



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.1551

(06/97)

SERIES Q: SWITCHING AND SIGNALLING
Intelligent Network

Application of Intelligent Network Application Protocols (INAP) CS-1 for UPT Service Set 1

ITU-T Recommendation Q.1551

(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
Interworking with Standard-A INMARSAT system	Q.1100–Q.1109
Interworking with Standard-B INMARSAT system	Q.1110–Q.1149
Interworking with the INMARSAT aeronautical mobile-satellite system	Q.1150–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1999
BROADBAND ISDN	Q.2000–Q.2999

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

ITU-T RECOMMENDATION Q.1551

APPLICATION OF INTELLIGENT NETWORK APPLICATION PROTOCOLS (INAP) CS-1 FOR UPT SERVICE SET 1

Summary

This Recommendation describes the application of INAP CS-1 (Refined) for UPT Service Set 1. It explains how to use INAP as defined in CS-1 Refinements to provide the UPT service. Clauses 1 to 3 contain general information including the Chosen architecture. Clause 4 describes the application contexts used for UPT. Clause 5 describes both the UPT information model and the UPT security model as used in SDF. In clause 6 the UPT specific behaviour of the SCF is described for every UPT procedures, using both a textual and an SDL description. Clause 5 constitutes the core part of the UPT Recommendation. The other clauses are included in this Recommendation because clause 5 is not in itself self-explanatory. The definition of the data model specifies (with some flexibility) the internetwork interface and as such should be considered as the main part of this Recommendation.

Source

ITU-T Recommendation Q.1551 was prepared by ITU-T Study Group 11 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 5th of June 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/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
1 Scope	1
2 References and abbreviations	1
2.1 References	1
2.2 Definitions	2
2.3 Abbreviations	2
3 UPT phase 1 requirements	3
3.1 Architecture requirements	3
3.2 Requirements on the network	4
3.2.1 Requirements on the originating network side	5
3.2.2 Requirements on the terminating network side	5
3.2.3 Requirements on the fixed network	5
4 UPT application contexts	5
5 UPT information model	5
5.1 Introduction	5
5.2 UPT information base	6
5.2.1 Information base	6
5.2.2 Structure of the UPT information model	16
6 SCF procedures	19
6.1 General	19
6.1.1 Overview	19
6.1.2 Charging procedures in the SDLs	20
6.1.3 Conventions and notation	20
6.1.4 SLP description	24
6.2 Generic sequences	27
6.2.1 Identification and authentication	27
6.2.2 Feature request identification	36
6.2.3 Release of the calling user	39
6.2.4 Connection of an SRF	40
6.2.5 Disconnection of an SRF	43
6.3 Personal mobility	44
6.3.1 Registration for incoming calls	44
6.3.2 Deregistration for incoming calls	57

	Page
6.4 Call handling	63
6.4.1 Outgoing UPT call	64
6.4.2 Incoming UPT call	97
6.5 Service profile management procedure	120
6.5.1 Service profile interrogation procedure.....	120
6.5.2 Service profile modification procedure.....	127
Annex A.....	136

Introduction

This Recommendation describes the application of INAP CS-1 (Refined) [1] for UPT Service Set 1 [8]. Clauses 1 to 2 contain general information. Clause 3 describes the network architecture for UPT. In clause 4 the application contexts used for UPT are listed. Clause 5 describes both the UPT information model and the UPT security model as used in the SDF. In clause 6 the UPT specific behaviour of the SCF is described for every UPT procedures, using both a textual and an SDL description. Clause 5 constitutes the core part of the UPT Recommendation. The other clauses are included in this Recommendation because clause 5 is not in itself self-explanatory.

Recommendation Q.1551

APPLICATION OF INTELLIGENT NETWORK APPLICATION PROTOCOLS (INAP) CS-1 FOR UPT SERVICE SET 1

(Geneva, 1997)

1 Scope

This Recommendation specifies the application of Core INAP for the UPT service and describes the internetwork interface. It is applicable to UPT Service Set 1 as defined within Recommendation F.851 [8], but is limited to only those features which can be supported by INAP CS-1. This Recommendation includes procedures for handling the call forwarding supplementary services, these enhancements to the procedures are optional. In this Recommendation only the generic SSF-SCF INAP operations have been used, this does not preclude the use of the DP Specific operations. For the SSF-SRF case, this Recommendation does not use the assist handoff procedure.

The UPT service relies on the IN architecture as described in [4].

2 References and abbreviations

2.1 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore 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.

- [1] ITU-T Recommendation Q.1218 (1995), *Interface Recommendation for intelligent network CS-1*.
- [2] ITU-T Recommendation Q.76 (1995), *Service procedures for Universal Personal Telecommunication – Functional modelling and information flows*.
- [3] ITU-T Recommendation X.521 (1997), *Information technology – Open Systems Interconnection – The directory – Selected object classes*.
- [4] ITU-T Recommendation Q.1214 (1995), *Distributed functional plane for intelligent network CS-1*.
- [5] ITU-T Recommendation Q.1213 (1995), *Global functional plane for intelligent network CS-1*.
- [6] ITU-T Recommendation Q.1600 (1997), *Interaction between ISUP and INAP*.
- [7] ITU-T Recommendation Q.763 (1993), *Formats and codes of the ISDN User Part of Signalling System No. 7*.
- [8] ITU-T Recommendation F.851 (1995), *Universal Personal Telecommunication (UPT) – Service description (service set 1)*.

2.2 Definitions

No definitions have been identified.

2.3 Abbreviations

This Recommendation uses the following abbreviations

AC	Application Context
BCSM	Basic Call State Model
CFB	Call Forwarding on Busy
CFNR	Call Forwarding on No Reply
CFU	Call Forwarding Unconditional
CS-1	Capability Set 1
DEREG_IN	Deregistration for incoming calls
DTMF	Dual Tone Multi Frequency
FRI	Feature Request Identification
FSM	Finite State Model
IA	Identification and Authentication
IN	Intelligent Network
INAP	Intelligent Network Application Protocol
INCALL	Incoming Call
IP	Intelligent Peripheral
ISDN	Integrated Services Digital Network
ns	Sent part of the sequence number, i.e. the 16 least significant bits
OUTCALL	Outgoing Call
PIN	Personal Identification Number
PSTN	Public Switched Telephone Network
PUI	Personal User Identity
REG_IN	Registration for incoming calls
SCF	Service Control Function
SCSM	SCF State Model
SDF	Service Data Function
SDFh	Home Service Data Function
SDFo	Originating Service Data Function
SIB	Service Independent Building Blocks
SLP	Service Logic Program
SLPI	Service Logic Program Invocation
SPI	Service Profile Interrogation
SPM	Service Profile Modification
SRF	Specialized Resource Function
SSF	Service Switching Function
SSP	Service Switching Point

TCAP	Transaction Capabilities Application Part
UPT	Universal Personal Telecommunication
UPTAC	UPT Access Code
UPTAN	UPT Access Number

3 UPT phase 1 requirements

The UPT service set 1 is a set of UPT features that can be implemented without major changes to current technology, and is basically restricted to provision in PSTN and ISDN, with voice and telephony type services. This clause includes a number of operational requirements.

3.1 Architecture requirements

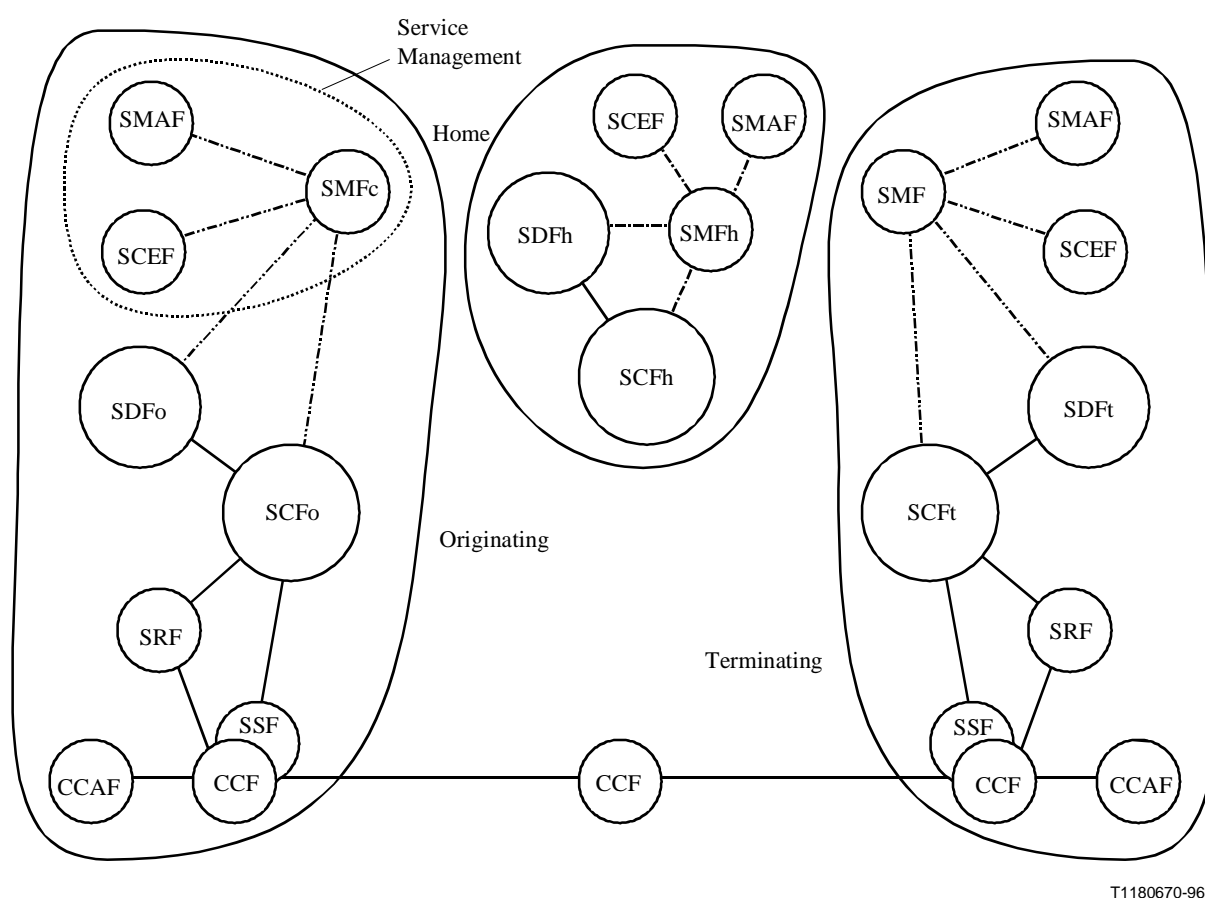


Figure 1/Q.1551 – General UPT functional architecture

Figure 1 gives an overview of the general UPT functional architecture. Apart from standard Intelligent Network (IN) terminology, the following notations are used in Figure 1:

- SCFh Home SCF;
- SDFh Home SDF;
- SMFh Home SMF;
- SCFo Local ("visited") SCF, originating side;
- SDFo Local ("visited") SDF, originating side;

- SMFo Local ("visited") SMF, originating side;
- SCFt Local ("visited") SCF, terminating side;
- SDFt Local ("visited") SDF, terminating side.

The functional architecture for UPT Service Set 1 is described in Figure 2. The differences with the general UPT functional architecture are :

- The interconnection of networks takes place between the SCFo and SDFh functional entities, as indicated by the arrow on the figure. The interface between SCF and SDF is specified in IN CS-1;
- SDFh stores all data related to the UPT user (i.e. the database in UPT Phase 1 is centralized);
- SDFh must as a consequence provide access control functions to check whether or not requests received from remote entities are authorized requests or not;
- SDFh performs the authentication of the UPT user;
- SDFo stores a list of agreements, which indicates the identity of all the service providers whose subscribers are allowed to access UPT service in SDFo's network;
- SDFo stores a list of service limitations resulting from agreements with service providers or network limitations;
- SDFo also stores information related to the management of the UPT service in its network, e.g. charging records which will be used later on for accounting.

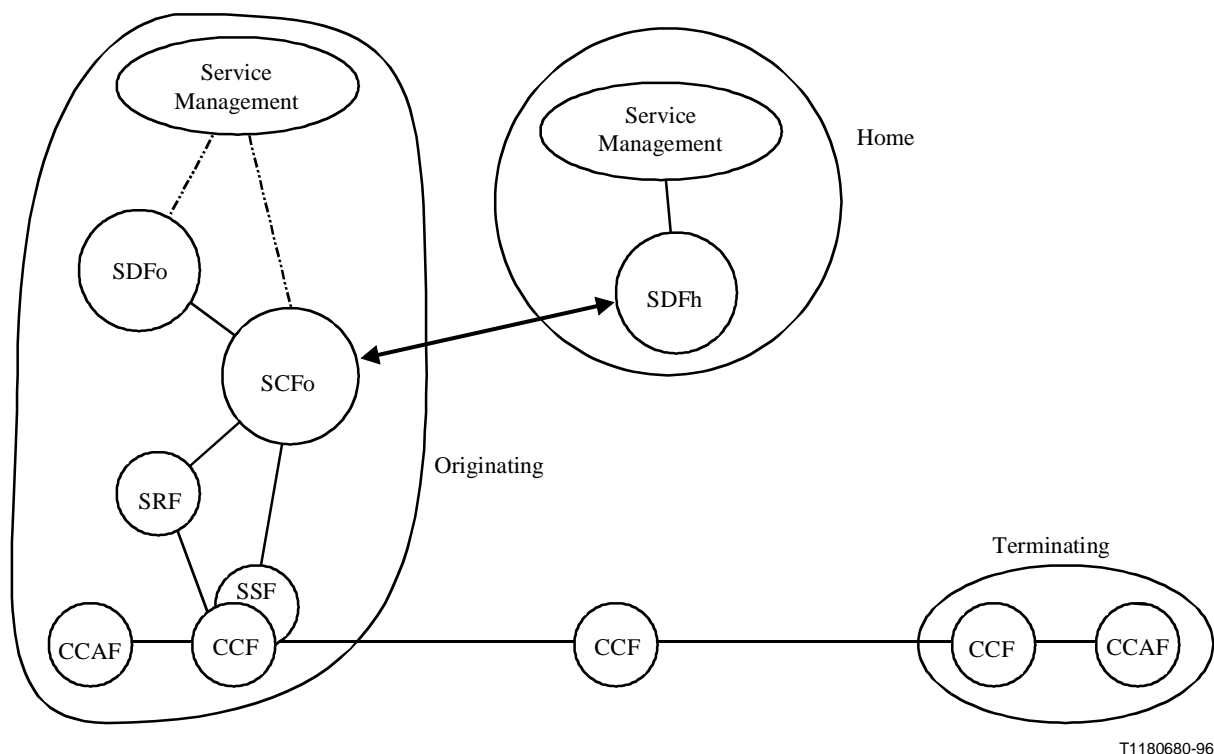


Figure 2/Q.1551 – Phase 1 UPT functional architecture

3.2 Requirements on the network

The UPTAC, UPTAN and UPT number must be recognized by the SSP.

3.2.1 Requirements on the originating network side

- Signalling systems used on the UNI will be those for the PSTN and ISDN. For interaction with the user, user-information is to be collected in-band by means of Dual Tone Multi Frequency (DTMF). This means that for ISDN the D-channel is not used for collecting user-information.
- In the PSTN, DTMF devices or terminals have to be used, because, for example, the non-fixed length numbers are ended with an #.
- ISDN terminals have to be provided with DTMF functionality, otherwise it would not be possible to carry in-band information for user interactions.

3.2.2 Requirements on the terminating network side

- Signalling systems used at the UNI will be those for the PSTN and ISDN.

3.2.3 Requirements on the fixed network

- Feature interactions with line services are outside the scope of this Recommendation.

4 UPT application contexts

The UPT Service Set 1 shall use the application contexts as defined in [1]. The following application contexts are used at the interfaces with the SCF:

SSP-SCP:

- itu-t recommendation q 1218 scf-ssf-objects(1) generic-ssf-to-scf(0) version1 (0);
- itu-t recommendation q 1218 scf-ssf-srf-objects(1) srf-to-scf(3) version1(0).

SCP-SDP:

- itu-t recommendation q 1218 sdf-objects(10) 3 1

5 UPT information model

5.1 Introduction

The UPT phase 1 needs a significant amount of data stored in the SDFs. Data for UPT are contained in the following data model¹. The aim of that model is two-fold: first to provide a list of all the data needed to support UPT Service Set 1 from the service and secondly to present the data as formally as possible so that they are ready to be used as parameters of the database operations.

Due to the amount of information contained in the data model, the model needs to be formally organized. The information and its associated structure make up the UPT Information Base (UPT-IB). The contents of the data model is described in 5.2.1 and its structure in 5.2.2. The generic information base on which the present UPT-IB is based is described in Recommendation Q.1218 [1], where the different classes are specified.

The organization of information provided with the UPT-IB does not imply any physical mapping of information even though some specific mappings will facilitate the use of the model.

¹ The alignment of this model with the management model of a customer (as defined in the Q.8xx-Series of Recommendations) is for further study.

Subclause 5.2 defines the objects, their attributes and the relations between them. This part of the data model is UPT-specific (but could probably be extended to other services). The complete ASN.1 module gathering the ASN.1 definitions of the subsequent subclauses is provided in Annex A.

