example above, there is still only one passive leg (shared between Call Segments 1 and 2), therefore still only 1 instance of a BCSM. After User Interaction has been completed and the resource has been disconnected, the speech connection between the legs can be re-established by means of a MergeCall IF, which will connect the split leg to the same CP with the other leg(s) again and remove the second CP.

With IN CS-1, SRF connections can only be made while call processing is suspended. This is allowed at any DP. With IN CS-2, SRF connections can also be made while call processing is not suspended, e.g. to allow detection and reporting of mid call events. For User Interaction during call processing, none of the legs connected to the CP may be in setup, i.e. all BCSM instances associated with the CS shall be in the O/T_Active or O/T_Suspended PIC, in order to avoid interference between call setup and user interaction. If during user interaction an EDP-R is detected, call processing can be resumed by sending a ContinueCS IF.

When an SRF connection is made while call processing is suspended, i.e. in response to a TDP-R or EDP-R, all subsequent call processing IFs received from SCF will not be executed until end of user

interaction, with the exception of the ContinueCS IF. All IFs leading to release of a CP, such as ReleaseCall or MergeCall will also not be executed until end of user interaction on that CP.

The Release of the CP by any other entity than SCF, i.e. Abandon/Disconnection of last leg, will release the SRF.

4.4 Feature Interactions Manager (FIM)/Call Manager (CM)

A brief description of the FIM is provided in 4.1. The particular FIM subjects described below include FIM/CM functionality and service logic instance interactions aspects. A high-level description of these subjects is provided below.

4.4.1 FIM/CM Functions

As described in 4.3.1, an IN CS-2 connection control IN-SSM provides an SCF with an abstract view of a single two-party or multi-party call segment, or of a pair of associated call segments. As such, the SCF can control multiple communication paths and connections, supported by multiple BCSMs. Overall management of these various elements of call segments is provided by the CM functionality. The CM interacts with the BCM and IN-SM to:

a) coordinate event reporting among multiple BCSMs for a given IN CS-2 connection control IN-SSM (e.g. event reporting when the same event is detected in multiple BCSMs simultaneously, such as "hook-flash", DTMF # or *XX, or when different events are detected in multiple BCSMs simultaneously, such as "hook-flash" from one party and "disconnect" from another);

b) coordinate the suspension and resumption of BCSM processing among multiple BCSMs for a given IN CS-2 connection control IN-SSM (e.g. when an event is detected in a BCSM for which the BCM requires further instructions on how to proceed, processing of all BCSMs for that IN CS-2 connection control IN-SSM may need to be halted);

c) enforce rules and restrictions applicable to an IN CS-2 connection control IN-SSM (e.g. rules and restrictions on when and how the SCF can manipulate legs, associate a pair of call segments, and merge a pair of associated call segments).

FIM functionality is described below.

d) The FIM should provide a service logic instance selection mechanism to determine which service logic instance to invoke at a DP. This mechanism should select the appropriate IN service logic instance or non-IN service logic instance, and for IN CS-2, may block the invocation of any other service logic instances for that particular DP (see 4.4.3).

e) The FIM may not always allow simultaneously active IN and non-IN service logic instance for IN CS-2 that control the call/connection. There are both static and dynamic mechanisms of realizing this restriction. The static mechanism may involve service management functionality (e.g. via service provisioning), whereas the dynamic mechanism may involve more complex FIM capabilities. For IN CS-2, the simplest mechanism should be implemented (see 4.4.2).

f) The FIM should provide mechanisms to support simple, restricted service logic instance interactions between simultaneously active service logic instances from different SCFs acting on the same call segment (see 4.4.3).

4.4.2 Service logic instance interactions considerations

It is recognized that services provided by an IN-structured network will be composed of one or more service features, which are constructed from one or more reusable units of capabilities (e.g. SIBs) provided to users by the network. It is also recognized that one or more service features may be

simultaneously active on a single call. Finally, it is recognized that both IN service features and non-IN service features may be simultaneously active on a single call. A service feature interactions mechanism is needed to manage the potential interactions (both desirable and undesirable) between such service features. Given that these service features are realized by service logic instances, this mechanism needs to be described in terms of rules and procedures relative to triggering, compatibility, precedence, invocation, execution, and event reporting for multiple service logic instances. This subclause addresses the static and dynamic aspects of service logic instance interactions management, as well as mechanisms for determining compatibility and precedence.

a) *Static and dynamic aspects*

There are two aspects to service logic instance interactions management, to include the static and dynamic aspects. These two aspects are discussed below.

- *Static aspects*

  The static aspects of service logic instance interactions management concerns the provision of service features to end users. To illustrate this, consider the following example: end user A already has service feature X, and it is known that service feature X and service feature Y are mutually incompatible; if an attempt is made to provision the invocation of a service logic instance for service feature Y to end user A using OAM procedures, then this attempt should be rejected.

- *Dynamic aspects*

  There are three items to be considered under the dynamic aspects of service logic instance interaction management.

  – If at a particular DP there is more than one service logic instance which can be invoked, then a decision must be made as to which of these service logic instances will be invoked first (i.e. service logic instance selection).

  – If a service logic instance can be invoked, then a decision must be made as to whether or not the new service logic instance is compatible with any service logic instances already active on the same call segment.

  – If the new service logic instance is compatible with any service logic instances already active on the same call segment, then a decision must be made as to its precedence for call processing events (such as signalling messages) with respect to other active service logic instances; if the new service logic instance is incompatible, it should be blocked.

  For the latter two items, there are at least two potential approaches to service logic instance interactions management.

  – The first approach is to make decisions as part of DP processing; with this approach, decisions about service logic instance compatibility and precedence are made *before* a service logic instance is invoked.

  – The second approach is to make decisions independent of DP processing; with this approach, decisions about service logic instance compatibility and precedence are made *after* a service logic instance is triggered.

  The first approach is simpler, though restrictive, since it can prevent service logic instances from being invoked, only requiring the management of a limited number of service logic instance interactions. The second approach is more complex, though flexible, since it does not prevent service logic instance from being invoked, thus requiring a mechanism that can manage all possible service logic instance interactions. Due to this complexity, the second approach is considered beyond the scope of IN CS-2.

b) *Mechanisms for determining compatibility and precedence*

At present, knowledge concerning the compatibility of service features and their precedence is "hard-coded" in to the SSF/CCF. This mechanism relies on specifying each possible interaction for every possible combination of service features. As the number of service features gets large (one of the aims of IN), this specification quickly becomes complex, complicating the task of the service designer. Furthermore, as each new service feature is added, its many possible interactions must be identified and specific rules and data must be introduced into the SSF/CCF or SCF to specify how each interaction is to be resolved.

A more general mechanism than "hard-coding" would be a "data-driven" mechanism in which the service designer could specify service feature compatibility and precedence during service creation and provisioning. The service creation environment could provide the service designer with information about the specific service features for a particular subscriber, enabling the service designer to specify such things as which service features are blocked by a new service feature, the relative precedence of the new service feature to other service features, and the DP at which the service logic instance for the service feature should be invoked. The output of such a mechanism could be introduced directly into the SSF/CCF or SCF from the service creation environment.

The ultimate mechanism would be to use an expert system approach to reduce the burden on the service designer.

For IN CS-2, the existing mechanisms for service logic instance interactions management will have to be used beyond what is described herein for IN-IN service logic instance interactions in the SSF/CCF, and 4.4.2 c), 4.4.2 d), and 4.4.3 for IN-non IN service logic instance interactions in the SSF/CCF. That is, the interactions between service logic instances (both IN and non-IN) will have to be specified as part of the service feature description, with vendor-specific mechanisms to resolve remaining interactions in the specified manner. In addition, it may be possible to adopt a data-driven approach if mechanisms can be incorporated into the service creation environment to prompt the service designer for compatibility and precedence information, then download the appropriate data into the SSF/CCF or SCF. The expert system approach is considered to be beyond the scope of IN CS-2.

c) *IN and non-IN service logic instance interactions*

There are desirable and undesirable IN and non-IN service logic instance interactions in the SSF/CCF. Table 4-9 identifies these for IN CS-2.

Table 4-9/Q.1224 – IN and non-IN service logic instance interactions

<table>
<tr><th colspan="2" rowspan="3"></th><th colspan="4">Non-IN</th></tr>
<tr><th colspan="2" rowspan="2">Connection Control (CC)</th><th colspan="2">Non-connection control</th></tr>
<tr><th>Passing or using information</th><th>Notification</th></tr>
<tr><td rowspan="3">IN</td><td>CC</td><td colspan="2"></td><td>Can’t be independent</td><td>OK</td></tr>
<tr><td rowspan="2">Non-CC</td><td>Request</td><td>Restricted (e.g. translation)</td><td>• OK for passing information (e.g. CLID) or
• Need precedence if using same info or same DP</td><td>OK</td></tr>
<tr><td>Notification</td><td>OK</td><td>OK</td><td>OK</td></tr>
</table>

This table classifies IN and non-IN service logic instances first by whether or not they involve connection control (e.g. leg manipulation). Non-IN service logic instances that do not involve connection control are further classified by their involvement in call/service processing. This includes involvement in passing end-to-end information on a call (e.g. user-to-user information, calling number delivery) or using call-related information (e.g. for number translation), and involvement only in terms of receiving notification of call-related events (e.g. answer, disconnect). IN service logic instances are also further classified by their involvement in call/service processing. This includes involvement in terms of receiving requests and providing non-connection control instructions (e.g. Proceed call processing with new information), and involvement only in terms of receiving notification of call-related events. Based on these classifications, a matrix of interaction restrictions for IN CS-2 can be developed, as reflected in Table 4-9.

From the table, it is evident that IN service logic instances that involve connection control must be completely independent of non-IN service logic instances that involve connection control. This is a consequence of the single point of control constraint for IN CS-2. Further, it is evident that IN and non-IN service logic instances that only involve notification of events can interact with any other type of IN and non-IN service logic instances, since these do not involve any type of control. The remaining interactions are restricted as follows:

- *IN CC vs. non-IN passing or using information* – In this case, the service logic instances cannot be processed independent of each other since IN service logic instances that involve connection control may prevent passing of end-to-end information by changing or interrupting connections.
- *IN non-CC request vs. non-IN CC* – In this case, IN service logic instances are restricted to those that only manipulate basic call-related information (e.g. for destination number translation), and do not change the flow of basic call processing (e.g. given that basic call processing is suspended while waiting for IN call handling instructions, processing resumes from the point at which it was suspended when instructions are received). In this case, IN service logic instances can be invoked to enhance non-IN connection control [see discussion in 4.4.2 d)].
- *IN non-CC request vs. non-IN passing or using information* – In this case, passing end-to-end information should be transparent to IN service logic instances. However, IN and non-IN service logic instances may be competing for the same call-related events or

information. Straightforward precedence and exclusion mechanisms can be used to resolve this contention for IN CS-2. These mechanisms are described in 4.4.3.