5.2 UPT information base

5.2.1 Information base

The information model has been organized in object classes. Each object class is a general representation of an object of telecommunications (service, user, subscriber, etc). An object is an instance of the object class. Each object class is characterized by attributes. The attributes contain the data needed to fulfil the service.

Several object classes have been identified as well as their attributes. Figure 3 gives the inheritance relationships between the different object classes. All the object classes are subclasses of **top** which is an abstract class from which all the other classes are subclasses.

Apart from **top**, 14 types of object classes have been identified:

- alias;
- country;
- organization;
- organizational unit;
- administrative unit;
- UPT provider;
- partner;
- agreed service;
- UPT user profile;
- called UPT user;
- calling UPT user;
- UPT user profile alias;
- supplementary service;
- call forwarding.

They are described in more detail in the following subclauses. The classes **top**, **country**, **organization**, **organizational unit** and **alias** are part of Recommendation X.521 [3] and are not described further here.

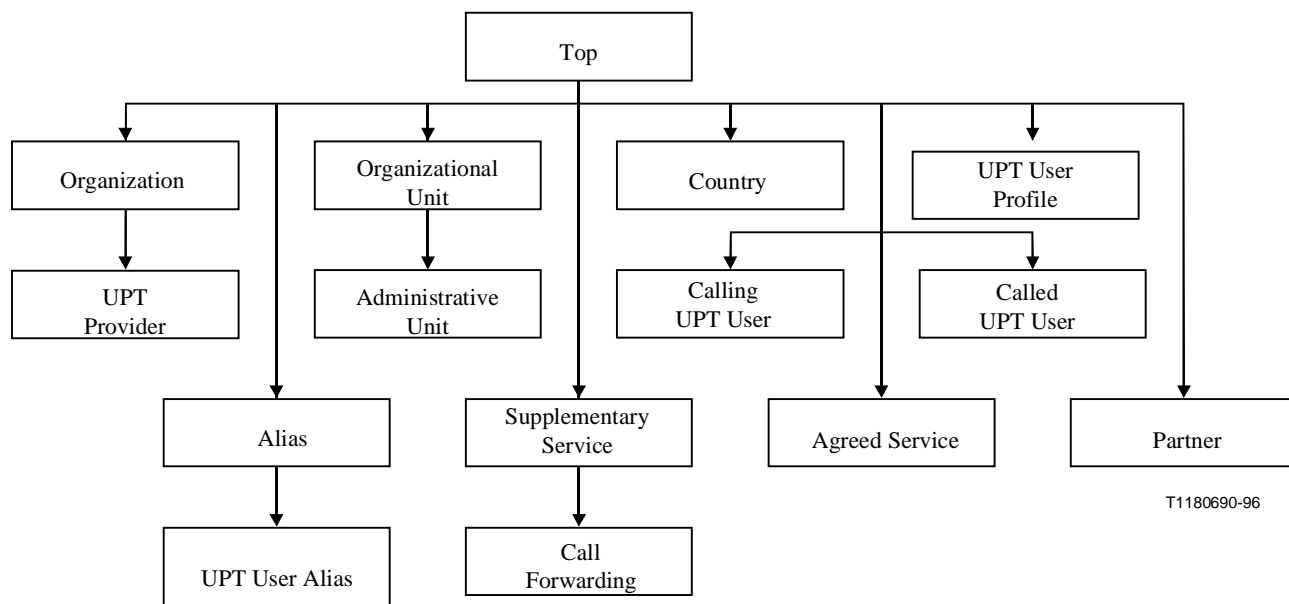


Figure 3/Q.1551 – Inheritance for the object classes

5.2.1.1 UPT provider

This object-class defines a UPT provider. It gives all the information concerning the provider that is necessary to support the UPT service. The definition involves:

- identifying the provider.

The following ASN.1 description shall be used to define the UPT provider object class:

```

uptProvider OBJECT-CLASS ::= {
  SUBCLASS OF          {organization}
  MUST CONTAIN        {providerId}
  MAY CONTAIN         {description}
  ID                  id-oc-uptProvider}

providerId ATTRIBUTE ::= {
  WITH SYNTAX          AddressString {ub-providerId}
  EQUALITY MATCHING RULE numericStringMatch
  SUBSTRINGS MATCHING RULE reversePrefixMatch
  SINGLE VALUE          TRUE
  ID                   id-at-providerId}
  
```

ub-providerId INTEGER ::= 3

AddressString {INTEGER: ub-max-value} ::= NumericString (SIZE (1..ub-max-value))

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **providerId** attribute identifies the UPT provider. This identifier is a numeric string. It may be part of a numbering plan. In case of the home service provider, it shall be possible to get the **providerId** value from a translation of the UPT number or the PUI.

5.2.1.2 Partner

This object-class defines a partner of a UPT provider, i.e. another UPT provider with whom a roaming agreement exists. It gives all the information concerning the partner that is necessary to support the UPT service. The definition involves:

- identifying the provider.

The following ASN.1 description shall be used to define the UPT partner object class:

```
partner OBJECT-CLASS ::= {  
    MUST CONTAIN {providerId}  
    MAY CONTAIN  {description}  
    ID           id-oc-partner}
```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **providerId** attribute identifies the UPT provider (see UPT provider object class).

5.2.1.3 Agreed Service

This object class gives the service, that is provided to home users by a visited provider, and its restrictions of use. The definition involves:

- identifying the service;
- giving the restrictions on the use of the service. The restrictions on a service under agreement might be different from those on a service proposed by a provider to its subscribers.

The following ASN.1 description shall be used to define the agreed service object classes:

```
agreedService OBJECT-CLASS ::= {  
    MUST CONTAIN {providedServiceId}  
    MAY CONTAIN  {providedLocations|  
                 description}  
    ID           id-oc-agreement}
```

```
providedServiceId ATTRIBUTE ::= {  
    WITH SYNTAX          Service  
    EQUALITY MATCHING RULE integerMatch  
    SINGLE VALUE  
    ID                  id-at-providedServiceId}
```

```
Service ::= INTEGER {  
    -- basic services 0-9  
    isdnTelephony (0),  
    -- registration service 10-19  
    icRegistration (10),  
    -- profile service 20-29  
    serviceProfileModification (20),  
    -- charging service 30-39  
    standard (30),  
    -- routing service 40-49  
    callForwardingUnconditional (40),  
    callForwardingOnNoReply (41),  
    callForwardingOnBusy (42),  
    variableRoutingOnTime (43),  
    variableRoutingOnCallingLine (44)}
```



```

providedLocations ATTRIBUTE ::= {
    WITH SYNTAX                      AddressString{ub-international-isdn-number}
    EQUALITY MATCHING RULE          numericStringMatch
    SUBSTRINGS MATCHING RULE      reversePrefixMatch
    ID                              id-at-providedLocations}

```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **providedServiceId** attribute gives the service identifier;
- the **providedLocations** attribute gives the service restrictions on the use of the service, in particular the destination of a service (geographic coverage zone for a unique provider). The restrictions are specified at the provider level. They give the limits on the provision of a service agreed between the provider and other providers for the roaming users. If any other restriction appears in the future, it should accompany the attribute presently described.

NOTE – The values contained in the attributes describing ISDN addresses or part of them shall be built upon international addresses without the international prefix.

5.2.1.4 Administrative unit

This object-class defines an administrative unit that gathers users into one administrative unit. It gives all the information concerning the unit that is necessary to support the UPT service. The definition involves:

- identifying the administrative unit.

The following ASN.1 description shall be used to define the administrative unit object class:

```

administrativeUnit OBJECT-CLASS ::= {
    SUBCLASS OF      {organizationalUnit}
    ID                id-oc-administrativeUnit}

```

5.2.1.5 User profile

This object class defines a user profile. The user profile gives the service information attached to one of the users in a subscription. This information may differ from one user to another in the same subscription. This definition involves:

- identifying the user;
- giving the list of the allowed services;
- giving service parameters for the allowed services.

The following ASN.1 description shall be used to define the user profile object class:

```

userProfile OBJECT-CLASS ::= {
    MUST CONTAIN  {pui}
    MAY CONTAIN   {description|
                    commonName|
                    surname|
                    seeAlso|
                    userCredit|
                    userPassword|      --as defined in Recommendation X.509
                    specialPassword|
                    variablePassword|
                    allowedServiceFeatures|
                    callInfoRecords|
                    activeChargingService|
                    nbOfFailedAuthentications}
    ID            id-oc-userProfile}

```

```

pui ATTRIBUTE ::= {
    WITH SYNTAX                               AddressString{ub-pui}
    EQUALITY MATCHING RULE                   numericStringMatch
    SINGLE VALUE                               TRUE
    ID                                           id-at-pui}

ub-pui INTEGER ::= 15

specialPassword ATTRIBUTE ::= {
    WITH SYNTAX                               OCTET STRING (SIZE (0..ub-special-password))
    EQUALITY MATCHING RULE                   octetStringMatch
    ID                                           id-at-specialPassword}

ub-special-password INTEGER ::= 8

variablePassword ATTRIBUTE ::= {
    WITH SYNTAX                               OCTET STRING (SIZE (0..ub-variable-password))
    EQUALITY MATCHING RULE                   octetStringMatch
    ID                                           id-at-variablePassword}

ub-variable-password INTEGER ::= 15

nbOffFailedAuthentications ATTRIBUTE ::= {
    WITH SYNTAX                               INTEGER (1..ub-max-nbOffFailedAuthentications)
    ORDERING MATCHING RULE                   integerOrderingMatch
    SINGLE VALUE                               TRUE
    ID                                           id-at-nbOffFailedAuthentications}

ub-max-nbOffFailedAuthentications INTEGER ::= 6

userCredit ATTRIBUTE ::= {
    WITH SYNTAX                               INTEGER (1..ub-maxUserCredit)
    ORDERING MATCHING RULE                   integerOrderingMatch
    SINGLE VALUE                               TRUE
    ID                                           id-at-userCredit}

ub-maxUserCredit INTEGER ::= 10000

callInfoRecords ATTRIBUTE ::= {
    WITH SYNTAX                               CallInfoRecord
    ID                                           id-at-callInfoRecords}

CallInfoRecord ::= SEQUENCE {
    authenticationTime                        [0] UTCTime,
    callStopTimeValue                        [1] UTCTime,
    callStartTimeValue                       [2] UTCTime,
    callingAddressValue                       [3] IsdnAddress,
    calledNumber                             [4] IsdnAddress,
    duration                                [5] INTEGER (0..2147483647) OPTIONAL,
    routingAddress                           [6] IsdnAddress OPTIONAL,
    forwardedToAddress                       [7] IsdnAddress OPTIONAL,
    invokedSupplementaryServices             [8] CFServices OPTIONAL,
    visitedNetwork                           [9] NetworkCode OPTIONAL,
    callCost                                [10] Cost OPTIONAL,
    surcharges                              [11] Cost OPTIONAL,
    releaseCause                            [12] Cause OPTIONAL}

Cost ::= CHOICE {
    pulse [0] INTEGER (1..ub-pulse),
    cost  [1] CurrencyValue}

NetworkCode ::= NumericString (SIZE (1..ub-networkCode))

```

```

ub-networkCode INTEGER ::= 3
ub-pulse INTEGER ::= 10000
CurrencyValue ::= CHOICE {
    usDollar          [0] Currency,
    frenchFranc       [1] Currency,
    germanMark        [2] Currency,
    dutchGuilder      [3] Currency,
    italianLira       [4] Currency,
    englishPound      [5] Currency,
    spanishPeseta     [6] Currency,
    swedishKrone      [7] Currency,
    norwegianKrone    [8] Currency,
    japaneseYen       [9] Currency}
Currency ::= REAL
CFServices ::= SET OF Service (40..49)
Cause ::= OCTET STRING (SIZE(minCauseLength..maxCauseLength))
minCauseLength INTEGER ::= 2
maxCauseLength INTEGER ::= 20

activeChargingService ATTRIBUTE ::= {
    WITH SYNTAX          Service (30..39)
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE         TRUE
    ID                   id-at-activeChargingService}

allowedServiceFeatures ATTRIBUTE ::= {
    WITH SYNTAX          Service
    EQUALITY MATCHING RULE integerMatch
    ID                   id-at-allowedServiceFeatures}

```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **pui** (personal user identifier) attribute is a number used to identify the UPT user. The PUI may be made up of a country code, a network code and a personal identifier [8]. It should be possible to retrieve the provider identifier of the user from the PUI;
- the **commonName** and **surname** attributes are used to name a user;
- the **seeAlso** attribute is a pointer to the object of **subscriber** object-class that describes the subscriber responsible for the user whose profile is described;
- several attributes are related to security. The **Password** attributes which could be of three types depending on the type of authentication available to the user give the password used by the user to authenticate. The **userProfile** object class should at least contain one type of **Password** attributes. The **nbOfFailedAuthentications** gives the remaining number of authentications that can be failed before the identifier is blocked. This number is limited by a value controlled by the subscriber or the provider;
- the **userCredit** gives the credit still available to the user. It also permits the update of the value of the **userCredit** attribute (however this calculation cannot be performed on-line by the SDF that is not able to do it);
- the **callInfoRecords** attribute contains all the call records related to a given user. It is used to keep track of the use of the service by a given user;
- the **allowedServiceFeatures** attribute gives the list of the services subscribed by the user and also the places where the services are available (locations are defined as contexts of the different service values).

5.2.1.6 User profile alias

This object class also describes the user profile. It is used to have another naming path for the user using in that case the UPT number. This definition involves:

- identifying the user;
- referring to the object (userProfile) that really contains the user profile information.

The following ASN.1 description shall be used to define the user profile alias object class:

```
userProfileAlias OBJECT-CLASS ::= {  
  SUBCLASS OF      {alias}  
  MUST CONTAIN     {uptNumber}  
  MAY CONTAIN      {description}  
  ID               id-oc-userProfileAlias}  
  
  uptNumber ATTRIBUTE ::= {  
    WITH SYNTAX      IsdnAddress  
    EQUALITY MATCHING RULE      numericStringMatch  
    SUBSTRINGS MATCHING RULE    reversePrefixMatch  
    SINGLE VALUE      TRUE  
    ID               id-at-uptNumber}
```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **uptNumber** attribute is the diallable number through which the user may be reached. The format of this attribute is an ISDN address (E.164).

5.2.1.7 Called UPT user

This object class defines the registered UPT user. The definition involves:

- identifying the service;
- giving the registration addresses for incoming;
- giving the charging information for split charging.

The following ASN.1 description shall be used to define the called UPT user object class:

```
calledUptUser OBJECT-CLASS ::= {  
  MUST CONTAIN      {name}  
  MAY CONTAIN      {description|  
                    icRegistrationAddress|  
                    allowedRegistrationAddress|  
                    defaultChargingReference}  
  ID               id-oc-calledUptUser}  
  
  defaultChargingReference ATTRIBUTE ::= {  
    WITH SYNTAX      IsdnAddress  
    EQUALITY MATCHING RULE      numericStringMatch  
    SUBSTRINGS MATCHING RULE    reversePrefixMatch  
    SINGLE VALUE      TRUE  
    ID               id-at-defaultChargingReference}
```

IsdnAddress ::= AddressString{ub-international-isdn-number}

```
icRegistrationAddress ATTRIBUTE ::= {  
  WITH SYNTAX      IsdnAddress  
  EQUALITY MATCHING RULE      numericStringMatch  
  SUBSTRINGS MATCHING RULE    reversePrefixMatch
```

ID

id-at-icRegistrationAddress}

```
allowedRegistrationAddress ATTRIBUTE ::= {  
  WITH SYNTAX                AddressString{ub-international-isdn-number}  
  EQUALITY MATCHING RULE      numericStringMatch  
  SUBSTRINGS MATCHING RULE    reversePrefixMatch  
  ID                          id-at-allowedRegistrationAddress}
```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **name** attribute gives the name or its identifier to which the registration addresses are attached. For internetworking purposes, this name has to be fixed;
- the **icRegistrationAddress** attribute gives the registration addresses for incoming calls and outgoing calls. It also contains the default registration addresses which are a value of the attribute with the default context. The other values have a time context that indicates the time validity of the values. Each registration address has a context that indicates the basic service to which the address is related;
- The **defaultChargingReference** attribute gives the default reference point for charging. The reference point may be specified with an **IsdnAddress** type;
- the **allowedRegistrationAddress** attribute contains complete or initial part of international ISDN addresses corresponding to areas where the user can be registered. The **icRegistrationAddress** attribute should take its values within the values of that attribute.

5.2.1.8 Calling UPT user

This object class defines the calling UPT user. The definition involves:

- identifying the object class;
- giving the rights for a calling UPT user.

The following ASN.1 description shall be used to define the calling UPT user object class:

```
callingUptUser OBJECT-CLASS ::= {  
  MUST CONTAIN      {name|  
                     allowedDestinations  
  MAY CONTAIN      {description}  
  ID                id-oc-callingUptUser}  
  
allowedDestinations ATTRIBUTE ::= {  
  WITH SYNTAX                AddressString{ub-locations}  
  EQUALITY MATCHING RULE      numericString  
  SUBSTRINGS MATCHING RULE    numericStringSubstringsMatch  
  ID                          id-at-allowedDestinations}
```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **name** attribute gives the name or its identifier for a calling UPT user. For internetworking purposes this name has to be fixed;
- the **allowedDestinations** attribute contains complete or initial part of international ISDN addresses corresponding to areas to which the user can set up a call.

5.2.1.9 Supplementary service

The **supplementaryService** object class contains the information related to the supplementary services in general. This object class is an abstract object class that is not used for the provision of

the UPT service. It allows the building of the **callForwardingService** object class (see the following subclauses) and other object classes for supplementary services.

The definition involves:

- identifying the object class;
- giving the activation status of the supplementary service.

The following ASN.1 description shall be used to define the supplementary service object class:

```

supplementaryService OBJECT-CLASS ::= {
    KIND                                abstract
    MUST CONTAIN                        {supplServId|
                                         supplServiceStatus}
    MAY CONTAIN {name|
                  description}
    ID                                id-ao-supplementaryService}

supplServId ATTRIBUTE ::= {
    WITH SYNTAX                        SS-Code
    EQUALITY MATCHING RULE            integerMatch
    SINGLE VALUE                       TRUE
    ID                                id-at-supplServId}

supplServiceStatus ATTRIBUTE ::= {
    WITH SYNTAX                        SupplServiceStatus
    SINGLE VALUE                       TRUE
    ID                                id-at-supplServiceStatus}

SupplServiceStatus ::= BIT STRING {
    provisioned (0),
    registered (1),
    activated (2)}

SS-Code ::= INTEGER {
    clip (11),
    clir (12),
    colp (13),
    colr (14),
    mci (15),
    cfu (16),
    cfb (17),
    cfna (18),
    cfnr (25),
    civr (26),
    tvr (27)}

```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **supplServiceId** attribute gives the name or the identifier of the supplementary service;
- the **supplServiceStatus** attribute indicates whether a service has been provisioned, registered or activated. This information may be modified by the user, using service profile modification procedures.

5.2.1.10 Call forwarding service

The object class defined in this subclause is the specialisation of the abstract object class defined in the previous subclause. The call forwarding services considered in UPT phase 1 are the following ones:

- call forwarding unconditional;
- call forwarding on no reply;
- call forwarding on busy;
- variable routing on time;
- variable routing on calling line.

The following ASN.1 description shall be used to define the call forwarding service object class:

```
callForwarding OBJECT-CLASS ::= {
  SUBCLASS OF {supplementaryService}
  MUST CONTAIN {forwardedToNumber|
                typesOfNotification}
  MAY CONTAIN {noReplyConditionTimer}
  ID          id-oc-callForwarding}
```

```
forwardedToNumber ATTRIBUTE ::= {
  WITH SYNTAX          IsdnAddress
  EQUALITY MATCHING RULE      numericStringMatch
  SUBSTRINGS MATCHING RULE   reversePrefixMatch
  ID                    id-at-forwardedToNumber}
```

```
typesOfNotification ATTRIBUTE ::= {
  WITH SYNTAX      TypesOfNotification
  SINGLE VALUE     TRUE
  ID              id-at-typesOfNotification}
```

```
TypesOfNotification ::= BIT STRING {
  servedUserForwardedCall (0),
  callingUserWithForwardedToNumber (1),
  callingUserWithoutForwardedToNumber (2),
  servedUserForwardingActivation (3)}
```

```
noReplyConditionTimer ATTRIBUTE ::= {
  WITH SYNTAX          INTEGER (1..ub-noReplyConditionTimer)
  EQUALITY MATCHING RULE      integerMatch
  SINGLE VALUE           TRUE
  ID                    id-at-noReplyConditionTimer}
```

```
ub-noReplyConditionTimer INTEGER ::= 10000
```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is as follows:

- the **forwardedToNumber** attribute indicates the address to which the call should be routed in case of call forwarding. This address is an E.164 address. For the variable routing service each address is associated with a context defining the dependence of the address on the time or on the calling line identity;
- the **typesOfNotifications** attribute gives the type of notifications that were requested from the user when registering the call forwarding service;
- the **noReplyConditionTimer** attribute gives the time after which a call is considered as not answered.

5.2.2 Structure of the UPT information model

5.2.2.1 Name forms

For each object class, the name forms define the attributes which will be involved in the naming of the object class. This attribute will identify instances of the object class. The naming attribute is chosen so that instances of the object class can be uniquely addressed. The naming attribute should be a mandatory attribute of the object class.

The following ASN.1 description may be used to define the naming attributes of all the object classes defined in the previous subclause. The name form gives the object class to be named and its naming attribute. For the object classes defined in the previous subclause, the following name forms shall be used:

```
uptProviderNameForm NAME-FORM ::= {  
    NAMES          uptProvider  
    WITH ATTRIBUTES {providerId}  
    ID              id-nf-uptProviderNameForm}  
  
partnerNameForm NAME-FORM ::= {  
    NAMES          partner  
    WITH ATTRIBUTES {providerId}  
    ID              id-nf-partnerNameForm}  
  
adminUnitNameForm NAME-FORM ::= {  
    NAMES          administrativeUnit  
    WITH ATTRIBUTES {organizationalUnitName}  
    ID              id-nf-adminUnitNameForm}  
  
agreedServiceNameForm NAME-FORM ::= {  
    NAMES          agreedService  
    WITH ATTRIBUTES {providedServiceId}  
    ID              id-nf-agreedServiceNameForm}  
  
userProfileNameForm NAME-FORM ::= {  
    NAMES          userProfile  
    WITH ATTRIBUTES {pui}  
    ID              id-nf-userProfileNameForm}  
  
userProfileAliasNameForm NAME-FORM ::= {  
    NAMES          userProfileAlias  
    WITH ATTRIBUTES {uptNumber}  
    ID              id-nf-userProfileAliasNameForm}  
  
calledUptUserNameForm NAME-FORM ::= {  
    NAMES          calledUptUser  
    WITH ATTRIBUTES {name}  
    ID              id-nf-calledUptUserNameForm}  
  
callingUptUserNameForm NAME-FORM ::= {  
    NAMES          callingUptUser  
    WITH ATTRIBUTES {name}  
    ID              id-nf-callingUptUserNameForm}  
  
callForwardingNameForm NAME-FORM ::= {  
    NAMES          callForwarding  
    WITH ATTRIBUTES {supplServId}  
    ID              id-nf-callForwardingNameForm}
```


The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is given below:

- the **providerId** attribute is the naming attribute for the **uptProvider** object class;
- the **organizationalUnitName** attribute is the naming attribute for the **administrativeUnit** object class;
- the **providerId** attribute is the naming attribute for the **partner** object class;
- the **providedServiceId** attribute is the naming attribute for the **agreedService** object class;
- the **uptNumber** attribute and the **pui** attribute are respectively used to name the **userProfile** and the **userProfileAlias** object classes;
- the **name** attribute is the naming attribute for both the **callingUptUser** and the **calledUptUser** object classes;
- the **supplServId** attribute is the naming attribute for the **callForwarding** object class.

5.2.2.2 Structure rules

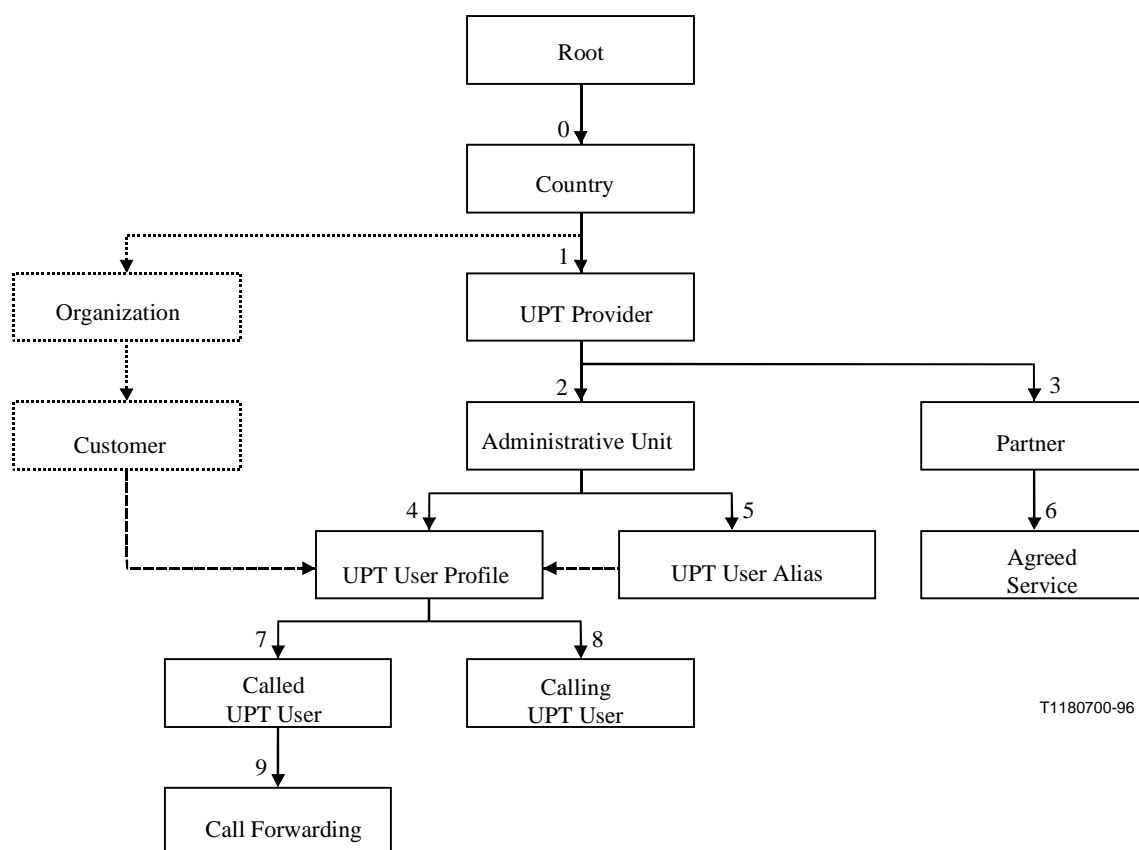


Figure 4/Q.1551 – Naming structure for UPT

In the database, a data item is located with the name of the object to which it belongs. The name of the object is the concatenation of the names of the objects superior to it in the naming structure. This implies that a hierarchical structure exists between the objects to create the object names. The structure rules provide the relationships between the objects in the naming context. This structure is independent of the structure defined for the object-classes in the inheritance context and in the existence context.

The relationships between the object-classes are represented by lines in Figure 4.

To build a name to access a given object, it is necessary to follow a path defined on the figure. For example to access **userProfile** the path is (1,3). Each object is uniquely named. However an object can have another name through the use of an alias like for **userProfile**. The **userProfile** can be named with the **userProfileAlias** that directly points to **userProfile**. The dashed arrow shows the relationship between the alias and the object class it represents whereas the plain arrows show the structure rules.

In Figure 4, the object classes **subscriberProfile** and **UserProfileAlias2** have been introduced to show how the concept of subscriber could be integrated in the model without modifying it. The two classes are not described in this Recommendation, because they are not accessed through the SCF-SDF interface. They are not used for the UPT on-line service, even though they may be used in the UPT service. The dotted lines are used to represent the object classes linked to the subscriber since they are not an integral part of this Recommendation.

The following ASN.1 description shall be used to define the structure rules for the global naming of all the object classes:

```

sr0 STRUCTURE-RULE ::= {
    NAME FORM      countryNameForm
    ID              0}

sr1 STRUCTURE-RULE ::= {
    NAME FORM      uptProviderNameForm
    SUPERIOR RULES {sr0}
    ID              1}

sr2 STRUCTURE-RULE ::= {
    NAME FORM      adminUnitNameForm
    SUPERIOR RULES {sr1}
    ID              2}

sr3 STRUCTURE-RULE ::= {
    NAME FORM      partnerNameForm
    SUPERIOR RULES {sr1}
    ID              3}

sr4 STRUCTURE-RULE ::= {
    NAME FORM      userProfileNameForm
    SUPERIOR RULES {sr2}
    ID              4}

sr5 STRUCTURE-RULE ::= {
    NAME FORM      userProfileAliasNameForm
    SUPERIOR RULES {sr2}
    ID              5}

sr6 STRUCTURE-RULE ::= {
    NAME FORM      agreedServiceNameForm
    SUPERIOR RULES {sr3}
    ID              6}

sr7 STRUCTURE-RULE ::= {
    NAME FORM      calledUptUserNameForm
    SUPERIOR RULES {sr4}
    ID              7}

```

```

sr8 STRUCTURE-RULE::= {
    NAME FORM           callingUptUserNameForm
    SUPERIOR RULES     {sr4}
    ID                  8}

```

```

sr9 STRUCTURE-RULE::= {
    NAME FORM           callForwardingNameForm
    SUPERIOR RULES     {sr7}
    ID                  9}

```

The correspondence between the parts of the definition given above and the various pieces of the notation introduced by the object class is given below:

- the **country** object class can be accessed using the **countryCode** attribute (relation 0)
- the **uptProvider** object class can be accessed from the **country** object class using the **providerId** attribute (relation 1);
- the **administrativeUnit** object class can be accessed using the **organizationalUnitName** attribute from the **uptProvider** object class (relation 2);
- the **partner** object class can be accessed using the **providerId** attribute from the **uptProvider** object class (relation 3);
- the **agreedService** object class can be accessed using the **providedServiceId** attribute from the **partner** object class (relation 6);
- the **userProfile** object class can be accessed using the **pui** attribute from the **administrativeUnit** object class (relation 4) and its alias can be accessed using the **uptNumber** attribute from the same object class (relation 5);
- the **calledUptUser** and **callingUptUser** object classes can be accessed using the **name** attribute from the **userProfile** object class (relation 7 and relation 8);
- the **callForward** object class can be accessed using the **supplServId** attribute from the **calledUptUser** object class (relation 9).

6 SCF procedures

6.1 General

6.1.1 Overview

This subclause is an introduction to the UPT-specific procedures in the SCF.

The UPT-specific procedures described in this subclause form the UPT-specific Service Logic Program (SLP) needed in the SCF to handle the UPT procedures as described in Recommendation Q.76 [2]. An SLP is normally implementation-dependent and should not be standardized in general. However this Recommendation gives a global view of the UPT service and puts in context the database requests that have to be standardized.

These procedures are:

- the common sequences: these elementary procedures are independent of the actual UPT procedures but are executed before and after them. They include the Identification and Authentication (IA) procedure, the Feature Request Identification (FRI) procedure, the Release (RELEASE) procedure, the SRF Connection procedure (SRF_Connect) and the SRF Disconnection macro (SRF_Disconnect). They are further described in 6.2;
- the personal mobility procedures: this includes the Registration for Incoming Calls procedure (REG_IN) and the Deregistration for Incoming Calls (DEREG_IN). They are further described in 6.3;

- the UPT call handling procedures: this includes the Outgoing UPT Call procedure (OUTCALL) and the Incoming UPT Call procedure (INCALL). They are further described in 6.4;
- the service profile management procedures: this includes the Service Profile Modification (SPM) procedure and the Service Profile Interrogation (SPI) procedure which is further described in 6.5.

The UPT-specific SLP is invoked when an instance of the SCF State Model (SCSM) is created in the SCF FSM on receipt of an IN request related to the UPT service (i.e. indicated by the value of the "serviceKey" parameter of the InitialDP operation). This occurs when the SSF detects the presence of an incoming UPT call or of a UPT user request. The recognition of a UPT request is based on the format of the UPT number, UPT access code or UPT access number. This part of the SLP is also called the Access procedure and is executed in all the SLP Invocations (SLPIs).

The FSM behaviour is described in Recommendation Q.1218 [1] and is partly driven by the SLP.

The description technique for the SLP in this subclause uses both text and SDLs. The SLP is modelled by a single SDL process type UPT_SLP. To each SLPI corresponds a process instance. The UPT_SLP process is further described in 6.1.3.

6.1.2 Charging procedures in the SDLs

Several types of charging procedures have been defined in Recommendation Q.1218 [1]. Depending on the operations used, the charging is performed in the SCF or in the SSF:

- "FurnishChargingInformation" is used if call records are generated at the SSF side;
- "ApplyCharging" is used if call records are generated at the SSF side and collected by the SCF. It is used when real-time transfer of charging information is necessary.

In this Recommendation all the types of charging are included. However the choice of the type of charging procedure is an implementation choice and some charging operations may not be relevant for a given implementation. The parts of the service that are to be charged should be implementation-specific.

6.1.3 Conventions and notation

Although the TC interface is not subject to standardization, some conventions are used for representing it in terms of events which virtually occur at this interface. Such events are used as input and output signals of the SDL description.

For that purpose, the TC interface is modelled using the following pseudo-events:

- Events to TC:
 - Dialogue_Released(x): the dialogue with the functional entity x has been released by the FSM or by the peer;
 - <Operation_Name>.inv: a valid TC_Invoke_Ind primitive for <Operation_Name> operation has been received by the FSM;
 - <Operation_Name>.res: a valid TC_Result_L_Ind primitive for <Operation_Name> operation has been received by the FSM;
 - <Operation_Name>.err: a valid TC_U_Error_Ind primitive for <Operation_Name> operation has been received by the FSM. This signal has a formal parameter which represents the type or error being received;
 - <Operation_Name>.rej: a valid TC_U_Reject_Ind primitive for <Operation_Name> operation has been received by the FSM.