These restrictions are identified as guidelines to assist implementers in managing these types of interactions in a proprietary manner in those cases where mechanisms are not described in this Recommendation.

d) *Applying "Type A" IN technology to "Type B" services*

There are some circumstances in which it will be possible to apply "Type A" IN technology to certain aspects of "Type B" services. This applies to switch-based services in general, whether these services be of "Type A" or "Type B", and to "Type B" services in general, whether these be switch-based or CS-n based.

"Type A" services are characterized as "single-ended" and "single point of control". It also happens that IN CS-2 is limited to "single medium" (as opposed to "multi-media") services. By implication, "Type B" services differ from "Type A" services in at least one of the dimensions: (ends, points of control, media). Of main interest in the shorter term is variation of the number of ends affected. Some examples of "Type A" services are: Freephone, Virtual Private Network (VPN), Universal Personal Telecommunication (UPT), originating and terminating call screening, selective call forward on busy/don't answer, credit card calling, televoting, malicious call identification, and completion of call to busy subscriber. Currently defined "Type B" services are generally available using switch-based technology. It may be expected that equipment vendors will provide support and interworking of "Type A" and "Type B" services in their product portfolios. Such interworking will not necessarily be part of the standards for IN CS-2.

i) *Situations when "Type A" capabilities may be used with "Type B" or switch-based services*

In circumstances where a request for a "Type B" or switch-based service requires a check to see whether such a service may be performed, "Type A" technology may be applied before proceeding with the service.

In circumstances where several variations in a "Type B" or switch-based service are possible, a check to see which variation is to be performed may make use of "Type A" technology.

ii) *Determining when to use "Type A" capabilities*

In the active phase of a call, certain means for gaining the attention of the exchange (e.g. switch-hook flash) are context specific. In these circumstances, the context needs to be considered first to determine whether a "Type A" service request should occur.

For example, after receiving call waiting tone, a series of switch-hook flashes may be used to toggle between the two calls. In the absence of call waiting, a switch-hook flash may indicate a desire to add in a third party, with a subsequent switch-hook flash joining the three subscribers.

From these two cases, it can be seen that some care needs to be taken in determining whether or not it is appropriate to launch a "Type A" service query. In the example described, it would not be appropriate once the call waiting tone has been applied, nor after the waiting call has been answered. In the second case, it would be appropriate to see what should be done. Some options that could be indicated to the switch might be: ignore the switch-hook flash, proceed with normal three-party call, add-in fixed third party (e.g. supervisor), etc.

Taking full advantage of this approach will require some extension of the SSF-SCF interface to include identification of the specific service and instruction to proceed or not

with (standardized) services. The extent to which this can be standardized for IN CS-2 will depend on the time and resources available to do it as the standardization for the base capabilities to support "Type A" services proceeds.

iii) *Examples of services augmented by "Type A" capabilities*

- *Conference dial-in authorization*

  In this service, only authorized parties may dial-in to a conference bridge. Conferencing is, in general, a "Type B" service in that more than one end is involved when another subscriber joins the conference.

  An SSF supporting a conference capability, on receiving a request to join the conference, may use "Type A" technology to query an SCF for a list of authorized participants. This list would be updated through an OA&M process as conference reservations are made, and would include such things as: conference timings, participant identification, charging to be applied, etc. This list could even be updated in real time as the conference proceeds so that previously excluded subscribers may join in as directed by the conference "owner" or Chairman.

  In this way, a substantial degree of security may be added to a conference, especially one that is regularly held and at which sensitive information might be discussed.

- *Selective or distinctive call waiting*

  In order to determine whether a call waiting tone should be applied, the terminating exchange may consult an SCF for a screening list (inclusive or exclusive) to determine whether call waiting should be applied or whether alternative treatment should be given to the incoming call. In this way, "Type A" technology can be used to augment this service.

  In order to indicate certain special callers, a distinctive call waiting tone may be applied. "Type A" technology may be used to identify when this applies and, when there are several distinctive tones available, which should be applied. In this way, "Type A" technology can be used to augment this service.

4.4.3 FIM mechanisms

FIM mechanisms for IN CS-2 include precedence and priority mechanisms to manage the invocation of instances of IN and non-IN service logic, and exclusion mechanisms to manage the invocation of new instances of IN service logic when existing instances of IN service logic are still active. These mechanisms are described below.

a) *Precedence and priority*

Subclause 4.2.8 identifies the assumptions that a DP may be armed as both a TDP and EDP, and that a DP may be armed with multiple criteria, each for the invocation of a different instance of IN service logic. In addition, 4.4.2 identifies the additional assumption that a DP may be armed for instances of non-IN service logic, in addition to instances of IN service logic. These assumptions, along with the constraints for IN CS-2 identified in clause 2, form the basis of a set of precedence and priority rules that should be used when processing DP criteria. These rules are listed below:

i) when processing criteria for an armed DP, process criteria for a DP-Notification (DP-N) before a DP-Request (DP-R);

ii) when processing criteria for a DP-N or a DP-R, process criteria for EDPs before TDPs;

iii) when processing criteria for EDPs or TDPs, criteria processing rules for IN service logic and non-IN service logic must allow IN and non-IN service logic processing based upon priority of services;

iv) when processing criteria for IN or non-IN service logic, process criteria in priority order, as provisioned through administrative procedures.

Application of these rules may result in the following precedence ordering, with a priority ordering of multiple service logic instances at each level:

- *EDP-N for an instance of IN service logic* – A control or monitor relationship exists with an SCF for an existing IN service logic instance; the event detected at the DP is reported to the SCF in the context of the existing relationship and the next DP criteria is processed immediately. No response is expected from the SCF.
- *EDP-N for an instance of non-IN service logic* – The EDP is for an existing non-IN service logic instance in the SSF/CCF; the event detected at the DP is reported to the non-IN FM and the next DP criteria is processed immediately. No response is expected from the non-IN FM.
- *TDP-N for an instance of IN service logic* – The event detected at the DP is reported to the SCF via a new monitor relationship and the next DP criteria is processed immediately. No response is expected from the SCF.
- *TDP-N for an instance of non-IN service logic* – The TDP is for a non-IN service logic instance in the SSF/CCF; the event detected at the DP is reported to the non-IN FM and the next DP criteria is processed immediately. No response is expected from the non-IN FM.
- *EDP-R for an instance of IN service logic* – A control relationship exists with an SCF for an existing IN service logic instance; the event detected at the DP is reported to the SCF in the context of the existing control relationship. Call processing is suspended and a response is expected from the SCF.
- *EDP-R for an instance of non-IN service logic* – The EDP is for an existing non-IN service logic instance in the SSF/CCF; the event detected at the DP is reported to the non-IN FM. Call processing is suspended and a response is expected from the non-IN FM.
- *TDP-R for an instance of IN service logic* – No control relationship exists with an SCF; the event detected at the DP is reported to the SCF via a new control relationship. Call processing is suspended and a response is expected from the SCF.
- *TDP-R for an instance of non-IN service logic* – The TDP is for a non-IN service logic instance in the SSF/CCF; the event detected at the DP is reported to the non-IN FM. Call processing is suspended and a response is expected from the non-IN FM.

For those cases in which a response is expected and the response indicates that call processing should continue from the point at which it was suspended (i.e. the DP at which criteria were met and the event was reported), then the remaining DP criteria should be processed. If the response indicates that call processing should continue at a new point in call, then any remaining DP criteria at the point of suspension are not processed.

b) *Exclusion*

There is no explicit mechanism in IN CS-2 for managing exclusion of new instances of IN service logic when existing instances of IN service logic are still active. However, there are implicit exclusion mechanisms for IN CS-2 that have already been described. Subclause 4.2.9 describes rules that allow only one IN service logic instance at a time to control the same call segment (i.e. to send responses to the SSF/CCF). These rules exclude multiple IN service logic instances simultaneously controlling the same call segment, but do not exclude multiple IN service logic instances from receiving notification of events detected in a call segment that is being controlled by another IN service logic instance. In addition,

subclause 4.4.3 a) above describes precedence and priority rules for processing DP criteria for service logic instances. These rules identify that service logic instances at a lower precedence level or priority may not be invoked, depending on the disposition of previous service logic instances. This implies that DP criteria for multiple service logic instances at the same DP can be ordered in such a way as to manage this exclusion.

4.5 Relationship of SSF/CCF Model Components

4.5.1 General

Subclause 4.1 identifies relationships between the major components of the SSF/CCF model. The major relationships are those between the SCF and the IN-SM (via the SCF Access Function), between the IN-SM and the FIM/CM, and between the FIM/CM and the BCM. The relationship between the SCF and IN-SM is external to the SSF/CCF, and is a subject for standardization. The definition of this relationship follows the methodology described in clause 3/Q.1201. The other relationships are internal to the SSF/CCF, and are not subjects for standardization. These latter relationships are assumed to exist for explanatory purposes only, to better understand and describe the SSF/CCF model.

All of these relationships are described by the information flows between components. Information flows in the SSF/CCF model are identified in Figure 4-29, which only shows the relationships in half of Figure 4-1. These information flows are described below.

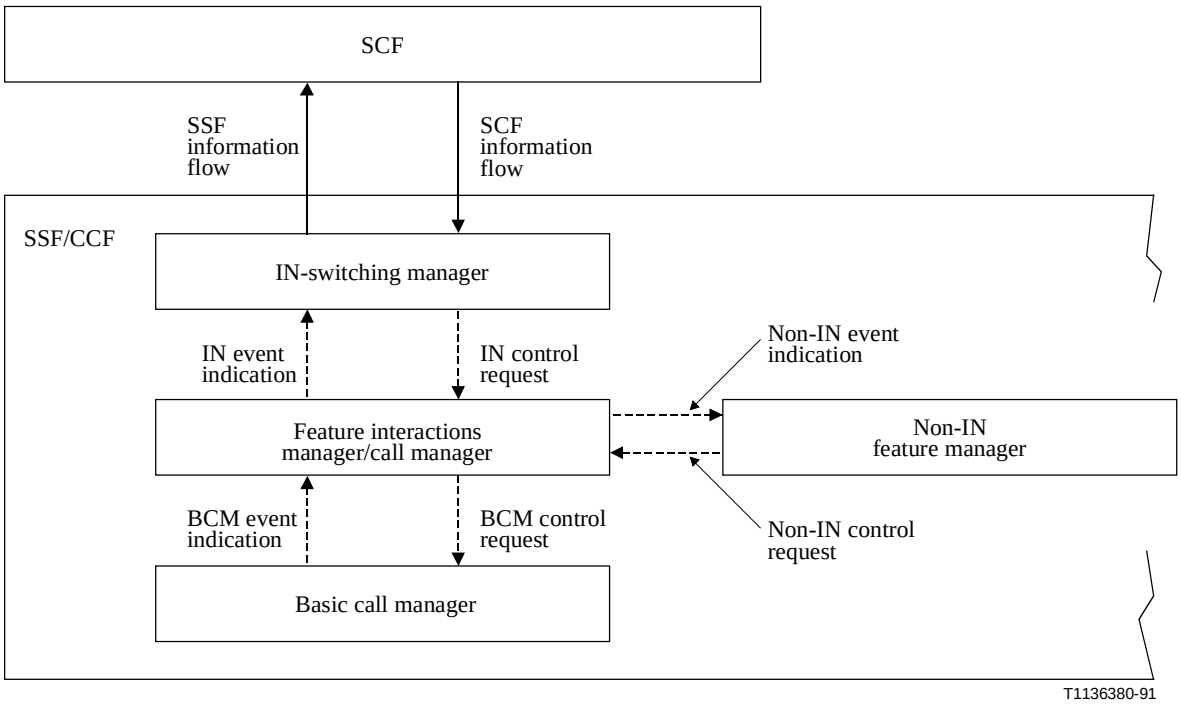


Figure 4-29/Q.1224 – SSF/CCF model information flows

a) *Relationship between SCF and IN-SM*

– *SSF information flow*: Information from the IN-SM to the SCF (via the SCF access manager in the SSF) that reports a call/connection processing event, as well as the current state of the call/connection instance in which the event is detected, as identified in clause 6.

– *SCF information flow*: Information from the SCF to the IN-SM (via the SCF access manager in the SSF) that requests the manipulation of the state of a call/service instance, invoking the types of functions identified in clause 6.

b) *Relationship between IN-SM and FIM/CM*

– *IN event indication*: Information from the FIM/CM to the IN-SM that reports a call processing event, the current state of the call in which the event is detected, and whether the event is to be handled by a new instance of IN service logic or an existing active instance.

– *IN control request*: Information from the IN-SM to the FIM/CM that indicates call/service processing functions requested by the SCF.

c) *Relationship between FIM/CM and BCM*

– *BCM event indication*: Information from the BCM to the FIM/CM that reports a BCSM event and the current state of the BCSM in which the event is detected.

– *BCM control request*: Information from the FIM/CM to the BCM that requests the manipulation of one or more BCSMs to influence call/service processing.

d) *Relationship between FIM/CM and non-IN FM*

– *Non-IN event indication*: Information from the FIM/CM to the non-IN FM.

– *Non-IN control request*: Information from the non-IN FM to the FIM/CM.

The use of these information flows is illustrated in the following subclause.

4.5.2 Typical sequence of model actions

This subclause describes a typical sequence of actions in the SSF/CCF model to illustrate the roles and relationships of the major model components. This illustration is not intended to imply or reflect any specific implementation. This scenario provides an example in which a new instance of an IN-SSM is invoked to provide an IN service feature to a user. At the start of this scenario, non-IN service logic instances of non-IN service logic instances are active, and there is no existing relationship between the SCF and SSF/CCF.

1) A user is interacting with the SSF/CCF via the CCAF to request the setup of a call. The BCM creates a BCSM to represent the basic call control functions required to establish and maintain this call for the user.

2) In the course of call setup for the user, an event is detected in the BCSM associated with the user's call. BCSM processing is halted at the DP.

3) The BCM processes the event at a DP in the BCSM to determine if the event should be reported (i.e. it determines if the DP is armed and DP criteria are met). If so, it sends a BCSM event indication reporting the event to the FIM/CM, along with the state of the BCSM at the time the event was detected. If the BCM needs instructions on how to proceed, BCSM processing remains halted at the DP until instructions are received. If not, the BCM continues normal BCSM processing. Thus, three scenarios are possible:

– the BCM determines that the event should not be reported; BCSM processing continues (e.g. no TDP armed);

– the BCM determines that the event should be reported, but does not need further instructions; BCSM processing continues (e.g. TDP-N armed);

– the BCM determines that the event should be reported, and needs further instructions (e.g. TDP-R armed); BCSM processing is halted and the BCM may continue to detect additional events before receiving instructions (the handling of these additional events is not addressed in this example).

4) The FIM/CM receives and processes the BCM event indication to determine if the event is to be processed by an IN service logic instance or a non-IN service logic instance. It also determines if the event is to be processed by a new instance of a service logic instance or an existing active instance.

5a) Assuming the BCM event is to be processed by a new instance of IN service logic, the FIM/CM sends an IN event indication to the IN-SM reporting the event, the state of the BCSM in which it was detected, and indicating that a new instance of IN service logic is to be invoked. Go to step 6.

5b) Assuming the BCM event is to be processed by a new instance of non-IN service logic instance, the FIM/CM sends a non-IN event indication reporting the event to the non-IN FM, the state of the BCSM in which it was detected, and indicating that a new instance of non-IN service logic is to be invoked. The non-IN FM receives and processes the non-IN event, and invokes the appropriate non-IN service logic instance. The non-IN FM executes the non-IN service logic instance, sending non-IN control requests to the FIM/CM as necessary to realize the service feature (the handling of subsequent information flows for such a non-IN service logic instance, if any, are not addressed in this example).

6) The IN-SM receives and processes the IN event indication. Given that a new instance of an IN service logic instance is to be invoked, the IN-SM creates a new instance of an IN-SSM to represent the state of the user's call and connection in a manner accessible to service Logic processing Programs (LPs) in the SCF (e.g. in connection points). It then sends an SSF information flow (via the SCF access manager) to the SCF providing a view of the current state of the IN-SSM.

7) The SCF receives and processes the SSF information flow. Given that a new instance of IN service logic is to be invoked, the SCF invokes an SLP instance (SLPI) that realizes the desired service feature. The SLPI is provided a view of the current state of the IN-SSM, and issues an SCF information flow to the SSF to request the IN-FM to manipulate the state of the IN-SSM, and/or to perform the call associated out-channel interaction as appropriate to realize the service feature. The SCF information flow may also indicate the set of events that should be reported to the SLPI [i.e. it indicates the set of BCSM and IN-SSM EDPs or/and the out-channel interaction related events (e.g. a reception of the FACILITY IE from the user in the BCSM to be armed for this particular service logic instance)].

8) The IN-SM receives the SCF information flow (via the SCF access manager) and processes it to manipulate the state of the IN-SSM, and/or to perform the call associated out-channel interaction as requested. In doing so, it generates an IN control request to the FIM/CM. It also monitors the IN-SSM for the IN-SSM events indicated in the request (if any).

9) The FIM/CM receives and processes the IN control request, and determines if it is valid based on other active service logic instances. It then sends a BCM control request to the BCM to notify it of the functions to be performed and of any BCSM events or/and the out channel interaction events to monitor for.

10) The BCM receives and processes the BCM control request, and manipulates one or more BCSMs to satisfy the request. In manipulating the BCSMs, it performs the appropriate bearer control and resource control functions or/and the out-channel interaction. The BCM also monitors the BCSMs for the BCSM events or/and the out-channel interaction events indicated in the BCM control request (if any).

11) If the BCM detects a BCSM event in a BCSM, it repeats step 3 to send a BCSM event indication to the FIM/CM.

If the out-channel interactions related event is detected by the BCM, the BCM sends the event indication to the FIM/CM.

12) The FIM/CM repeats step 4 to determine how to process the event. In this case, the event is for an active IN service logic instance. It sends an IN event indication to the IN-SM, indicating that the event is for an existing instance of an IN service logic.

13) The IN-SM receives and processes the IN event indication as in step 6, with the following differences. Given that the event is for an existing instance of an IN service logic, as represented by an existing IN-SSM instance. It updates the state of the existing IN-SSM to reflect the state of the user's connection(s) and/or request via the out-channel interaction, and reports the event and current IN-SSM state to the SCF in an SSF information flow. No new IN-SSM instance is created.

14) The SCF receives and processes the SSF information flow as in step 7, with the following difference. Given that the event is for an existing instance of an IN service logic, as supported by an existing SLPI. It does not invoke a new instance of an SLP. The SLPI then repeats its actions in step 7 to send an SCF information flow to the SSF to request the IN-SM to manipulate the state of the IN-SSM or to perform the out-channel interaction, and to indicate the next set of EDPs or/and the out-channel interaction related events of interest, if any.

15) Steps 8 to 14 are repeated until the IN service logic instance is ended. The IN service logic instance ends when the SLPI is no longer interested in any EDPs, or SSF/CCF processing has progressed beyond the point at which any EDPs can be encountered.

4.6 Relationship of SSF/CCF to SCF

This subclause only addresses call associated relationships as supported by a IN CS-2 connection control IN-SSM.

a) An SSF/CCF can have call associated relationships with multiple SCFs, and an SCF can have call associated relationships with multiple SSF/CCFs. Each relationship is treated as a one-to-one relationship.

b) When the SSF/CCF initiates a relationship, it reports the state of the IN-SSM in which the TDP was detected. The state information that is included in the information flows between the SSF/CCF and the SCF is defined by the information elements included in the information flows, based on the analysis of IN CS-2 SIBs and detailed DFP modelling.

c) Once a control relationship is established between the SSF/CCF and the SCF, the SCF can request the SSF/CCF to monitor for and report subsequent events (i.e. arm EDPs), as well as to stop monitoring (i.e. disarm EDPs).