- Events from TC:
 - Release_Dialogue(x): the FSM is requested to release the dialogue with functional entity x;
 - <Operation_Name>.inv: the FSM is requested to invoke the <Operation_Name> operation;
 - <Operation_Name>.res: the FSM is requested to send a positive result for the <Operation_Name> operation;
 - <Operation_Name>.err: the FSM is requested to send an error for the <Operation_Name> operation. This signal has a formal parameter which represents the type of error being reported.

These conventions assume that the SCF FSM performs some logical transformations on the primitives received and sent on the TC/INAP interface. These transformations are such that:

- from a receiving point of view, only the events which have an impact on the FSM are represented as input signals in the model of the TC interface;
- if several events in the FSM have the same impact, they are combined into a single input signal in the TC interface model;
- from a sending point of view, only the events which cannot be autonomously triggered by the FSM are defined as output signal in the TC interface model.

These transformations and the associated assumptions are summarized in Tables 1 and 2.

Table 1 indicates for each TC service primitive which can be received by the FSM, the corresponding event on the TC interface and (if any) the subsequent request passed to TC by the FSM according to Recommendation Q.1218 [1].

Table 2 indicates for each primitive which can be passed to TC by the FSM, whether it is generated autonomously by the FSM or at the request of the SLP. In the former case, the associated signal passed by the FSM is indicated. In the latter case the name of the corresponding event on the TC interface is also provided.

Table 1/Q.1551 – Events on the TC interface on reception of TC-primitives

Input on the TC/FSM interface	Corresponding output on the TC interface	Associated subsequent output on FSM/TC interface
TC-Begin-Ind	(SLPI creation)	–
TC-Continue-Ind Initial	Bind.res (Note 1)	–
TC-Continue-Ind Subsequent	–	–
TC-End-Ind	Dialogue_Released(x)	–
TC-Notice-Ind	Dialogue_Released(x)	TC-End-Req (Local)
TC-U-Abort-Ind	Bind.err (Note 2) or Dialogue_Released(x)	
TC-P-Abort-Ind	Dialogue_Released(x)	–
TC-Invoke-Ind (Valid)	<Operation_Name>.inv	–
TC-Invoke-Ind (Invalid)	Dialogue_Released(x)	TC-U-Reject-Req, TC-End-Req
TC-Result-Ind (Valid)	<Operation_Name>.res	–
TC-Result-Ind (Invalid)	Dialogue_Released(x)	TC-U-Reject-Req, TC-End-Req

Table 1/Q.1551 – Events on the TC interface on reception of TC-primitives (concluded)

Input on the TC/FSM interface	Corresponding output on the TC interface	Associated subsequent output on FSM/TC interface
TC-U-Error-Ind (Valid)	<Operation_Name>.err	–
TC-U-Error-Ind (Invalid)	Dialogue_Released(x)	TC-U-Reject-Req, TC-End-Req
TC-U/R-Reject-Ind	<Operation_Name>.rej (Note 3)	–
TC-L-Reject-Ind	Dialogue_Released(x)	TC-End-Req
TC-L-Cancel-Ind (Class 1,3)	Dialogue_Released(x)	TC-U-Abort-Req
TC-L-Cancel-Ind (Class 2,4)	–	–
NOTE 1 – If a Bind-Result PDU is received.		
NOTE 2 – If a Bind-Error PDU is received.		
NOTE 3 – If it appears that this signal is of no use in the SLP, it can be replaced by Dialogue_Released.		

Table 2/Q.1551 – Origins of TC-primitives

Output to TC	Originated by	Subsequent output from FSM to SLP
TC-Begin-Req	FSM Initiated	–
TC-Continue-Req	FSM Initiated	–
TC-End-Req (local)	FSM Initiated	Dialogue_Released(x)
TC-End-Req (basic)	Release_Dialogue(x) from SLP	–
TC-End-Req (basic)	FSM Initiated	Dialogue_Released(x)
TC-U-Abort-Req	FSM Initiated	Dialogue_Released(x)
TC-Invoke-Req	<Operation_Name>.inv received from SLP	–
TC-Result-Req	<Operation_Name>.res received from SLP	–
TC-U-Error-Req	– FSM Initiated (parameter missing, unexpected parameter, etc.); or – <Operation_Name>.err (error.name) received from SLP	–
TC-U-Reject-Req	FSM Initiated, followed by TC-End-Req (basic) to TC (if the dialogue exists)	Dialogue_Released(x)

The following BNF description summarizes the convention used for naming the signals exchanged across the TC interface:

<Internal_Signal> := <Dialogue_Control_Event> | <Operation_Event>

<Dialogue_Control_Event> := <Dialogue_Event_Name> <Functional_Entity>

<Dialogue_Event_Name> := "Dialogue_Released" | "Release_Dialogue"

<Functional_Entity> := "SSF" | "SRF" | "SDFo" | "SDFh" | "SDFhA" | "SDFhB"

<Operation_Event> := <Operation_Name> | <Result> | <Error> | <Reject>

<Result> := <Operation_Name> ".res"

<Error> := <Operation_Name> ".err"

<Reject> := <Operation_Name> ".rej"

<Operation_Name> := "P&C" | "PLAYANN" | "FURNCHGINFO" | "CONNTORES" |
"APPLYCHG" | "INITIALDP" | "CONNECT" | "REQREPBCSM" |
"ETC" | "ARI" | "EVREPBCSM" | "BIND" | "SEARCH" | "MODIFY" |
"ADD" | "REMOVE" | "RELEASECALL" | "SRFRPT" |
"DISCFWDCONN"

Tables 3.1 and 3.2 give the mapping between the names used in the above convention and the actual operation names. Table 3.2 gives the name of the operations on the SCF-SDF interface. These operations have to be used to fulfil the UPT service. Table 3.1 describes the other operations of Recommendation Q.1218 [1]. These operations may be used for the UPT service; however, other operations, even proprietary operations, could be used because they are applied within one network whereas SCF-SDF operations may cross network boundaries.

Table 3.1/Q.1551 – Intra-network operations used by the UPT SLP

Operation Name	SDL Signal
ApplyCharging	APPLYCHG
ApplyChargingReport	APPLYCHGRPT
AssistRequestInstructions	ARI
Connect	CONNECT
ConnectToResource	CONNTORES
DisconnectForwardConnection	DISCFWDCONN
EstablishTemporaryConnection	ETC
EventReportBCSM	EVREPBCSM
FurnishChargingInformation	FURNCHGINFO
InitialDP	INITIALDP
PlayAnnouncement	PLAYANN
PromptAndCollectUserInformation	P&C
ReleaseCall	RELEASECALL
RequestReportBCSMEvent	REQREPBCSM
SpecializedResourceReport	SRFRPT

Table 3.2/Q.1551 – Inter-network operations used by the UPT SLP

Operation Name	SDL Signal
Bind	BIND
Search	SEARCH
RemoveEntry	REMOVE
ModifyEntry	MODIFY
AddEntry	ADD

The following additional conventions are used as far as the parameter representation is concerned:

- the parameters of an INAP operation are given in a comment box next to the signal box associated with the INAP operation;
- the ASN.1 value notation is used to describe these parameters. This notation is extended to support variable parameters (noted with the suffix Var). The type of these parameters is given in Recommendation Q.1218 [1] as part of the operation definitions;
- the optional parameters depending on implantation choices are not represented.
- a counter in the SCF is used to count the number of unsuccessful attempts (failed requests) to prevent the service or the network from being misused. This counter is named Counter1.
- a counter in the SCF is used to count the number of times a database operation is sent to a busy SDF to discard the operation after too many attempts. This counter is named Counter2. Together with this counter, a timer is used to wait before sending again a database operation to the busy SDF.

6.1.4 SLP description

The UPT SLP is represented by the behaviour of the SCF in the 'UPT_SLP' process (Figure 5). This process calls several SDL procedures, each of which corresponds to one of the UPT procedures defined in Recommendation Q.76 [2]. These SDL procedures are further described from 6.2 to 6.5. This SLP reflects the options adopted by the UPT Stage 2. The ordering of the operations might be different, especially if the SLP is derived from a stage 2 based on SIBs (Recommendation Q.1213 [5]).

An instance of the 'UPT_SLP' process is created by the SCSM on receipt of an InitialDP invocation with the "serviceKey" parameter identifying the UPT service. The SCSM moves at the same time to the state "Preparing SSF Instructions". The detection point "Analysed Information" indicates the presence of an incoming UPT call or a UPT user request.

The SLPI starts with the Access procedure. If the "calledPartyNumber" includes the UPT Access Code (UPTAC) or the UPT Access Number (UPTAN), the 'UPT_SLP' process enters the Identification and Authentication procedure and the user request is further processed. Otherwise it calls the INCALL procedure and stops once it has been executed.

The "UPT_SLP" process starts by calling the IA procedure to identify and authenticate the user. After a successful authentication, the service logic executes the FRI procedure to get the feature requested by the user.

Then, depending on the feature code provided by the UPT user, the REG_IN, the DEREG_IN, the OUTCALL, the SPM or the SPI SDL procedure is called.

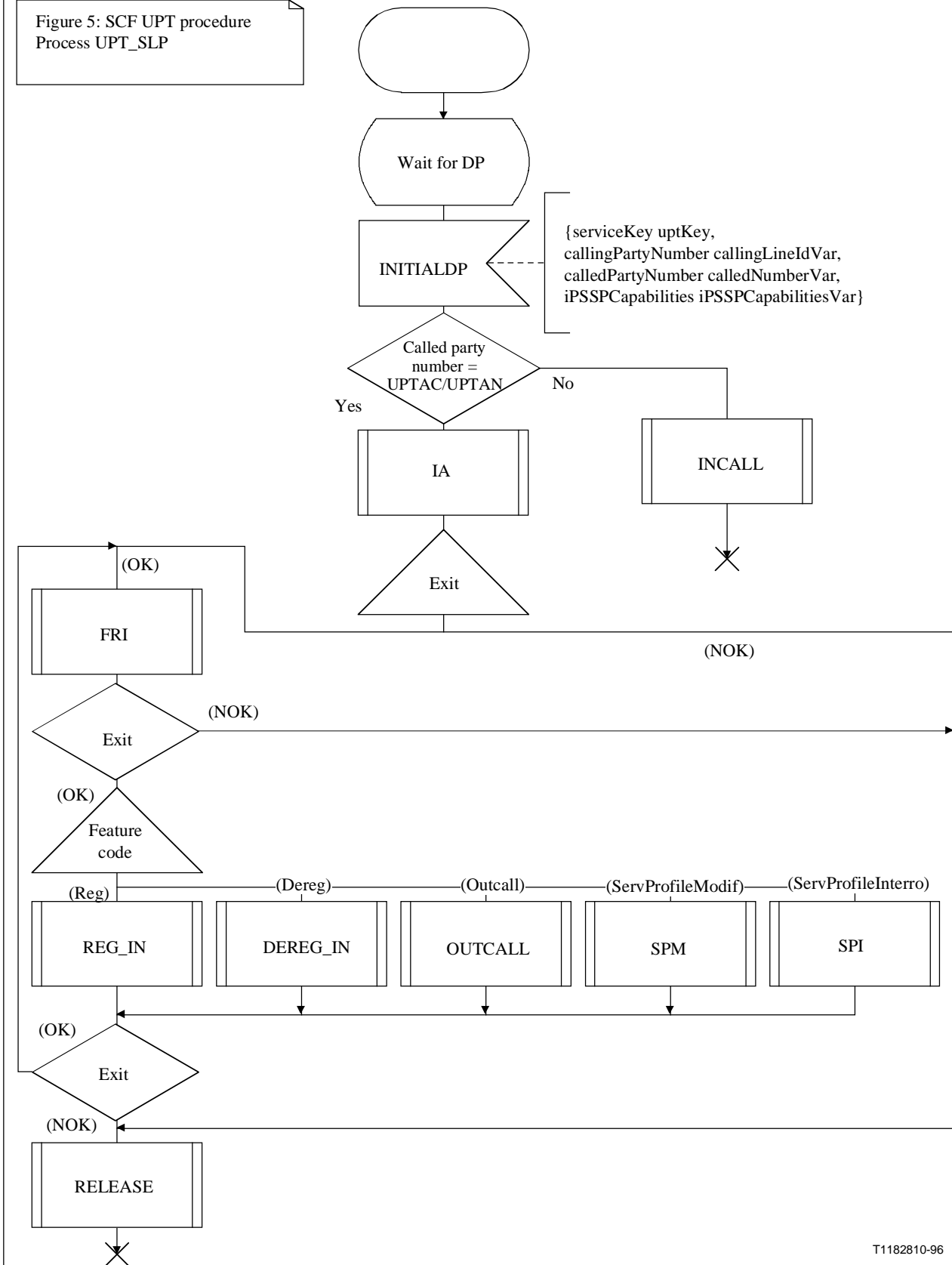
As soon as a procedure is terminated, the user can either identify a new request (new call to the FRI SDL procedure) or abandon the follow-on procedure. The follow-on procedure allows a user to perform a sequence of service features with one unique authentication. It is represented in the SDL diagram by a loop that goes from a successful service feature procedure to the FRI procedure.

At any stage of the follow-on procedure, the user can abandon. If he does so, the RELEASE procedure is called.

The user can also be released by the network with the RELEASE procedure, if authentication has failed, if an operation error has occurred or if the user has misused the UPT service. The follow-on procedure ends when the user has performed all his requests and abandons or when the user is released by the network.

In any state of the different procedures (except in the states of the user/network initiated release procedure), the user can abandon, he can be disconnected or errors for pending operations can occur. In the latter case, the dialogue with the SSF and the other functional entities are released before ending the process.

Figure 5: SCF UPT procedure
Process UPT_SLP



T1182810-96

Figure 5/Q.1551 – SCF UPT procedure – Process UPT_SLP

6.2 Generic sequences

This subclause regroups sequences of messages common to several procedures of UPT protocols. The information flows describing those sequences originated from Recommendation Q.76 [2]. They are the basis to the SDL procedures presented in the following subclauses. In the SDL specification, the timers for the various states and operations are not described, only service-related timers are shown.

6.2.1 Identification and authentication

6.2.1.1 General

The Identification and Authentication (IA) procedure takes place each time a UPT user is requested to identify himself. This is always the case when the user asks for the access to the UPT service.

The user identifies himself with his personal identifier and secret codes depending on the security option chosen and the type of terminal. If the code matches the code stored in the database for the given identifier and for a predefined algorithm, the user gets access to the UPT service and other procedures can follow.

Within the SDL procedure describing the IA procedure, the user should have the possibility to make several identification attempts. The procedure has two logical outputs²:

- OK: the procedure has succeeded and the user may proceed to the next procedure;
- NOK: the procedure has failed and the user is released by the network; the reason of this release can be either that the last permitted identification attempt has failed, that the maximum number of rejected requests is reached or that an error has occurred. The SCF and the SDF maintain a counter of the rejected attempts to have the user released by the network after a given number of consecutive retries to prevent misuse of the service and network. The other possibility to have this kind of output corresponds to the abandon of the request by the user.

At any stage of the IA procedure, the user can abandon, he can be disconnected or errors for pending operations can occur. Therefore the SDL notation "State *" is used to show that those events can occur at any state of the procedure.

6.2.1.2 Detailed procedure

Figure 6 shows the IA procedure.

The UPT request is notified to the SCF by a INITIALDP. The SSF has recognized the UPTAC (or UPTAN) in a user request. The purpose of the UPTAC (or UPTAN) is to identify the SCF that can handle UPT requests. The need for an identification and authentication of the user has been recognized. The procedure can start. The SCF sends a REQREPBSCM to request the SSF to monitor a call-related event and to send a notification back to the SCF when the event is detected. For this procedure where only one party is involved, the only types of events to be notified are the user's abandon or his disconnection of the SCF. For charging purposes, the SCF sends also a FURNCHGINF to instruct the SSF to create a call record. At the beginning of the procedure, the counter in the SCF (Counter1) for unsuccessful attempts (failed requests) to prevent the service or the network from being misused is reset.

² A parameter named Exit is used to distinguish the two outputs: OK and NOK.

Identification of the user

Then the SCF asks for the connection of an SRF; this is done through the "SRF Connect" procedure (SRF, see 6.2.4). Once the SRF is connected, the SCF is able to ask the user to provide his identity. This is done with a P&C. The SCF moves to the state "Waiting for user info". Three situations move it out of this state:

- the user has abandoned the procedure: the SCF receives an EVREPBSCSM indicating the user release. The IA procedure is terminated and followed by a release procedure. This is included in the "State *" of the SDL diagram;
- an error has occurred for the P&C operation or for the CONNTORES operation in the case of relayed operations: the SCF receives an error indication. This error indication includes a timer expiry, a rejection of an operation or an error response:
 - if the error is of type UnavailableResource in response to the P&C operation, the implementer can (if it is possible) select another SRF and restart the connection procedure. This option which is implementer-dependent is not shown on the SDL diagram;
 - if the error is an ImproperCallResponse error, the user is allowed to retry the authentication procedure from the beginning, but the number of retries is limited. A prompt informs him about his mistake and a counter of failed attempts is incremented;
 - for the remaining errors, the IA procedure is terminated with Exit = NOK. The user is released.
- the user has provided the data in a correct format: the SCF receives a P&C response and the procedure can continue as described below.

Authentication

In UPT phase 1, a user has three possible ways of authentication (simple, PIN or strong one-way) depending on the authentication procedures that the user has subscribed and on the type of terminal available. To each authentication procedure corresponds a sequence of messages. The two sequences are:

- PIN code authentication (manual): this is a two-step procedure. The user is requested to provide his identifier (PUI) and then to provide his PIN code via two consecutive P&Cs³. As it was said previously the only error for P&C that receives special treatment is ImproperCallerResponse. It is followed by a procedure that informs the SSF of the failed attempt with a FURNCHGINF0 and that allows the user to redial his authentication information or notifies him about a denied access to the UPT service depending on the number of failed attempts. This last procedure is used each time a user is suspected to have mistyped his identification sequence;
- strong one-way authentication (automatic): the user gives his identity and his variable authentication code at the same time. This is done automatically using a user device. The different pieces of data are combined with a separator '*'.

The use of a special leading digit ("*") differentiates the automatic authentication from the manual ones. After all the user information is received through a P&C response, the identity of the home service provider and the address of the database containing the user's information is extracted from

³ The user has always the possibility to combine the 2 pieces of data or more (if wanted) in one dialling sequence using a separator * to separate the different pieces of data. By this means, the user speeds up the UPT procedures.

the user identity⁴. If this operation cannot be performed (i.e. the format of the user identity is wrong), the user is allowed to re-enter his authentication data following a procedure already described above. Once the identity of the home service provider is known, a database operation checks in the local database if an agreement exists between the local service provider and the service provider of the UPT user (home service provider). There are three outcomes to this query:

- the user has abandoned the procedure: the SCF receives an EVREPBSCM indicating the user's release. The IA procedure is terminated and followed by the release procedure;
- an error has occurred: the SCF receives an error indication:
 - if a service error of type Busy occurs, the implementer has the possibility (if desired) to resend the database operation to this same SDF after a given time set with a timer. This remark applies to all error configurations where this type of error can occur (i.e. when waiting for a database answer). The Counter2 counts the number of retries;
 - otherwise the IA procedure is terminated and followed by the release procedure;
- a response to the database operation has been received: the content of the response is checked. If there is no agreement between the service providers, the SSF is informed of the end of the call and is requested to modify the call record by a FURNCHGINFO. The user is informed by a prompt that he cannot have access to the service and is later released. If there is an agreement between the service providers, a check with the home service provider takes place to authenticate the user and to know if the user is allowed to use the UPT service in that area.

If an agreement exists between the two service providers, the SCF starts an authenticated dialogue with the home database of the user. The dialogue is opened with a BIND operation whose argument contains all the authentication information provided by the user. The number of failed authentication attempts is limited for a given user's identifier. The SDFh monitors and keeps the count of the number of attempts. The counter used is attached to the PIN attribute. The three possible outcomes to this query are:

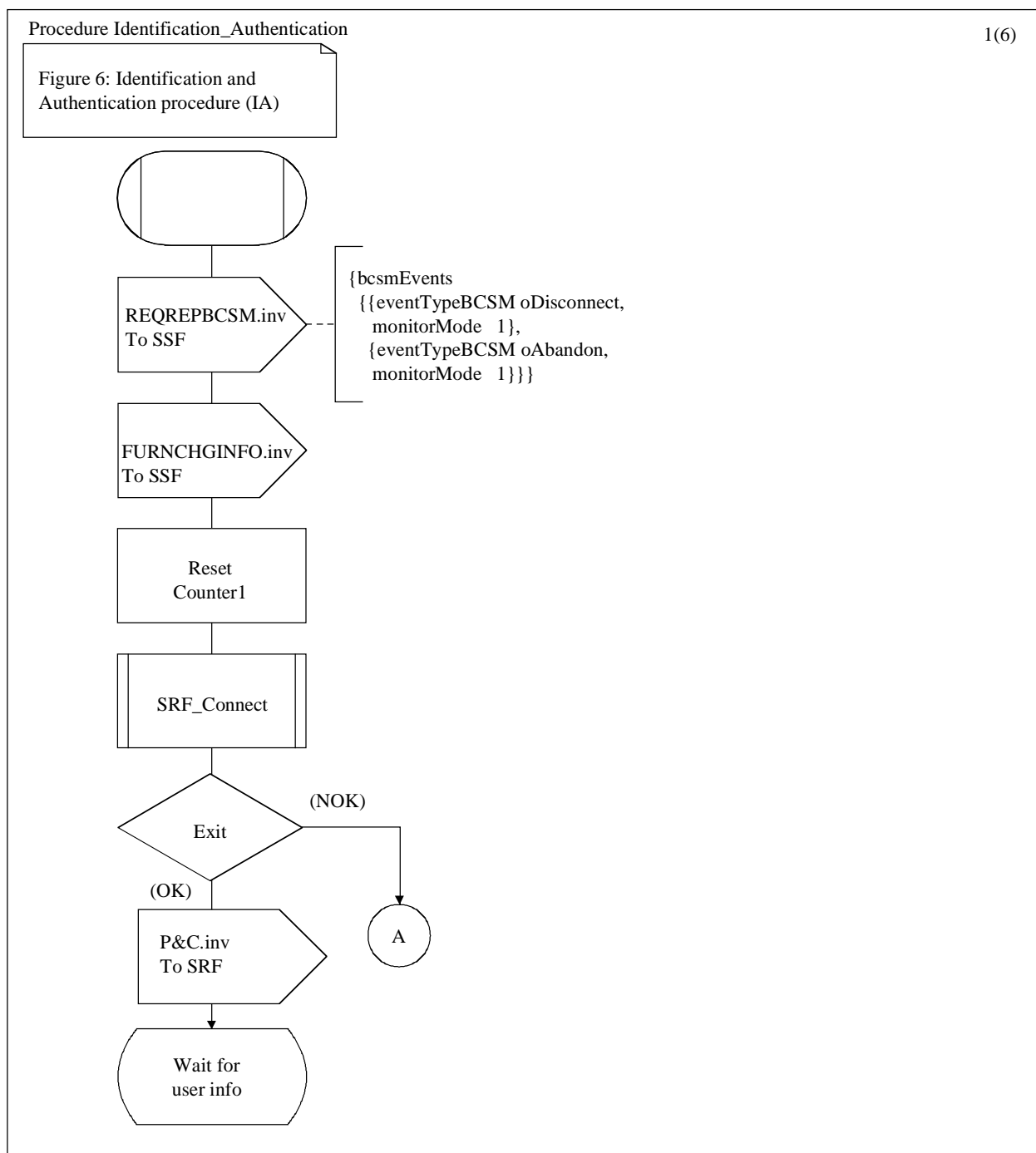
- the user has abandoned the procedure: the SCF receives an EVREPBSCM indicating the user release. The IA procedure is terminated and followed by a release procedure;
- the BIND operation fails: the SCF receives an error indication:
 - if the error is a service error (unavailable), the implementer can (if desired) resend the database operation to this same SDFh after a given time;
 - if the error is a security error (blocked credentials), indicating that the maximum number of authentication attempts has been reached, the call record is updated with a FURNCHGINFO and before being released by the network, the user is informed by a PLAYANN that his line/number is blocked;
 - if the error is a security error (invalidCredential), the SSF is requested to modify the call record (FURNCHGINFO) and the user is requested to restart dialling his identification information;
 - if the error is a security error (inappropriateAuthentication), the SSF is requested to modify the call record (FURNCHGINFO) and the user is informed by a PLAYANN;
 - otherwise the IA procedure is terminated and followed by a release procedure.
- the BIND operation succeeds: the user can go on to the next procedure (feature identification request, see 6.2.2).

⁴ It is assumed that the identifier contains enough information to know the home service provider and the corresponding SDF.

Once the authentication has been successfully performed, the SRF instructed by the SCF sends a prompt to the user requesting him to enter the code of the feature service he wants to access. This is only valid for all authentications and when the user has not yet entered the feature code.

NOTE – To decrease the signalling load on an inter-network interface, a first database operation could be sent with the authentication information in the BEGIN message. The operation to be sent would be the first database operation encountered in the FRI procedure (check on agreement between service providers). For the sake of simplicity, the different operations are treated separately.

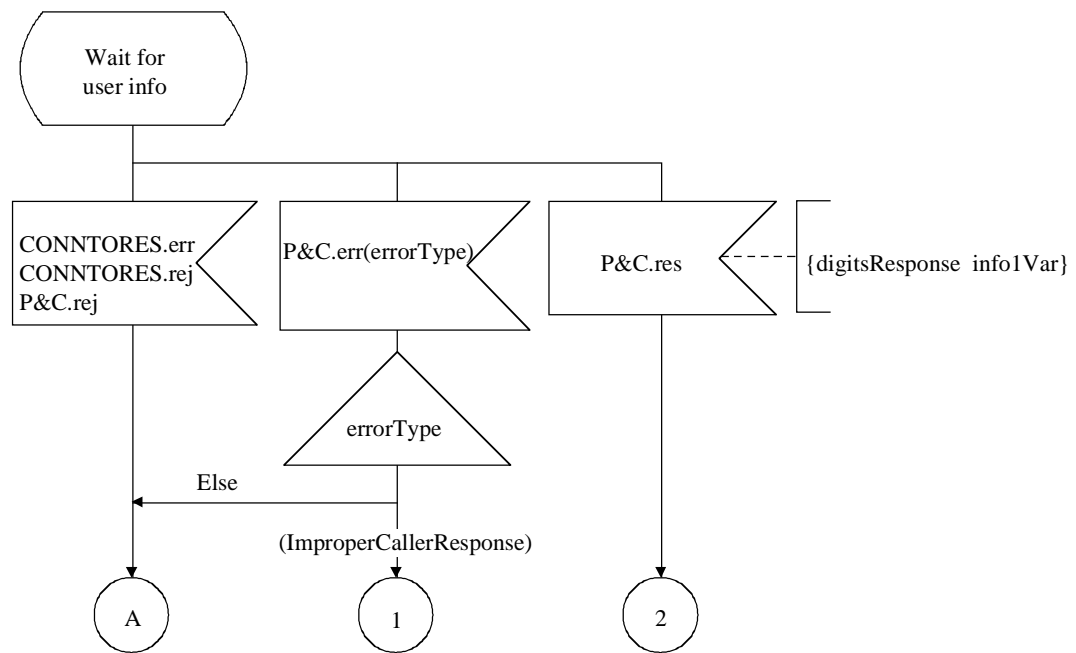
Before a release after a prompt, the SRF is disconnected. The disconnection is described in 6.2.5 with a macro.



T1182820-96

Figure 6/Q.1551 (sheet 1 of 6) – Identification and Authentication (IA) procedure

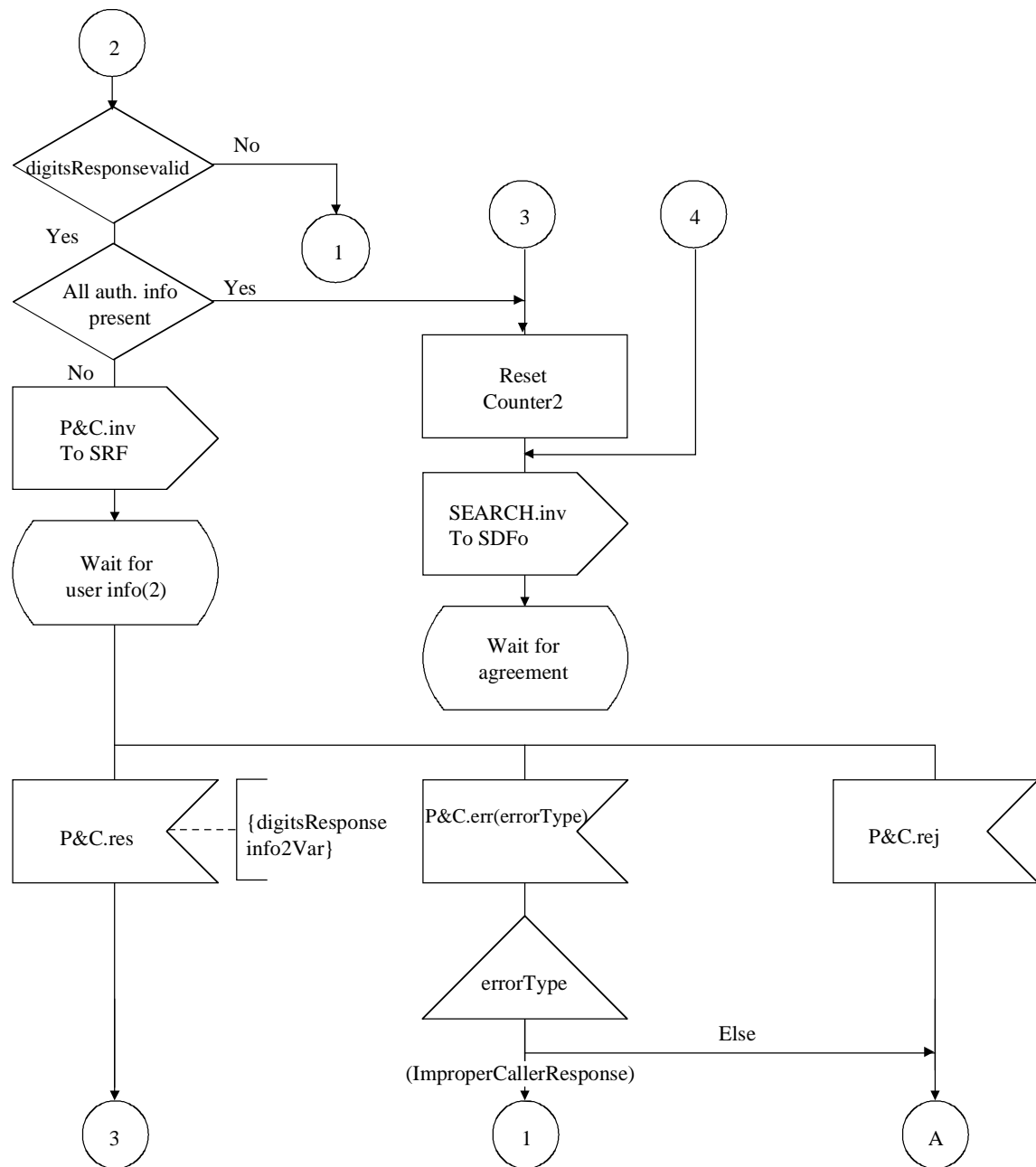
Figure 6: Identification and Authentication procedure (IA)



T1182830-96

Figure 6/Q.1551 (sheet 2 of 6) – Identification and Authentication (IA) procedure

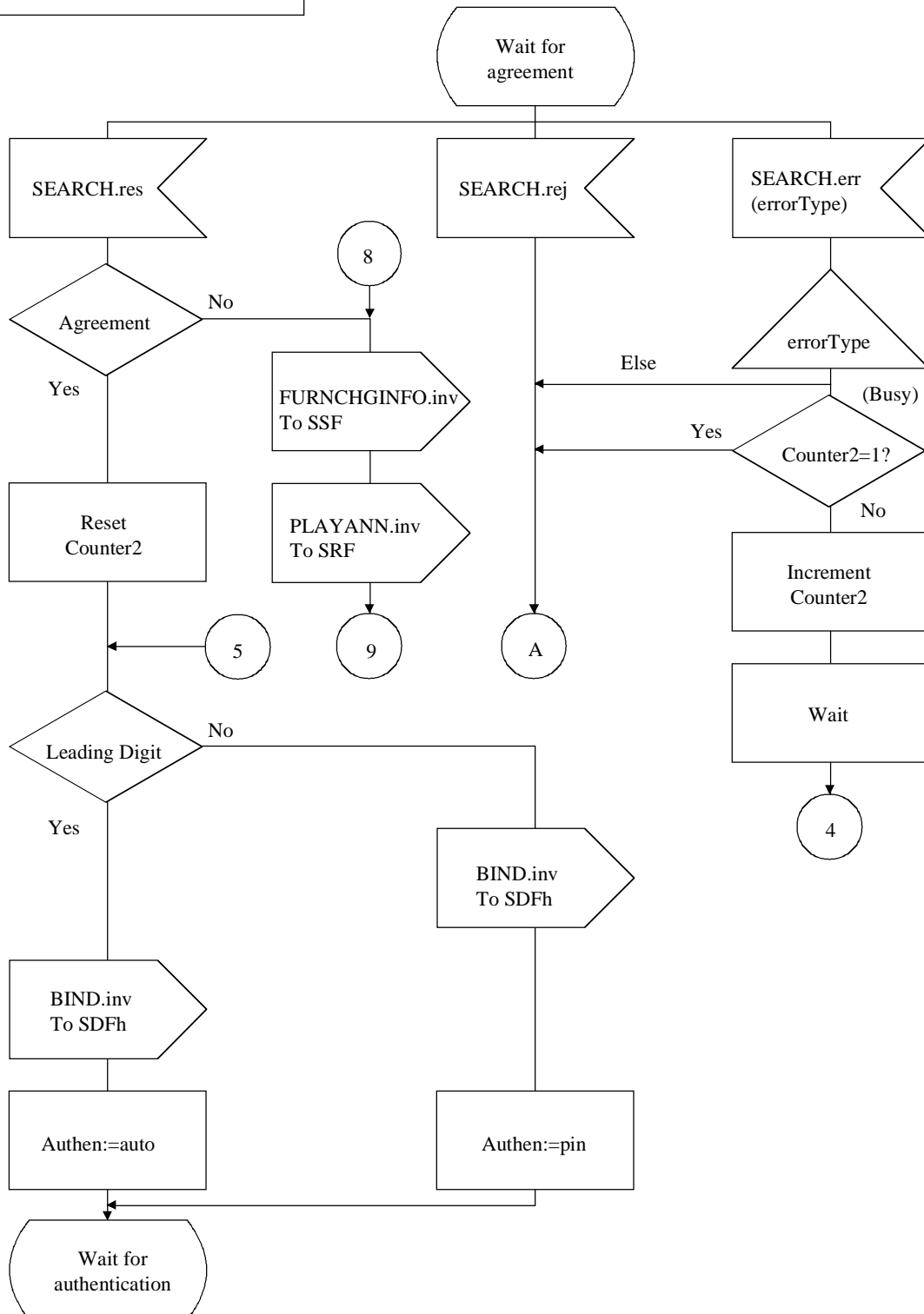
Figure 6: Identification and Authentication procedure (IA)