Annex A addresses the nature of the relationship between the SSF/CCF and the SCF. It describes general terminology and possible control and monitor scenarios for both normal and abnormal situations.

5 Specialized Resource Function (SRF) model

5.1 General

A model of the SRF is shown in Figure 5-1. The purpose of this model is to provide a framework for specialized resource functionality subjects with respect to the SRF.

SRF provides various specialized resources as shown in 4.3.4. The SRF is managed to place resources in- or out-of-service, e.g. for provisioning, administration and maintenance purpose. The SRF management is activated by a request from another functional entity, and never takes action by itself.

For call/service processing, the SRF has a logical relationship with the SSF/CCF and the SCF. The SCF controls the connection between the SSF/CCF and the SRF, and sends instructions to the SRF.

As part of the process of formulating a response to the SSF, the SCF may need to enter into a dialogue with a calling or a called party. This could, for example, take the form of a prompt and collect digits sequence.

The SCF in IN CS-2 will instruct the SRF to start a dialogue with a user after setting up a path between the SSF/CCF and the SRF. The dialogue between the SRF and the user allows the SRF to play an announcement and if appropriate, collect digits. If digits have been collected, the SRF will pass the digit information to the SCF.

A user, being prompted from the SRF, inputs MF tones, for example, to the SRF so that the collected digits can be reported to the SCF. When the service logic in the SCF does not need the resources anymore, the SCF requests the SSF/CCF to release the connection with the SRF and the resource in the SRF will be released.

5.2 SRF Components

This subclause describes the SRF components.

Main CS-2 SRF components are the same for the CS-1 SRF; they are:

1) Functional Entity Access Manager (FEAM);
2) SRF Resource Manager (RM);
3) Resources.

The SRF Resource Manager is contained in a block called Resource Control Part along with the Resource Logic Library and the Resource Logic Instances; it is possible to split the Resources in the following blocks:

1) Resource Function Part (RFP);
2) Data Part (DP).

Enhancement to the CS-1 SRF components and new components for the CS-2 SRF are described as follows:

Figure 5-1 shows the CS-2 SRF model.

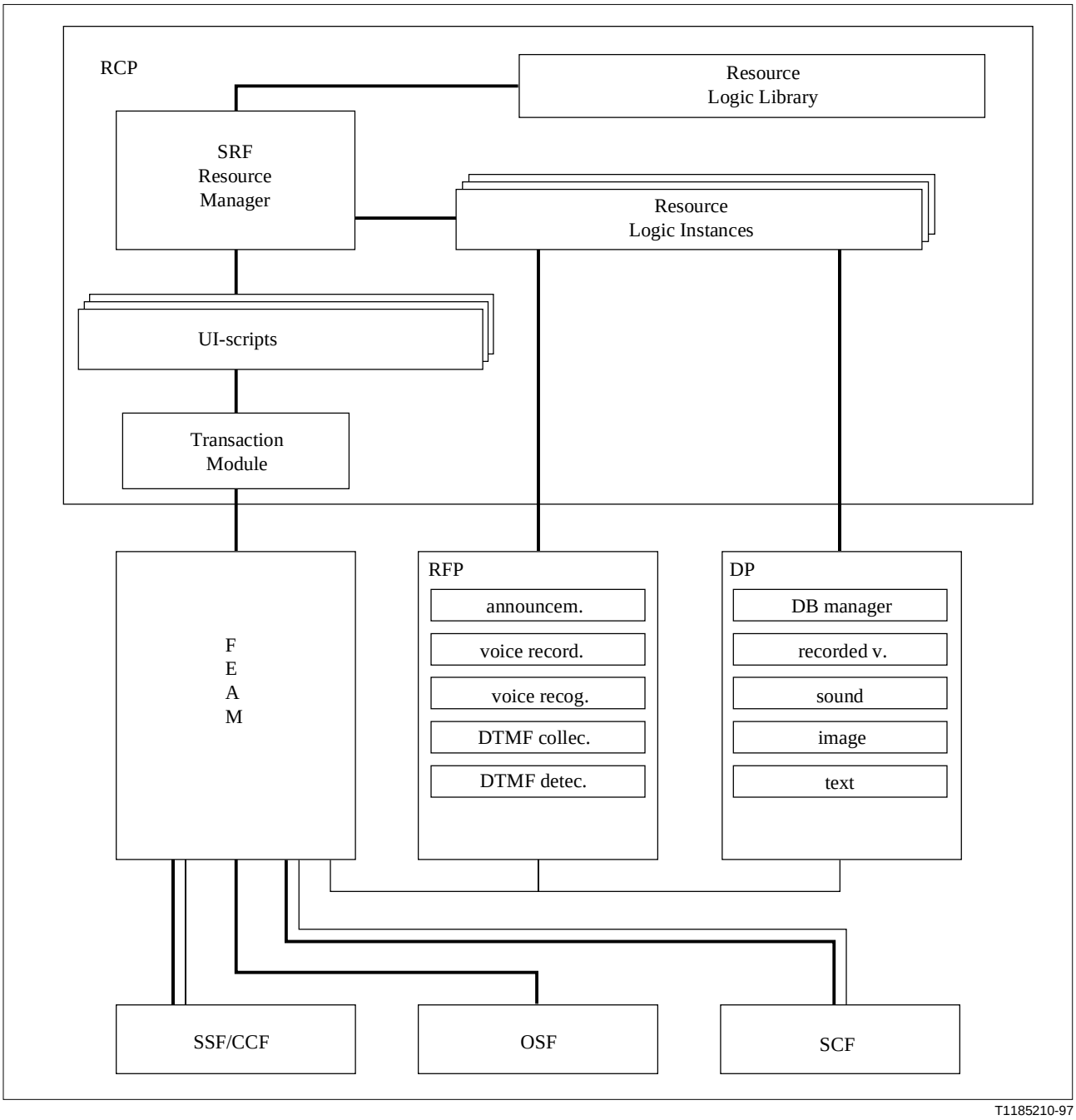


Figure 5-1/Q.1224 – SRF model

5.2.1 Functional Entity Access Manager (FEAM)

The FEAM provides the functionality necessary to exchange information with other functional entities:

– with SSF via the applicative protocol;

– with SCF via the direct protocol;

– with OSF via the direct protocol.

It also provides the functionality necessary to handle the voice connections.

5.2.2 Resource Control Part (RCP)

The RCP contains SRF service logic, and controls the service procedure using the capabilities of other blocks. To offer a specialized resource, RCP uses resource-function pair in RFP and data in DP.

Whenever RCP receives a call requesting specialized resource-function pair, it invokes the internal resource controller, which manages the first needed resource function pair to make a decision of admission or rejection of the call. The SRF sends ACK or NACK messages according to the decision by the resource controller.

There are as many controllers as special resource function types. The controllers accept or reject calls requesting a resource-function pair on the basis of characteristic parameters. A controller consists of an interface unit and a decision unit. The first one encodes and decodes the messages from/to RM and makes the input patterns for the decision unit; the characteristic parameter permitting the acceptation or rejection of the calls controlled by an algorithm within the decision unit. Figure 5-2 shows the model of Resource Controller.

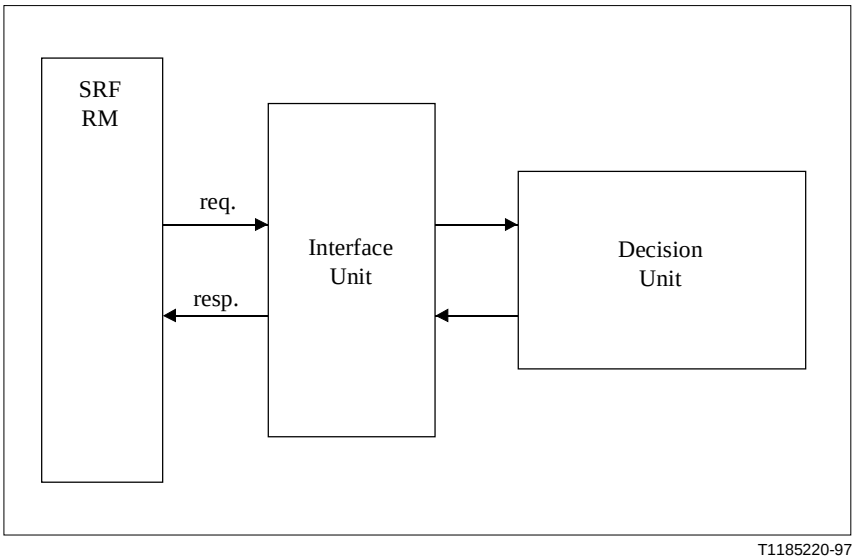


Figure 5-2/Q.1224 – Resource Controller model

5.2.2.1 SRF Resource Manager

The RM provides functionalities necessary for the SRF to manage the resources contained in it. The RM contains the capabilities to search for a resource, to allocate or de-allocate it, to manage the status of a resource and to control its actions.

5.2.2.2 Transaction module

It provides the functionality necessary for:

– detection of transaction from the communication links;

– routing of transactions to the right applications scripts.

5.2.2.3 User Interaction-scripts (UI-Script)

It provides to the SCF a vision of the different specialised resources that the SRF can perform. User Interaction-script is an aggregation of Resource Function.

5.2.2.4 Resource Logic Library

It indicates the SRF Logic and Physic Resources necessary for a given User Interaction-script.

5.2.2.5 Resource Logic Instances

It instances the SRF resources that are necessary for the correct execution of the invoked specialized resource.

5.2.3 Resource Function Part (RFP)

The RFP is a collection of resource-function pairs or functional element of resources. Resources in a resource-function pair for a service procedure are allocated and released together.

A preliminary set of CS-2 SRF resources are described in 4.3.4. It should be noted that other resources could be identified as new services/service features are considered for the CS-2. At the same time, the listed resources need to be prioritized as relevant services/service features will be for CS-2.

5.2.4 Data Part (DP)

The DP is composed of DataBase manager and database which contains recorded voice, sound, image, text, etc.

5.3 Objects of SRF management

Examples of specialized resources managed by the SRF are as follows:

– DTMF receiver.
– Tone generator.
– Announcements.
– Message sender/receiver.
– Synthesized voice/speech recognition devices with interactive prompting facilities.
– Text-to-speech synthesis.
– Protocol converters.
– Audio conference bridge.
– Information distribution bridge.

The following four objects are supported in IN CS-2 and defined as follows:

1) *DTMF receiver*

   This resource receives Dual Tone Multi-Frequency (DTMF) from a linked resource, and recognizes it as a standardized signal input.

2) *Tone generator/announcements*

   This resource provides in-channel information to the specified virtual resource.

3) *Message sender/receiver*

   This resource sends or receives messages, such as electronic messages, voice messages, etc., to/from users.

4) *Synthesized voice/speech recognition device with interactive prompting facilities*

This resource receives in-channel speech information from a linked virtual resource, and recognizes it as a standardized signal input. When the information is input from a user, it is recognized by this resource and this resource converts it to IN perceivable signals. When this resource receives an instruction to send a voice message with source-information, it is converted to voice message. Usually, such action is performed with interactive prompting.

The following four objects are not defined in IN CS-2:

5) *Audio conference bridge*

Receiving in-channel audio information from any other linked virtual resources, this resource mixes this information and sends the mixed information to all the linked virtual resources. Another new virtual resource can be joined to or any virtual resources linked to it can be split from this connection resource. It is used as an audio conference bridge.

6) *Information distribution bridge*

Receiving in-channel information from a linked virtual resource, this resource distributes the information to all the other linked virtual resources. Another new virtual resource can be joined to or any virtual resources which receive distributed information can be split from the connection resource. It is used as a broadcasting device.

7) *Text-to-speech synthesis*

8) *Protocol converters*

6 Service Control Function (SCF) model

6.1 General

A model of the SCF is shown in Figure 6-1. The purpose of this model is to provide a framework for service logic processing subjects with respects to the SCF.

The prime function of the Service Control Function (SCF) is execution of service logic provided in the form of Service Logic processing Programs (SLPs) and, accordingly, it includes the SLP execution supporting functions, such as service logic selection/interaction management, functional entity access management, SLP provisioning management, etc.

6.2 SCF components

6.2.1 General

To realize the above defined functionality, the SCF model is shown in Figure 6-1. It is noted that this shows a conceptual model of SCF and is not intended to imply an actual implementation of the SCF.

The SCF platform provides a Service Logic Execution Environment (SLEE) on which a Service Logic processing Program (SLP) runs to provide pertinent service processing. An SLP is a service application program invoked by the SLEE and is used to realize service processing under the control of the SLEE. The simultaneous invocation and execution of multiple SLPs are also managed by the SLEE.

Each of the entities shown in Figure 6-1 will now be described in the successive subclauses.

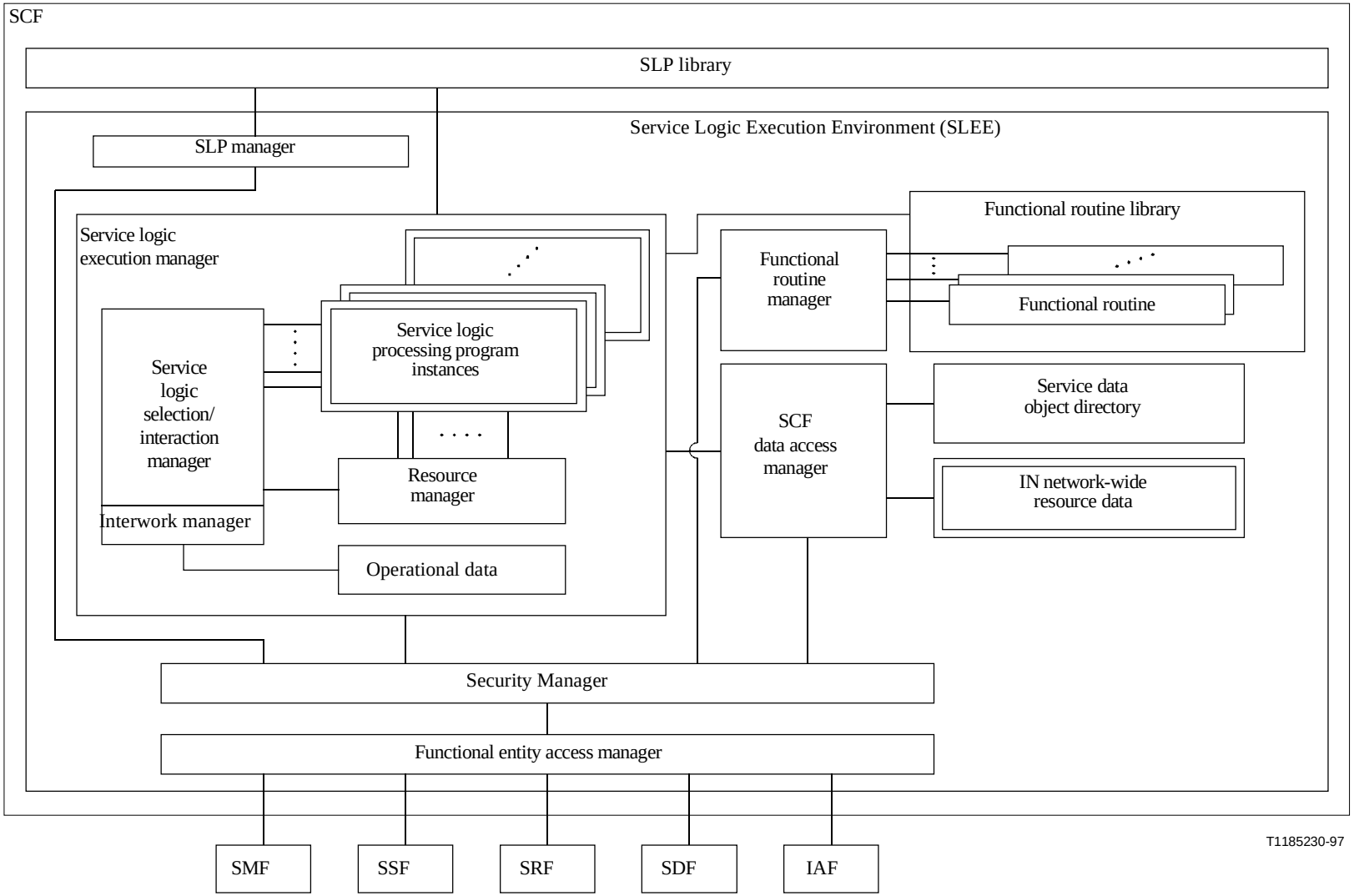


Figure 6-1/Q.1224 – SCF model

6.2.2 Service logic execution manager (SLEM)

6.2.2.1 General

The SLEM is the functionality that handles and controls the total service logic execution action. The SLEM contains Service Logic processing Program Instances (SLPIs), service logic selection/interaction manager, and resource manager. It also interacts with SCF data access manager and functional entity access manager to support SLPI execution. In addition to these aspects, the SLEM needs functionality to:

– execute SLPIs and maintain transient data associated with SLPIs (i.e. information that only persists during the lifetime of the SLPI, such as SLPI state information);

– execute functional routines in support of SLPI execution;

– manage SLPI access to SCF and SDF data via the SCF data access manager (see 6.2.3);

– manage the exchange of information between SLPIs and entities in other functional entities via the functional entity access manager (see 6.2.5).

6.2.2.2 Service Logic Selection/Interaction Manager (SLSIM)

The SLSIM is the entity that selects an SLP for execution and controls the simultaneous execution and/or execution order of multiple SLPs in the same SCF. IN CS-2 does not define whether or not the SLSIM is explicitly divided into two different entities, i.e. the service logic selection manager and the service logic interaction manager.

As part of the functionality, the SLSIM provides a means to manage service interactions by managing interactions among multiple SLPIs in the same SCF that are simultaneously active on a single call. The relationship between SLSIM and the feature interaction manager/call manager in the SSF/CCF is not defined in IN CS-2.

SLP selection is performed via the SLSIM in response to:

– an external event from another functional entity;

– the occurrence of internally recognized conditions (e.g. time of day or other internal events); and

– the execution of a functional routine via an SLPI that requests the execution of another SLP.

In addition, the SLSIM should invoke the execution of the selected SLP and provide for mutual exclusion and precedence during this SLP selection and invocation:

– mutual exclusion prevents the invocation of an SLP whose execution would be incompatible with a currently executing SLPI;

– precedence provides a scheme to select a particular SLP from a set of SLPs which meet the same selection criteria.

6.2.2.3 Service Logic processing Program Instance (SLPI)

A Service Logic processing program (SLP) is a service application program invoked by the SLEE and used to realize service processing. It contains logical constructs which, when executed, control the flow of service execution, and statements that, when executed, invoke functional routines in the SCF to access network resources and data needed for service execution. When an SLP is selected and invoked, it is referred to as a service logic processing program instance (SLPI). In contrast to an SLP, a corresponding SLPI is a dynamic entity that actively controls the flow of service execution and invokes SCF functional routines.

Functional routines are the functionality in the SCF that can be invoked by SLPIs to cause a sequence of functional entity actions to be performed in the network in support of service execution. This sequence of functional entity actions provides the functionality defined for a service independent building block (SIB) on the global functional plane. Therefore, functional routines are considered to be service independent. Potential categories of functional routines are described in 6.3.

6.2.2.4 Resource manager

The resource manager provides the functionality to control the allocation of local SCF resources and provides access to network resources in support of SLPI execution. The resource manager contains functionality to:

– identify and locate local SCF resources;

– identify and locate network resources via the SCF data access manager and IN network-wide resource data (see 6.2.3.3);

– identify one or more local SCF resources requested by a particular SLPI;

– release one or more local SCF resources no longer needed by a particular SLPI; and

– interact with other functional entities via the functional entity access manager to provide for the reservation and release of network resources to be used by SLPIs.

It is noted that SRF selection is not always performed by the SLEM resource manager, in some cases, selection is performed by an SSF, for example, upon assist/hand-off procedure being used.

6.2.2.5 Internetworking manager

Internetworking manager provides the functionality needed for the internetworking:

• access control based on the originating entity and the associated service/feature;

• secure access to the service logic and monitoring of the illegal access;

• accounting of the interworking activity (if needed).