T1182840-96

Figure 6/Q.1551 (sheet 3 of 6) – Identification and Authentication (IA) procedure

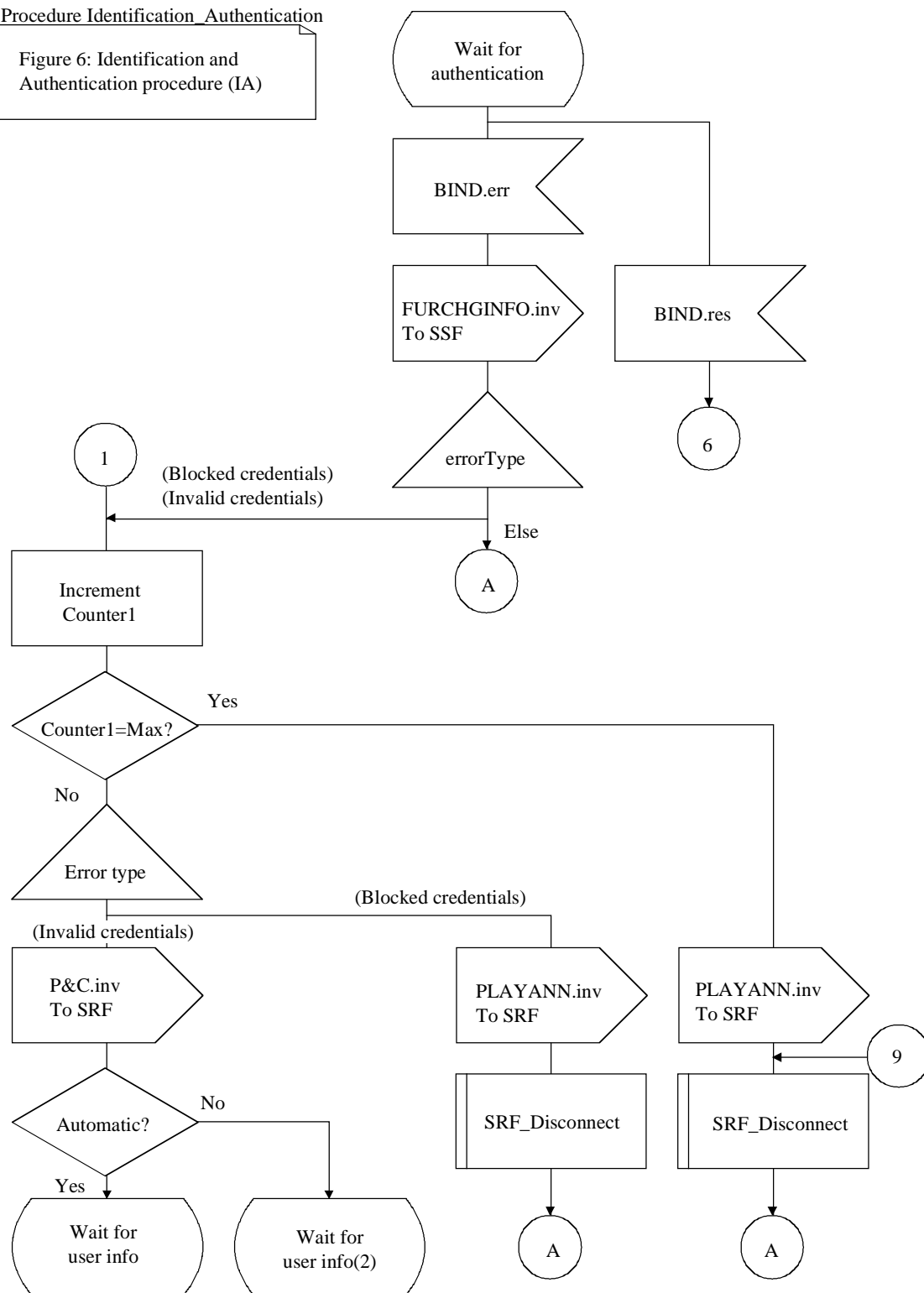
Figure 6: Identification and Authentication procedure (IA)



T1182850-96

Figure 6/Q.1551 (sheet 4 of 6) – Identification and Authentication (IA) procedure

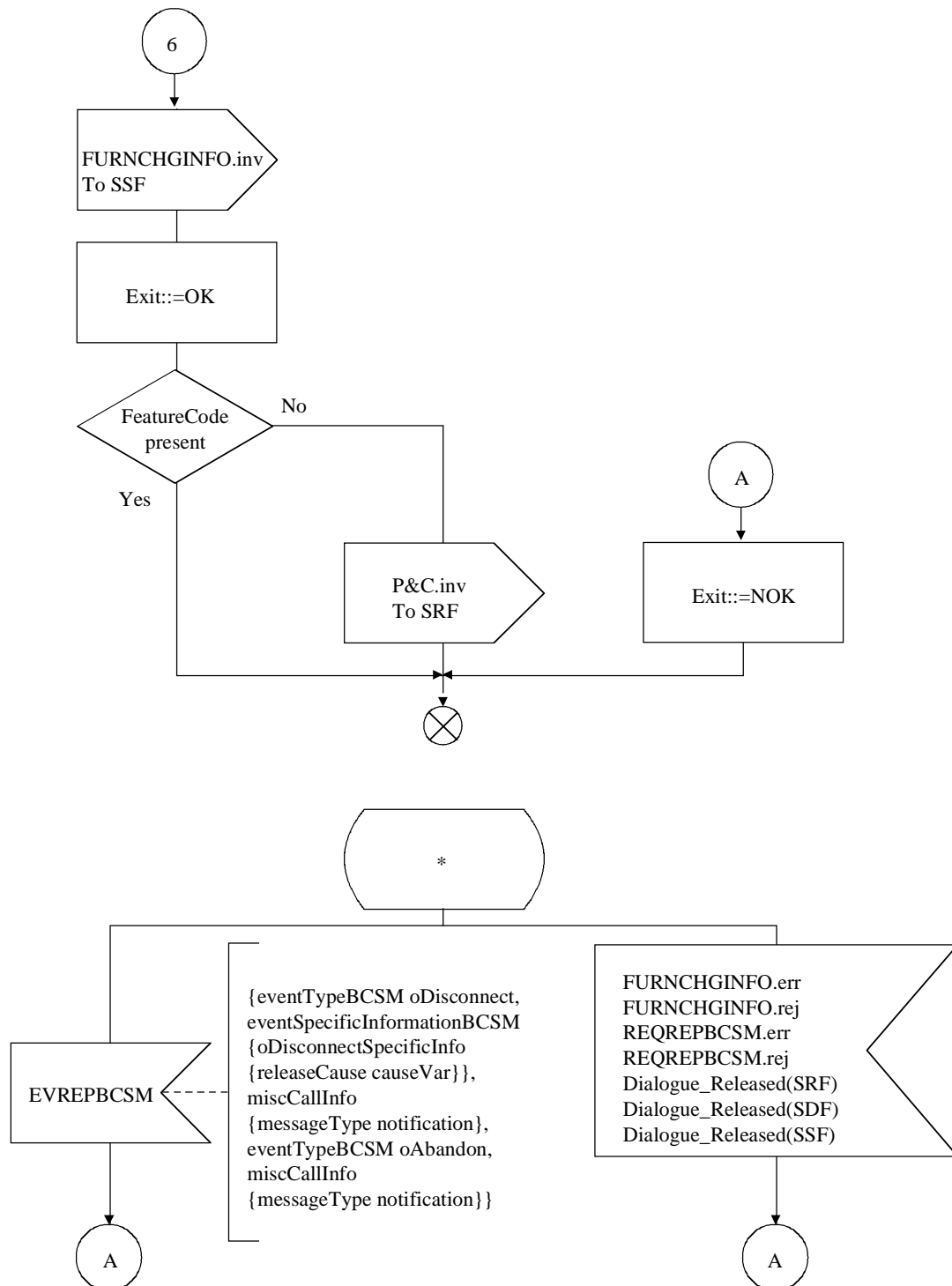
Figure 6: Identification and Authentication procedure (IA)



T1182860-96

Figure 6/Q.1551 (sheet 5 of 6) – Identification and Authentication (IA) procedure

Figure 6: Identification and Authentication procedure (IA)



T1182870-96

Figure 6/Q.1551 (sheet 6 of 6) – Identification and Authentication (IA) procedure

6.2.2 Feature request identification

6.2.2.1 General

The Feature Request Identification (FRI) procedure takes places:

- after a successful authentication;
- within the follow-on procedure after any other procedure.

It is used to identify a feature request and check if the feature can be supported for the user.

The associated procedure is named Feature Request Identification (FRI). It has two logical outputs. They are identical to the ones for the IA procedure.

Like in the IA procedure, at any stage of the FRI procedure, the user can abandon, he can be disconnected or errors for pending operations can occur. Therefore the same "State *" applies to the FRI procedure.

6.2.2.2 Detailed procedure

The SDL diagram for this procedure is shown in Figure 7.

Identification of the service feature

The SCF starts to check if the feature was already given by the user. If so, no P&C is sent to the user to ask him for the requested feature code. If not, the P&C is sent and the SCF moves to the state "Wait for feature code". It is waiting for the user response to a previous P&C. There are three responses to expect:

- the user has abandoned the procedure: the SCF receives an EVREPBCSM indicating the user's release. The FRI procedure is terminated and followed by a release procedure;
- an error has occurred: the SCF receives an error indication. What follows is a typical error treatment for a P&C. Only ImproperCallerResponse is considered. It is followed by a procedure that informs the SSF of the failed attempt with a FURNCHGINFO and that allows the user to redial the feature code or that notifies him about a denied access to the UPT service after the maximum number of retries is reached;
- the user has answered the request: the SCF receives a P&C response.

Service provider agreements

Once the user information is collected, the user information is used to send a database operation to the SDFo. This operation checks if particular agreements exist between the local service provider and the user's home service provider for the requested feature. The three possible outcomes are:

- the user has abandoned the procedure: the SCF receives an EVREPBCSM indicating the user release. The FRI procedure is terminated and followed by a release procedure;
- an error has occurred: the SCF receives an error indication:
 - if a service error of type Busy occurs, the implementer can resend the database operation to this same entity after a given time;
 - otherwise the FRI procedure is terminated and the user is released by the network.
- a response to the database operation has been received: the content of the response is checked. If there is no agreement between the service providers, the SSF is informed of the failure of the FRI procedure and is requested to modify the call record by a FURNCHGINFO. Depending on the number of retries available, the user is informed with a P&C that he can perform another FRI procedure or he is informed with a PLAYANN that he will be released by the network. If there is an agreement between the service providers, the user has access to the requested feature.

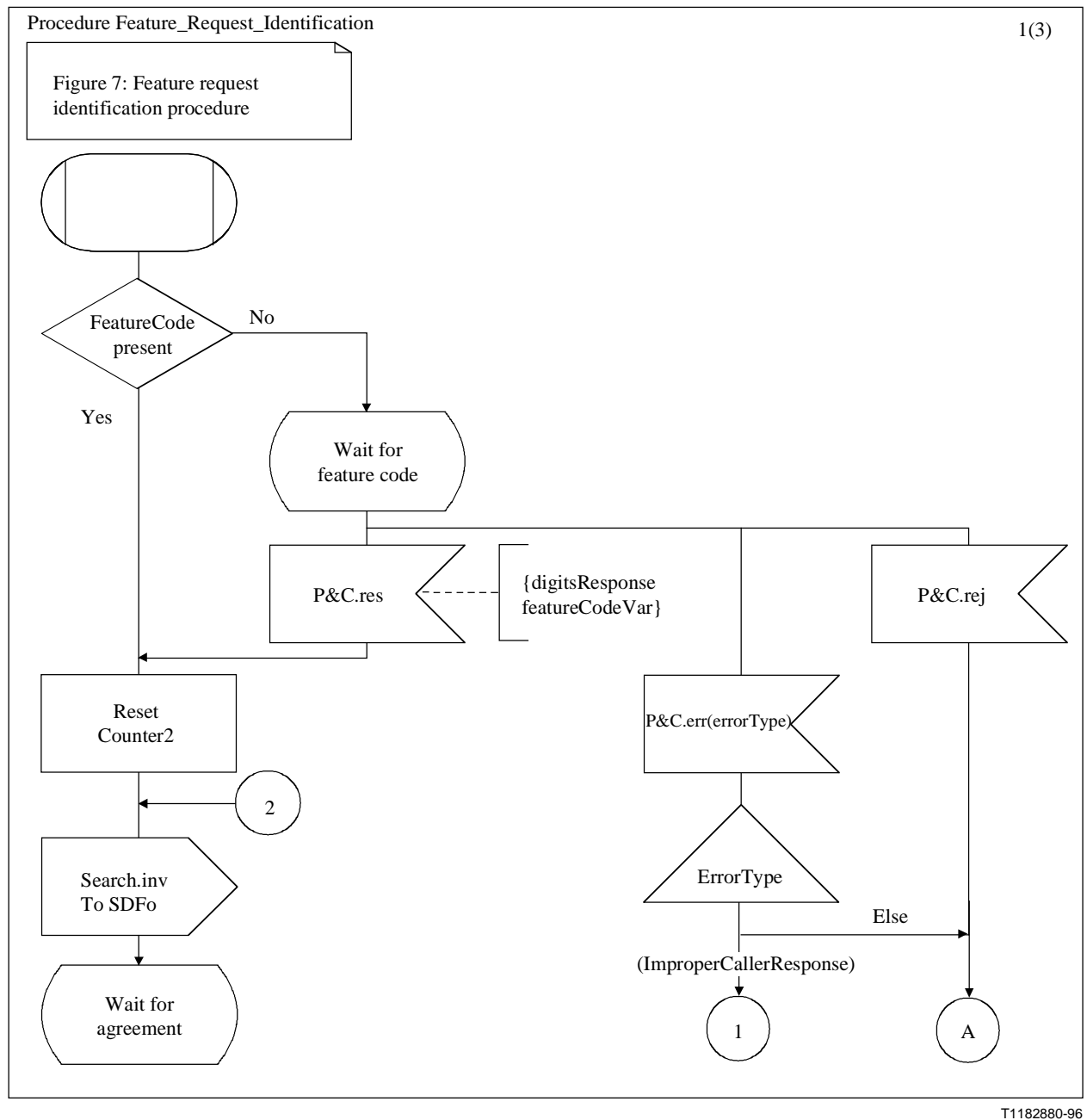
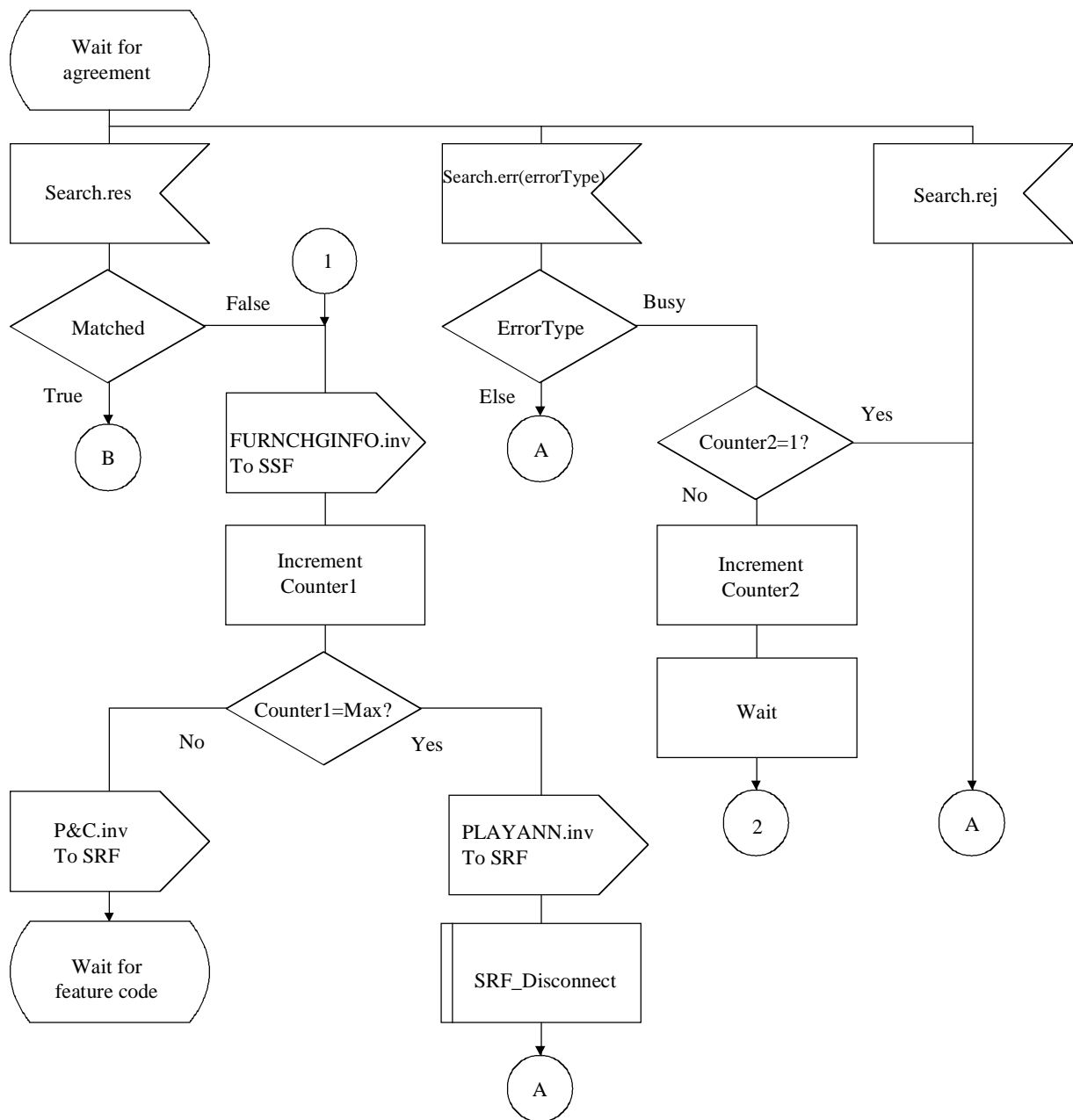


Figure 7/Q.1551 (sheet 1 of 3) – Feature Request Identification (FRI) procedure

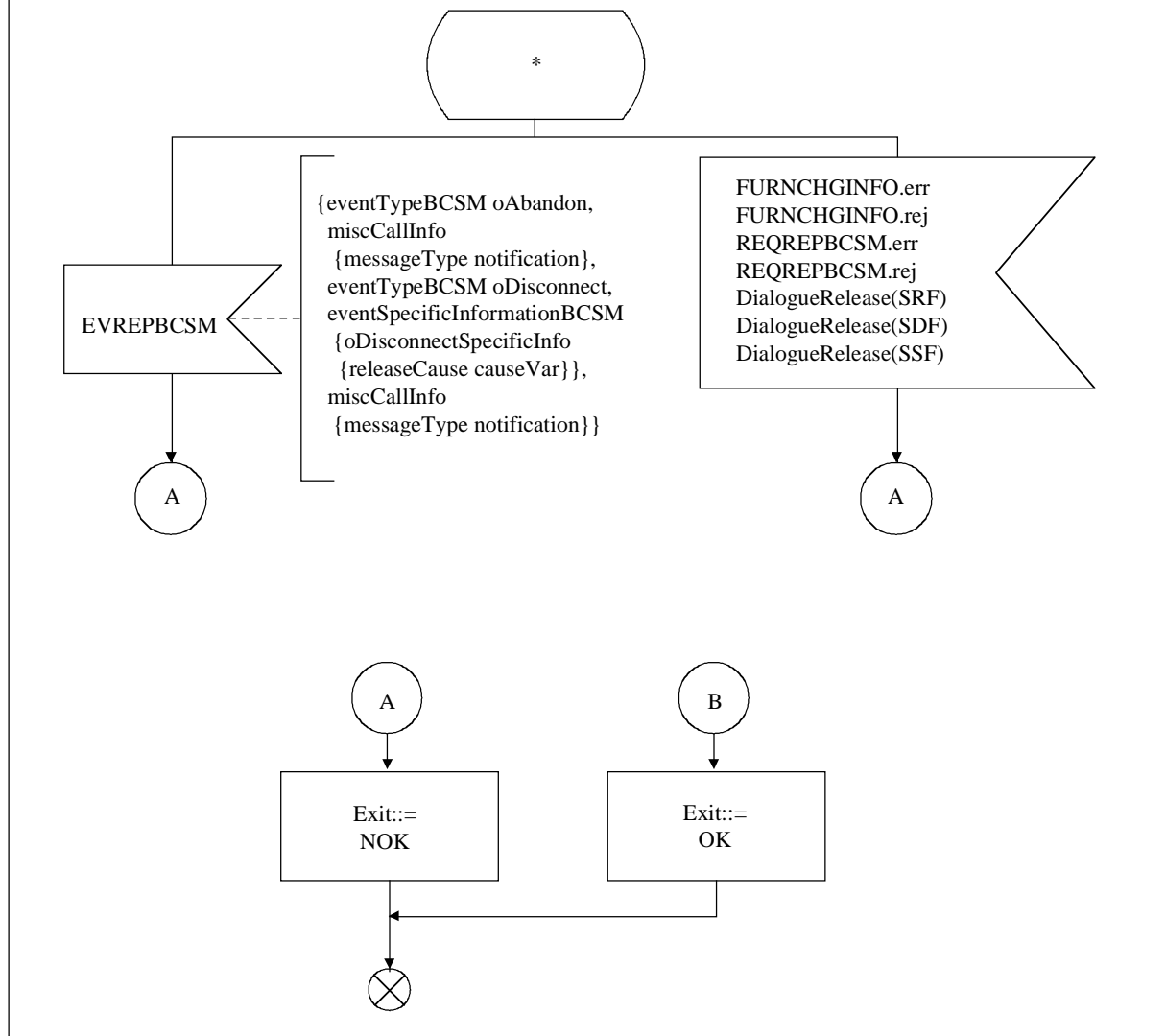
Figure 7: Feature request identification procedure



T1182890-96

Figure 7/Q.1551 (sheet 2 of 3) – Feature Request Identification (FRI) procedure

Figure 7: Feature request identification procedure



T1182900-96

Figure 7/Q.1551 (sheet 3 of 3) – Feature Request Identification (FRI) procedure

6.2.3 Release of the calling user

6.2.3.1 General

The release of the calling user takes place at the end of every UPT user request (including one or more feature requests).

The following situations may be encountered:

- release initiated by the user hanging up in any state. The release may also be initiated by the originating network;
- forced release initiated by the IN-node.

Even though the two types of released are triggered by different events, the release procedure is the same in both cases. The release procedure generally includes the release of all the external

connections (lines, circuits, etc.) and the release of all engaged IN resources (intelligent peripheral, etc.).

6.2.3.2 Detailed procedure

The SDL diagram for this procedure is shown in Figure 8.

At the beginning of the procedure, the SCF sends a Call Release to the SSF and closes the dialogues established with the other entities. The procedure is then terminated.

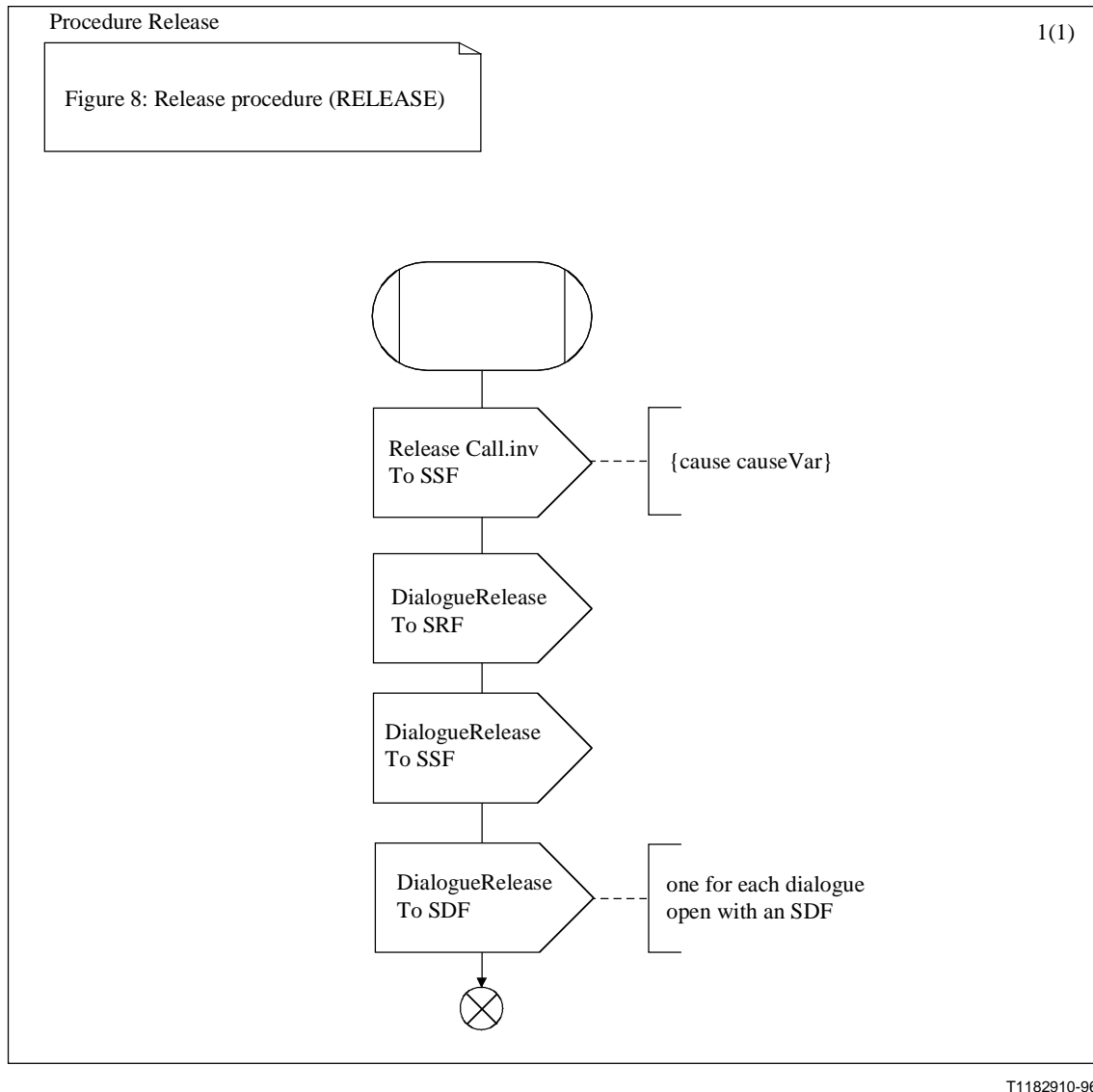


Figure 8/Q.1551 – Release procedure (RELEASE)

6.2.4 Connection of an SRF

6.2.4.1 General

The connection of an SRF takes place when there is a need to send a message to a user. It is also used to collect information provided by the user. For example, this procedure occurs at the beginning of the IA procedure to send messages to the user and to receive back provided information (like

authentication information). It also occurs for the follow-on procedure following an outgoing call procedure. It allows the SCF to interact with the user.

Two technical options are available to connect an SRF. They depend on the type of interface used between the SCF and the SSF or on whether the operations from the SCF are relayed through the SSP (i.e. SSF and SRF integrated or not). Both options are presented in the procedure.

The associated procedure is named SRF connection (SRF_Connect). It has two logical outputs. They are identical to the ones for the FRI procedure. Like in the IA and in the FRI procedures, at any stage of the procedure, the user can abandon.

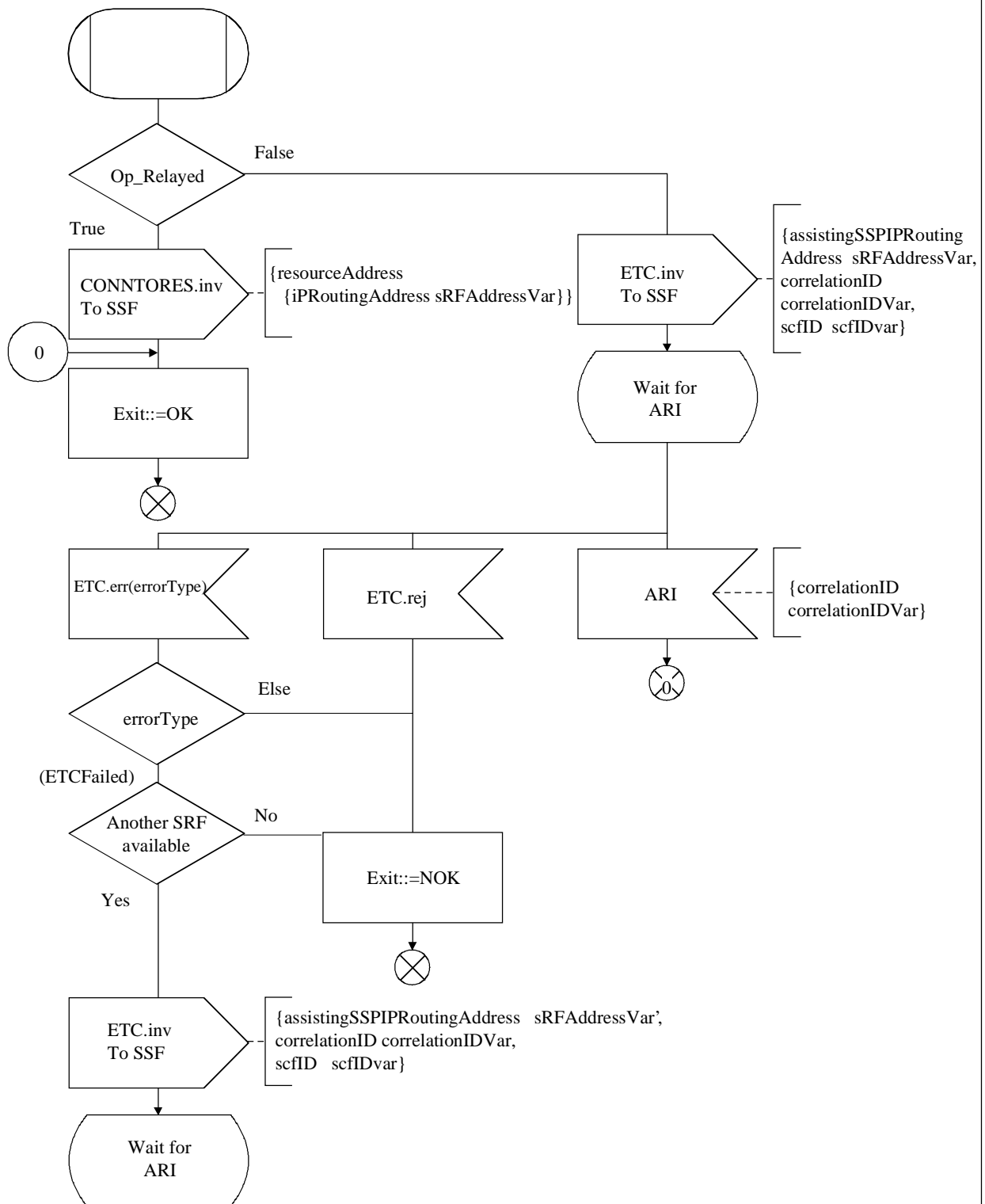
6.2.4.2 Detailed procedure

The SDL diagram for this procedure is shown in Figure 9.

The technical choice concerning the relation between the IP and the SSP is left open in the SDL procedure. An attribute internal to the SCF called Op_Relayed contains indications on whether the operations are relayed or not by the SSP. To show this implementation choice, the value of the attribute is checked and the following sequence of the messages depends on the result of the check. However this choice is not present in a given implementation where only one message sequence is selected. The two possible sequences are:

- operations relayed: the SCF sends a CONNTORES to the SSF. The message orders the SSF to connect an SRF;
- operations not relayed: the SCF sends an ETC to the SSF and waits for the confirmation of the connection with an SRF through an ARI message. If the confirmation is received, the procedure normally ends (exit = OK). If the confirmation is not received, this means that a problem arose with the previous ETC operation. Either the operation was rejected and the user should be released by the network or an error was detected in the processing of the operation. The only case of error recovery is for the ETCFailed error. Then the SCF has the possibility of contacting a new SRF (if any available) via a new ETC operation. The other error cases lead to a release (exit = NOK).

Figure 9: SRF Connect procedure (SRF_Connect)



T1182920-96

Figure 9/Q.1551 – SRF Connect procedure (SRF_Connect)

6.2.5 Disconnection of an SRF

6.2.5.1 General

The disconnection of an SRF takes place either when a message has been sent to a user announcing to him the end of his request or when a call initiated by the user is about to be processed. It is used to release the resources engaged in the dialogue between the SRF and the SCF. This procedure occurs at the end of a user request or when the resources are needed to set up a call.