6.2.3 SCF data access manager

6.2.3.1 General

The SCF data access manager provides the functionality needed to provide for the storage, management, and access of shared and persistent information in the SCF (i.e. information that persists beyond the lifetime of a SLPI). The SCF data access manager also provides the functionality needed to access remote information in SDFs. The SCF data access manager interacts with the SLEM to provide these functionalities to SLPIs.

Figure 6-1 identifies two structures that contain SCF data. These include:

– the service data object directory; and

– the IN network-wide resource data.

These are described in the following subclauses.

6.2.3.2 Service data object directory

Figure 6-1 identifies a service data object directory. It provides a means to address the appropriate SCF for access to a specific data object.

The SLEM interacts with the SCF data access manager to access service data objects in SDFs. The SCF data access manager uses the service data object directory to locate service data objects in the network in a manner transparent to the SLEM (and its SLPI). As such, the SLEM (and its SLPIs) has a global and uniform view of service data objects in the network.

6.2.3.3 IN network-wide resource data

This is a structure in which information resides about the location and capabilities of resources in the network accessible to SLPIs. It provides a means to address the appropriate functional entity (e.g. SRF) for access to specific resources with the appropriate capabilities.

The SLEM resource manager interacts with the SCF data access manager to access network resource data. The SLEM resource manager provides SLPIs with access to network resources in a manner transparent to SLPIs. As such, SLPIs have a global and uniform view of resources in the network.

6.2.4 Functional routine manager

Functional routine manager will be used for reception and distribution of functional routines to functional routine library via functional entity access manager. This entity also manages the addition, deletion and suspension of a particular functional routine. Such management of functional routines by the SMF is not defined in IN CS-2.

Functional routine library is an entity where the actual functional routines are residing.

6.2.5 Functional Entity Access Manager (FEAM)

The functional entity access manager provides the functionality needed by the SLEM to exchange information with other functional entities via messages. This message handling functionality should:

– be transparent to SLPIs;

– provide reliable message transfer;

– ensure sequential message delivery;

– allow message request/response pairs to be correlated;

– allow multiple messages to be associated with each other; and

– comply with OSI structures and principles.

6.2.6 SLP manager

The SLP manager manages the reception and distribution function of SLPs from other entities. The SLP manager, therefore, interworks with Functional Entity Access Manager (FEAM). This entity also manages addition, deletion and suspension of a particular SLP. The management of the SLP is via the SMF.

6.2.7 Security manager

The security manager provides the functionality needed for providing secure access to the functions of the SCF by other Functional Entities (SCF, SDF). This functionality should:

– generate security parameters for outgoing messages to provide data confidentiality, data origin authentication, data integrity for a specific interface;

– verify security parameters for incoming messages to provide data confidentiality, data origin authentication, data integrity for a specific interface. If an incoming message has an invalid security parameter then the message should be rejected and the invalid access attempt logged.

The level of security applied to an interface is determined by which of the security parameter features are active. The selection of active features is performed through interaction with the SMF.

6.3 Functional routine categories

The following categories of functional routines are proposed as framework for describing the SCF functionality accessible to SLPIs:

6.3.1 SLPI management functional routines

– functional routines to facilitate SLPI initialization and termination;

– functional routines to invoke other SLPs.

6.3.2 SLPI communication functional routines

– functional routines to support communication between SLPIs.

6.3.3 Timer management functional routines

– functional routines to retrieve the current time and date;

– functional routines to manage asynchronous timers in the SCF;

– functional routines to block the invocation of an SLP for a certain defined period.

6.3.4 Data management interface functional routines

– functional routines to access and manipulate SCF data (i.e. service data object directory and IN network-wide resource data) and network data (i.e. in an SDF) globally and uniformly via the SCF data access manager.

6.3.5 Asynchronous event handling functional routines

– functional routines to perform appropriate functions in response to asynchronous events (e.g. events reported by other functional entities, SLPI execution error events, and internal SCF events);

– functional routines to facilitate termination of a service execution and initialization of related resources.

6.3.6 Connection management functional routines

– functional routines to manipulate legs and connection points via interaction with the IN-feature manager in the SSF.

6.3.7 Specialized resource management functional routines

– functional routines to access and use specialized network resources globally and uniformly via the SLEM resource manager (interacting with the SRF).

6.3.8 OAM functional routines

– functional routines to respond to request for OAM activities and gather OAM-related information (e.g. data collection, traffic management, error handling, charging) through interactions with the SMF.

7 Service Data Function (SDF) model

7.1 General

A model of the SDF is shown in Figure 7-1. The purpose of this model is to provide a framework for service data functionality subjects with respect to the SDF.

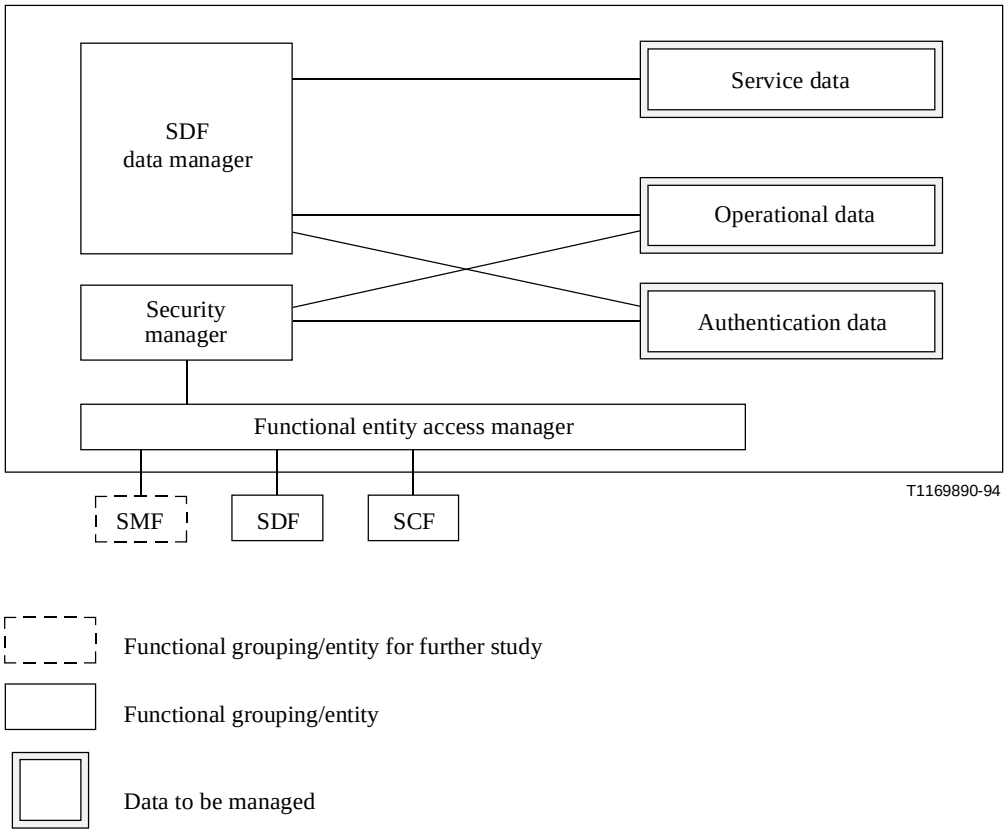


Figure 7-1/Q.1224 – SDF model

Subclause 7.2 describes the detailed SDF architecture, and subclause 7.3 clarifies and classifies data types which are handled by the SDF.

The SDF contains and manages the data which are related to Service Logic processing Programs (SLPs) and accessed in the execution of the SLP Instances (SLPIs). Therefore, data such as SLP selection data and SCF directory, which are accessed before the execution of an SLPI, are not included in the SDF handling data.

7.2 SDF components

7.2.1 General

To realize the above defined functionality, the recommended SDF model is shown in Figure 7-1. Each of the functional entities shown in Figure 7-1 will be described in the successive subclauses. This is not meant to imply any specific implementation.

7.2.2 SDF data manager

The SDF data manager provides the functionality needed for storing, managing and accessing information in the SDF. If, for example, the data are physically structured as a database, the SDF data manager may also handle database accessing language such as an SQL.

7.2.3 Functional entity access manager

The functional entity access manager provides the functionality needed by the SDF data manager to exchange information with other functional entities, i.e. SCF, SDF and SMF, via messages. This message handling functionality should:

– provide reliable message transfer;

– ensure sequential message delivery;
– allow message request/response pairs to be correlated;
– allow multiple messages to be associated with each other; and
– comply with OSI structures and principles.

Here, the functional entity access manager may access other SDFs, because the data distribution in the network can completely be transparent to the SCF.

7.2.4 Security manager

The Security manager provides secure access to the different types of data held in the SDF, for example, denied access to the data for unauthenticated users. This functionality should:

– check the access rights of the SCF;
– authenticate users with provided information;
– count the failed authentication attempts for a given user (whether the implementation of this function can be realised in the SDF has not been investigated in IN CS-2);
– block data access;
– assign user's access rights;
– memorise user's access rights during his request;
– control user's right to access specific data.

The security manager also provides the functionality needed for providing secure access to the functions of the SDF by other Functional Entities (SCF, SDF). This functionality should:

– generate security parameters for outgoing messages to provide data confidentiality, data origin authentication, data integrity for a specific interface;
– verify security parameters for incoming messages to provide data confidentiality, data origin authentication, data integrity for a specific interface. If an incoming message has an invalid security parameter then the message should be rejected and the invalid access attempt logged.

The level of security applied to an interface is determined by which of the security parameter features are active. The selection of active features is performed through interaction with the SMF.

7.3 Data types handled by SDF

The data which are handled by the SDF can be classified into the following types:

1) Authentication data – These data are used to authenticate a user that accesses the database through a SCF, e.g. a PIN code, the value of a counter for failed authentication. The set of authentication data used is associated to a level of access rights.
2) Operational data – These data are not needed by the SLPIs, but are used by the SDF itself for operational and administrative purposes, e.g. references to an object class, access control data.
3) Service data – These data are used for the provision of a service, e.g. a subscriber profile, service provider agreements. These data can be used by several services if necessary.

8 Call Unrelated Service Function (CUSF) model

8.1 General

A model for the CUSF (Figure 8-1) shows the CUSF model for a single ended service logic instance related to a user. Figure 8-1 shows the framework for the call unrelated modelling subjects with respect to the CUSF.

The aspects of the CUSF model briefly described below include the Basic Non-Call Manager (BNCM), the IN-Non-Switching Manager (IN-NSM), the Feature Interactions Manager (FIM)/Non-Call Manager (NCM), the relationship of the BNCM and IN-NSM to the FIM/NCM. Additional detail is provided in subsequent subclauses.

a) **BNCM** – It provides an abstraction of an association and proceedings for call unrelated interactions between a user and a network. It detects basic call unrelated events that can read to the invocation of IN service logic instances or should be reported to active IN service logic instances, and manages resources required to support basic call unrelated control. The BNCM interacts with the FIM/NCM as described in the FIM/NCM description below.

b) **IN-NSM** – It interacts with the SCF via SCF Access Manager in the course of providing IN services/features to users. It provides the SCF an observable view of an CUSF call unrelated processing activities, and provides the SCF with access to CUSF capabilities, such as managing of an association and invoked operations. It also detects non-call processing events that should be reported to active IN service logic instances or can read to the invocation of IN service logic instances. It manages CUSF resources required to support IN service logic instances, if any.

c) **FIM/NCM** – It provides mechanisms to prevent invocation of multiple instances of IN and non-IN service logic instances on a single association. The ability of the FIM/NCM to arbitrate IN/non-IN call associated/call unrelated associated services/features and IN call unrelated associated services/features should be investigated. These two types of arbitration need an implicit relationship labelled B. The FIM/NCM integrates these interactions mechanisms with the BNCM and IN-NSM to provide the CUSF with a unified view of call unrelated processing internal to the CUSF for an association.

d) **BNCM relationship to IN-NSM –** The relationship that encompassed the interaction between the BNCM and the IN-NSM, through the FIM/NCM. The information flow related to this interaction is not externally visible and is not standardized for CS-2. However, an understanding of this subject is required to identify how basic call unrelated processing and IN call unrelated processing interact.

e) **BNCM and IN-NSM relationships to FIM/NCM –** The relationships that encompass the interaction between the BNCM and the FIM/NCM, and the IN-NSM and the FIM/NCM. The information flows related to these interactions are not externally visible and are not standardized for CS-2. However, an understanding of this subject is required in order to unify the BNCM, IN-NSM and FIM/NCM.

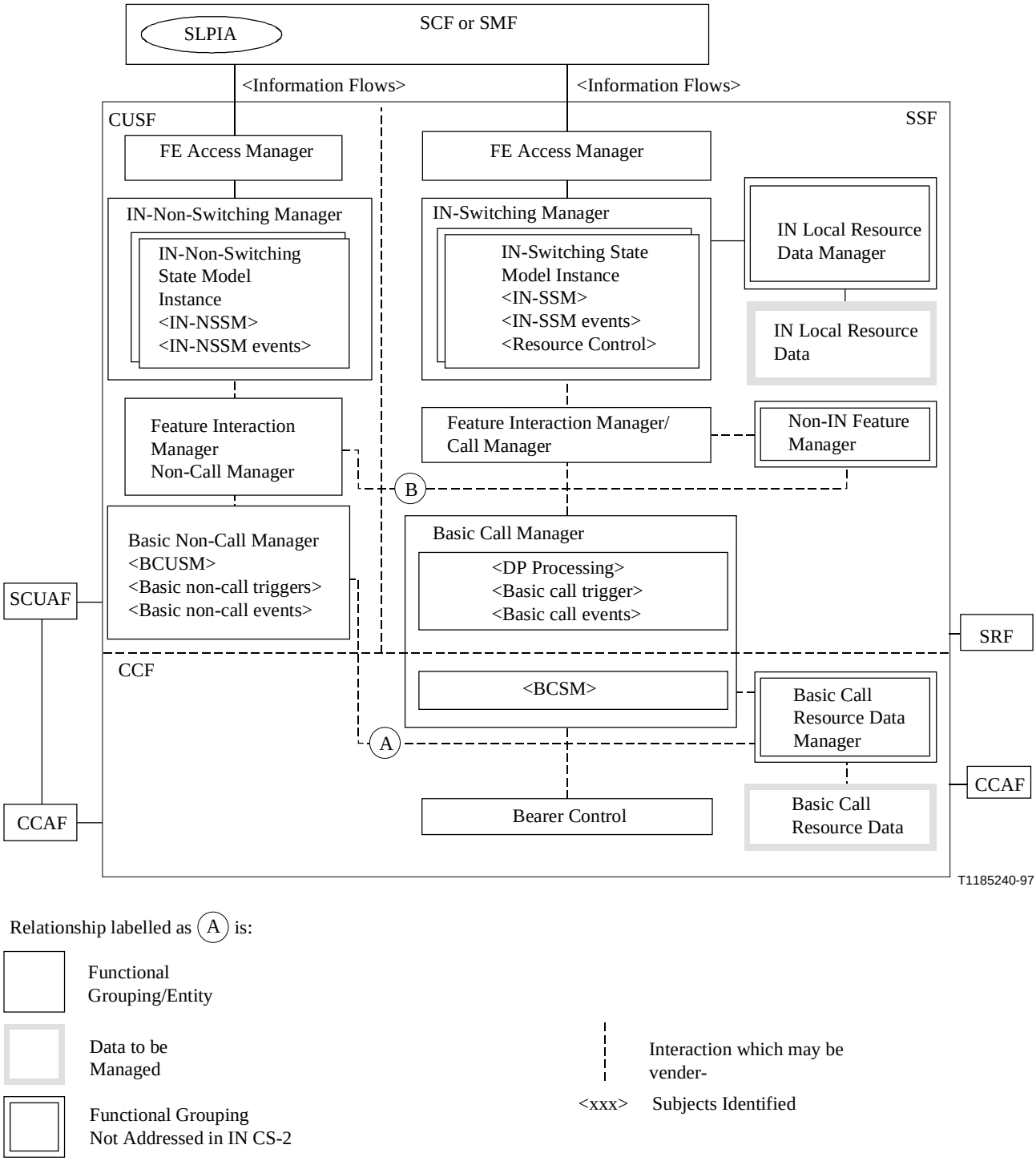


Figure 8-1/Q.1224 – The internal structure of CUSF and the relationship between FEs (not separated case)

8.2 Basic Non-Call Manager (BNCM)

Item a) in 8.1 provides a brief description of the BNCM. The BNCM subjects addressed below include the Basic Call Unrelated State Model (BCUSM), basic non call events that can lead to the invocation of IN service logic instances, and basic non-call events that should be reported to active IN service logic instances. A high-level description of these subjects is provided below.

8.2.1 BCUSM

This model relates to the call unrelated associated connection-oriented interaction between the user and the IN service logic using existing ISDN user-to-network signalling procedures.

Main concept of the BCUSM is similar to the BCSM. The BCUSM is a high-level description of the CUSF activities required to establish and maintain an association between users and service processing, to manage invoked operations. For this, it identifies a set of basic call unrelated associated activities in the CUSF and shows how these activities are joined together to process a basic call unrelated associated services/features (e.g. establish and maintain an association between a user and a network, manages it and operation invocations).

Many aspects of the BCUSM are not externally visible to IN service logic instances. However, aspects of the BCUSM that are reflected upward to the IN-NSM and FIM/NCM are visible to IN service logic instance. Only these aspects of the BCUSM will be subject of standardization. Therefore, the BCUSM is primarily an explanatory tool for providing a representation of CUSF activities that can be analysed to determine which aspects of the BCUSM will be visible to IN service logic instances, if any, and what level of abstraction and granularity is appropriate for this visibility.

The BCUSM identifies points in the basic call unrelated interaction when IN service logic instances are permitted to interact with basic call unrelated interaction processing (for the time being it is reasonable to assume ROSE procedures). In particular, it provides a framework for describing basic call unrelated events that can lead to the invocation of IN service logic instances or reporting of the events to the active logic, for describing those points in association and operation processing when the transfer of control can occur.

Figure 8-2 shows the components that have been identified to describe a BCUSM, Points in Association (PIA), Detection Points (DPs). Transitions and events are not depicted in the figure but they are self-explanatory. Figure 8-2 also shows the BCUSM. PIAs represent the states of an association for the call unrelated interaction. The concepts of DPs are same as the BCSM; however, an IN service logic instance can influence a call unrelated interaction processing with specifying a component of ROSE APDU (invoke, return result, error, reject) which should be delivered via appropriate signalling message between the user and the network (e.g. REGISTER, FACILITY, REL, and REL COMP on DSS 1).

8.2.2 BCUSM description for CS-2

In the following descriptions, the PIAs are related at a high level to the procedure taking place in an entity that supports ROSE. To help the understandings, Q.931 and Q.932 messages with information elements are attached to the description; however, these do not intend to limit the scope of BCUSM only to DSS 1.

The following is the BCUSM description for CS-2.

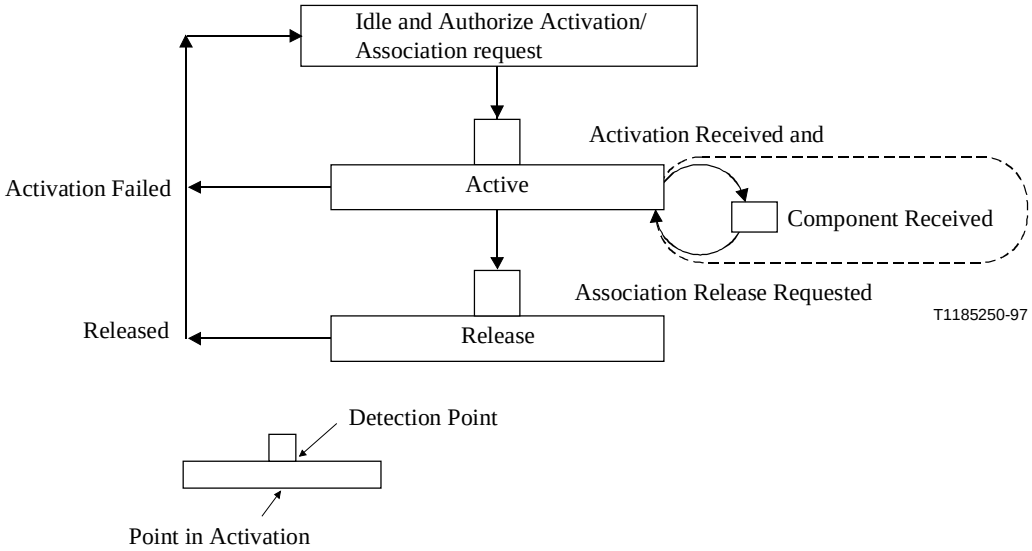


Figure 8-2/Q.1224 – BCUSM for CS-2

8.2.2.1 Idle and Authorize Activation/Association request

Entry Event

– All previously invoked operations between a user and a network are completed (event: Released), or the (last) invocation of an operation was rejected by the user or the network (event: Activation Failed). There is no relationship between the user and the network, e.g. no call reference exists between the user and the network.

Functions

– Logical connection is idled (e.g. no call reference, no data link connection).

– When the indication to establish an association and optionally invoke an operation is received (e.g. Q.932 REGISTER with FACILITY IE), the authority/ability of the user invoking the particular operation is verified. For example, some restrictions may be checked that are assigned to the used access, such as invoking User Registration is prohibited, if any.

Exit Event

– An indication to establish an association and optionally invoke an operation is received (e.g. Q.932 REGISTER with FACILITY IE), the authority/ability of the user invoking the particular operation was verified. (DP Activation Received and Authorized.)

8.2.2.2 Active

Entry Event

– An indication to establish an association and optionally invoke an operation is received (e.g. Q.932 REGISTER with FACILITY IE), the authority/ability of the user invoking particular operation was verified (DP Activation Received and Authorized), or a response/invocation from the user is received. (DP Component Received.)

Functions

– Logical connection is established (i.e. a call reference is assigned).

– The received operation is processed and an appropriate response is send to the user, or the response/invocation from the user is monitored.

Exit Event

– A response/invocation from the user is received. (DP Component Received.)

– Activation is failed. (Activation Failed.)

– An association release request is received from the user or the network (e.g. Q.931 REL). (DP Association Release Requested.)

8.2.2.3 Release

Entry Event

– An association release request is received from the user or the network (e.g. Q.931 REL), this request optionally contains response/invocation to the user or from the network. (DP Association Release Requested.)

Functions

– The association (logical connection) is being released with appropriate procedures (a pending response may be delivered with an association release message, e.g. Q.931 RELComp), this procedure may deliver final invocation/response to the user or to the network.

Exit Event

– The association is completely released (Released).

8.2.3 Transition for BCUSM

From	To
Activation Received and Authorized	Idle and Authorize Activation/Association request
Activation Received and Authorized	Active
Component Received	Idle and Authorize Activation/Association request
Association Release Requested	Idle and Authorize Activation/Association request
NOTE – These transitions are depicted here as they are beyond the normal BCUSM transitions.	

8.2.4 BCUSM DP criteria

As for CS-2 BCUSM, the DP criteria means to check the specific out-channel interaction activation at the Activation Received and Authorized, or Component Received DP against the specified pattern, for example an operation code for ROSE like interaction. The formal description is not defined for IN CS-2.

The following table denotes the applicability of DP criteria to DPs of BCUSM. The entries in the table can be (these are the same as the BCSM):

- Customer based;
- Trunk group based;
- Private facility based;
- Office based.

	DPs	
DP criteria	ActivationReceived AndAuthorized	ComponentReceived
Trigger Assigned	X	X
ITU-T ISDN supplementary services' operation codes (Note 1)	X	O
Regional/Network operator specific ISDN supplementary services' operation codes (Note 2)	X	O

X Applicable

O Optional

(the meaning of "X" and "O" is the same as for the BCSM)

NOTE 1 – The total number of the codes does not exceed the currently defined ISDN supplementary services, and it should be noted that only IN CS-2 relevant parts will be used (e.g. Call Forwarding Activation, etc.), and the exhaustive list of the codes is desirable to be defined in terms of the IN CS-2 benchmark services/service features point of view.

NOTE 2 – The total number of the codes is regional or network operator specific.

8.3 Description of relationship model

The model takes into account the following points. The model should:

A) be able to treat call unrelated triggers and call unrelated events, and to provide a view with the SCF for influencing the call unrelated interactions;

B) be able to model non-IN supported services/features, like the call diffraction registration or activation via the call unrelated interaction;

C) be able to manage invocation of services/features via the call unrelated interaction and necessary interactions between them and non-IN supported services/features, or call associated services/features, if any.

Items A) and B) result in the model with the CUSF (Non-Call Service Function, in Figure 8-1). Its internal structure is similar to the CCF and SSF model. The call unrelated triggers and call unrelated events are abstracted via the BCUSM (Basic Call Unrelated State Model), the FIM/NCM (Feature Interaction Manager/Non-Call Manager), and the IN-NSSM (IN-Non-Switching State Model Instance).

The model employs the new CUSF; however, the implicit relationships labelled A and B are necessary for DSS 1 under some implementations.

In this model, the non-IN supported services/features are executed via Non-IN Feature Manager in the SSF, even if the interaction is call unrelated. For example, the relationship A means the management of the Call Reference on an interface, and the relationship B means the Call Forwarding activation and de-activation because these will affect the basic call process in the CCF.

It is also possible to locate the Non-IN Feature Manager in the CUSF. At least, with these relationships, A and B, item C) is covered by the model (Figure 8-1).

There is still a freedom for each implementer which physical entity he/her maps the CUSF on. The separate allocation of the CUSF and its internal structure including the implicit relationships should be reconsidered when they become a standardization issue.

9 Service Management Function (SMF) model

9.1 General

A model of the SMF is shown in Figure 9-1. The purpose of this model is to provide a framework for service deployment functions, service provisioning functions, service operation control functions, and service monitoring functions. The FEs' interaction is via the Functional Entity Action Manager.

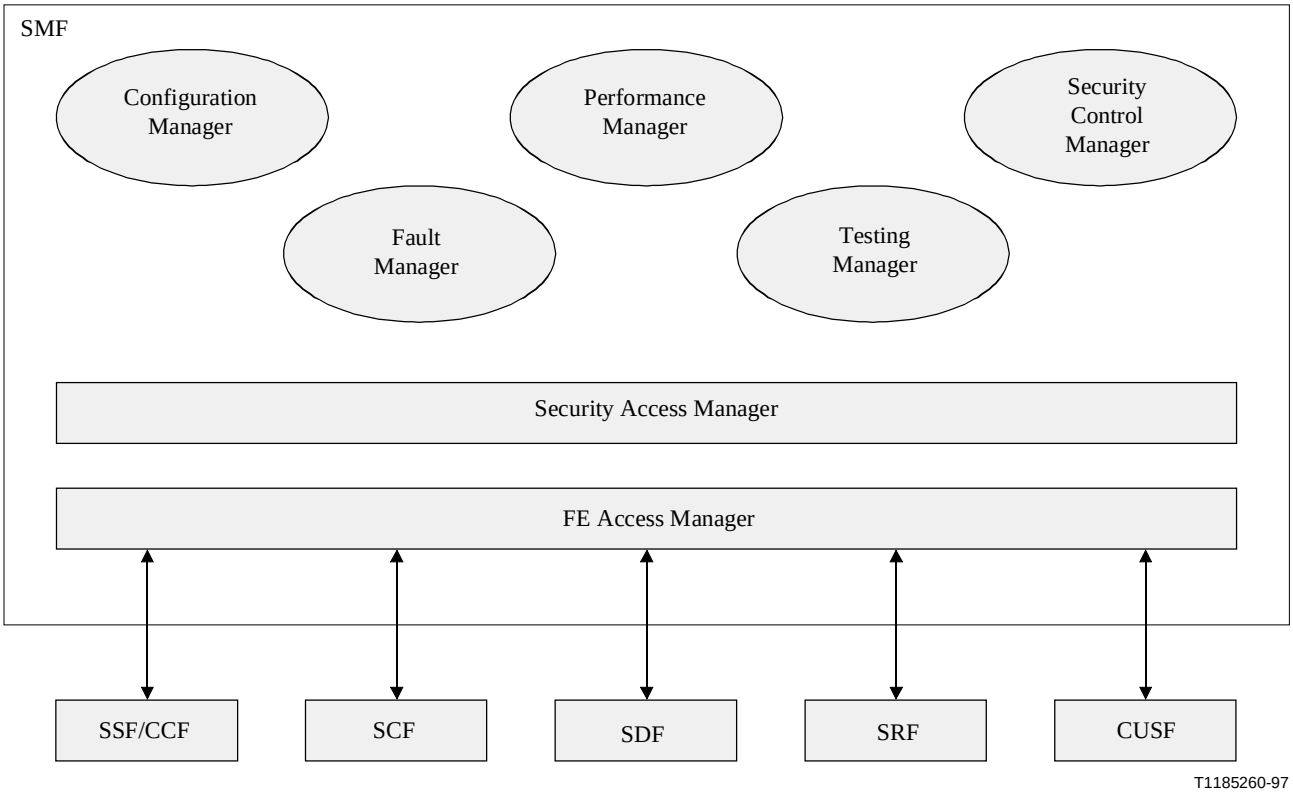


Figure 9-1/Q.1224 – SMF Model

9.2 SMF components

9.2.1 General

To realize the above defined functionality, the SMF model is shown in Figure 9-1. It is noted that this shows a conceptual model of the SMF and is not intended to imply an actual implementation of the SMF. The SMF platform provides a configuration manager, performance manager, fault manager, security control manager, and testing manager.

Configuration Manager: The Configuration Manager is responsible for changing configurations for service deployment and service operation control, or to correct faults.

Fault Manager: The Fault Manager analyses and correlates alarms and performs appropriate tests to determine root cause of fault conditions.

Performance Manager: The Performance Manager is responsible for gathering the appropriate performance information necessary to gauge the overall performance of service and network resources.

Testing Manager: The Testing Manager is responsible for the invocation and management of testing capabilities. The Testing Manager may be used by the Fault Manager in the fault resolution process.

Security Control Manager: The security control manager provides for logging of invalid access attempts and generation of alarms, activation and deactivation of appropriate security features in each FE to enable correct interface operation, and distribution of security related information.

Security Access Manager: The Security Manager provides the functionality needed for secure access to the functions of the IN FEs. This functionality applies security procedures to ensure data confidentiality, origin authentication and data integrity.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communication
Series Y	Global information infrastructure
Series Z	Programming languages