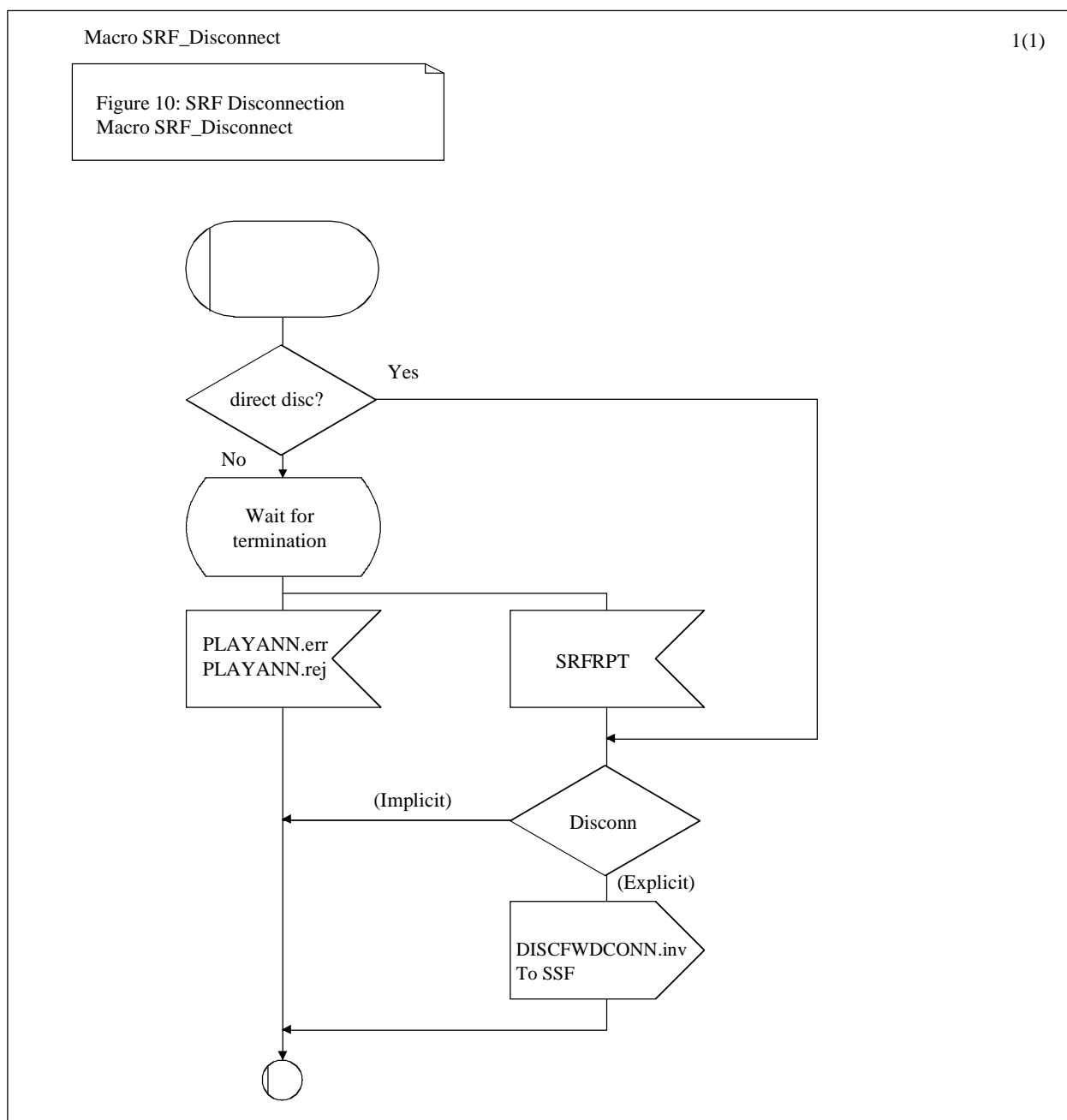
Two technical options are available to disconnect an SRF. They depend on the type of interface used between the SCF and the SSF or on whether the operations from the SCF are relayed through the SSP (i.e. SSF and SRF integrated or not). Both options are presented in the procedure.

The associated macro is named SRF disconnection (SRF_Disconnect). It has one logical output.

6.2.5.2 Detailed procedure

The SDL diagram for this procedure is shown in Figure 10.

Either the disconnection takes place after an announcement has been sent or after the abandon of the B user. In the first case an SRFRPT is expected whereas in the other case the disconnection can be executed right away. This is represented in the choice "direct disc". There are two possible ways to disconnect a SRF either by using the DISCFWDCONN operation or by a backward disconnect from the SRF. The choice between the two alternatives is controlled by the use of the "DisconnectFromIPForbidden" parameter of the PLAYANN and P&C operations. The variable 'disconn' is used instead of the full parameter name to have shorter names on the diagram.



T1182930-96

Figure 10/Q.1551 – SRF Disconnection – Macro SRF_Disconnect

6.3 Personal mobility

Personal Mobility Procedures are UPT procedures relating to the personal, or discrete, mobility of the UPT user, used in order to ensure that the UPT user is able to receive or make UPT calls. The personal mobility procedures do not involve, however, actual making or receiving of calls.

6.3.1 Registration for incoming calls

6.3.1.1 General

Registration for incoming calls is a feature by which a UPT user registers from the current terminal access for incoming UPT calls to be presented to that terminal access.

When registered, all incoming calls to UPT user will be presented to that terminal access for the duration specified by the UPT user (by a new registration or an explicit deregistration) or until a specified registration limitation. A UPT user's Incall Registration will cancel the previous one of that UPT user. Several UPT users may be registered for incoming calls to the same terminal access simultaneously.

The Identification and Authentication (IA) and Feature Request Identification (FRI) procedures must have been successfully completed before this procedure.

6.3.1.2 Detailed procedure

Figure 11 shows the Registration for Incoming Calls (REG_IN) procedure.

The REG_IN procedure is invoked by the UPT_SLP process when the UPT user answers with the Incall Registration Code during the FRI procedure. In order to simplify the description of the procedure, this subclause is structured in subclauses describing the different parts of the registration.

Terminal ID available

When the terminal ID is provided by the user with the feature code, the SCF formulates and sends a P&C operation to the SRF in order to ask the user if he wants to register at the terminal specified by the terminal ID and waits for the answer (state "Wait for User ACK"). The processing continues as described in subclause "User's acknowledgement for terminal ID".

CLI available

When CLI (Calling Line Identity) is available, the SCF formulates and sends a P&C operation to the SRF and waits for the answer (state "Wait for CLI ACK"). The SRF receives and reacts to the P&C and plays to the user the requested announcement asking the UPT user to indicate whether the registration will be at that terminal or not.

The following events move the SCF out of "Wait for CLI ACK" state:

- the user has abandoned the procedure: the SCF receives an EVREPBSCSM indicating the user release. The procedure is terminated with Exit = NOK. This is included in the "State *" of the SDL diagrams;
- an error has occurred for the P&C operation (P&C.err):
 - if the error is "ImproperCallerResponse", the SCF sends a FURNCHGINF0 to request the SSF to create a record for the user request, the number of retries is incremented and, if not exceeded, the prompt is repeated to the user that is allowed to make another attempt. If the number of retries is exceeded the processing continues as described in subclause "Maximum number of retries exceeded";
 - else for any other error the procedure is terminated with Exit = NOK.
- the P&C operation is rejected (P&C.rej): the procedure is terminated with Exit = NOK;
- the user answers "yes" to the prompt: the processing continues as described in subclause "UPT user's permission screening";
- the user answers "no" to the prompt: the processing continues as described in subclause "CLI not available";
- the user answers "esc" to the prompt: the SCF sends a FURNCHGINF0 to request the SSF to create a record for the user request, the number of retries is incremented and, if not exceeded, the SCF formulates and sends a P&C operation indicating that the user's request has been cancelled and allowing him to return to the Feature Request Identification (FRI) (follow-on). If the number of retries is exceeded the processing continues as described in subclause "Maximum number of retries exceeded".

CLI not available

If CLI is not available or the user's response to register on the current terminal is "no", the desired terminal identity is requested. The SCF formulates and sends a P&C operation to the SRF and waits for the answer (state "Wait for Terminal ID"). The SRF receives and reacts to the P&C and plays to the user the requested announcement to appeal for the terminal identity on which the user wants to register.

The following events move the SCF out of "Wait for Terminal ID" state:

- P&C operation is successful: the SRF echoes the received terminal identity to the user and sends the dialled digits to the SCF. The SCF formulates and sends the P&C operation in order to prompt the user to confirm or cancel the input data (state "Wait for User ACK"). This is described in subclause "User's acknowledgement for terminal ID";
- an error has occurred for the P&C operation (P&C.err):
 - if the error is "ImproperCallerResponse", the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries is incremented and, if not exceeded, the user is requested to enter valid information with a P&C operation. If the number of retries is exceeded the processing continues as described in subsection "Maximum number of retries exceeded";
 - else for any other error the procedure is terminated with Exit = NOK.
- the P&C operation is rejected (P&C.rej): the procedure is terminated with Exit = NOK.

Maximum number of retries exceeded

If the maximum number of retries is exceeded a PLAYANN is sent to SRF in order to inform the user to hang up, the procedure is terminated with Exit = NOK and the calling user is released by the Release procedure.

User's acknowledgement for terminal ID

If the user confirms the echoed terminal ID the processing continues as described in subclause "UPT user's permission screening".

If the user dials "esc" the same treatment described in subclause "CLI available" is applied.

If the user does not confirm the echoed terminal ID, the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request and the number of retries counter is incremented. If the maximum number of retries is not exceeded, a new attempt is allowed. The SCF formulates and sends a P&C to the SRF in order to prompt the user to retry the registration procedure and collect again the registration input data and returns to the "Wait for Terminal ID" state. If the number of retries is exceeded the processing continues as described in subclause "Maximum number of retries exceeded".

UPT user's permission screening

The SCF formulates and sends a Search operation to the local SDF to verify the UPT user's permission to register on a specified network access and wait for the SDFo answer (state "Wait for Screen results").

The following events move the SCF out of "Wait for screen results" state:

- an error has occurred for the Search operation (SEARCH.err):
 - if the error is "Busy" the Search operation may be sent again to the same SDF after a time-out (timer) and the "UPT user's permission screening" subclause is repeated. Note

that Figure 11 only shows the possibility of one retry, the maximum number of retries is Network Specific;

- if the error is "Insufficient Access Rights", the procedure is terminated as described in subclause "Unsuccessful registration";
 - for any other error or when COUNTER2 is exceeded the procedure is terminated with Exit = NOK.
- the Search operation is rejected (SEARCH.rej) the procedure is terminated with Exit = NOK;
 - a response to the Search operation has been received. The Counter2 is reset and the ResultArg received is checked:
 - if the registration is allowed the processing continues as described in subclause "Collect limitation information";
 - if the registration for incoming calls is not allowed, the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request and the number of retries counter is incremented. If the maximum number of retries is not exceeded, a new attempt is allowed. The SCF formulates and sends a P&C to the SRF in order to prompt the user to retry the registration procedure and collect again the registration input data and returns to the "Wait for Terminal ID" state. If the number of retries is exceeded the processing continues as described in subclause "Maximum number of retries exceeded".

Unsuccessful registration

The SCF formulates and sends FURNCHGINFO to request the SSF to send a record for all user requests and the number of retries counter is incremented. If the maximum number of retries is not exceeded, the SCF formulates and sends a P&C operation in order to inform the user that the request cannot be handled and to terminate or request another procedure (this is done by the FRI procedure). If the maximum number is exceeded, the processing continues as described in subclause "Maximum number of retries exceeded".

Collect limitation Information

If the registration is allowed, the SCF formulates and sends P&C operation to SRF to request the user to provide time limitation information and waits for the answer (state "Wait for limitation sequence"). The following events move the SCF out of "Wait for limitation sequence" state:

- P&C operation is successful: the SRF echoes the received limitation sequence to the user and sends the dialled digits to the SCF. The SCF formulates and sends the P&C operation to SRF in order to prompt the user to confirm or cancel the input data (state "Wait for limitation ACK."). This is described in subclause "User's acknowledgement for limitation sequence";
- an error has occurred for the P&C operation (P&C.err):
 - if the error is "ImproperCallerResponse", the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries is incremented and, if not exceeded, the user is requested to enter valid information with a P&C operation. If the number of retries is exceeded, the processing continues as described in subclause "Maximum number of retries exceeded";
 - else for any other error the procedure is terminated with Exit = NOK.
- the P&C operation is rejected (P&C.rej): the procedure is terminated with Exit = NOK.

User's acknowledgement for limitation sequence

If the user confirms the echoed limitation sequence, the processing continues as described in subclause "Collect limitation Information".

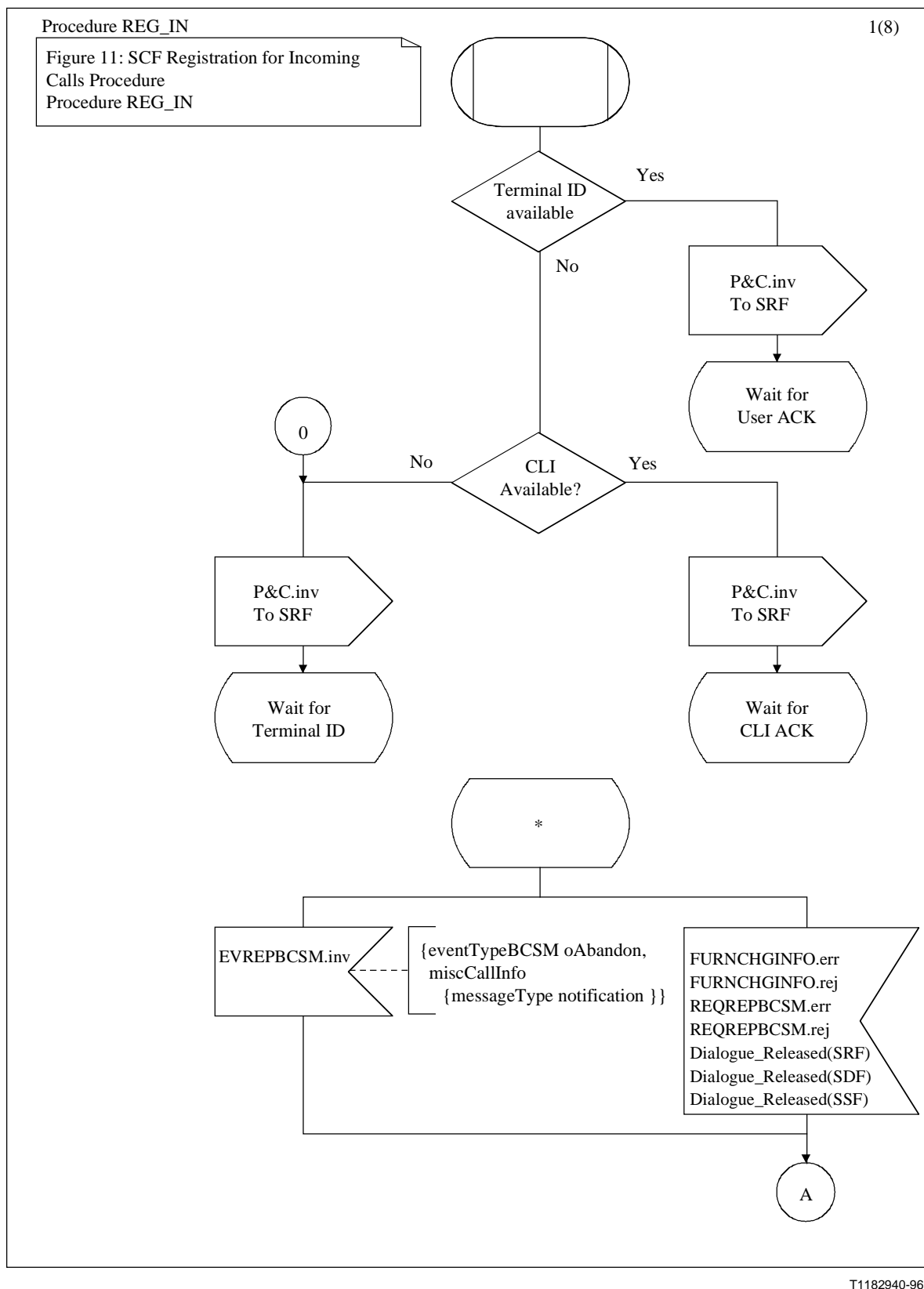
If the user dials "esc" the same treatment described in subclause "CLI available" is applied.

If the user does not confirm the echoed terminal ID, the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request and the number of retries counter is incremented. If the maximum number of retries is not exceeded, a new attempt is allowed. The SCF formulates and sends a P&C operation to the SRF in order to prompt the user to retry the registration procedure and collect again the limitation sequence and returns to the "Wait for limitation sequence" state. If the number of retries is exceeded, the processing continues as described in subclause "Maximum number of retries exceeded".

Update of UPT current location

When the registration for incoming calls is allowed, the SCF formulates and sends a ModifyEntry operation to the SDF home to check if the given number does not violate input restrictions and if so, to update the UPT user current location with the given information. The SCF formulates and sends a P&C operation in order to inform the user that his request has been processed. The SCF waits for the SDFh answer (state "Wait for Update Confirmation"). The following events move the SDF out of this state:

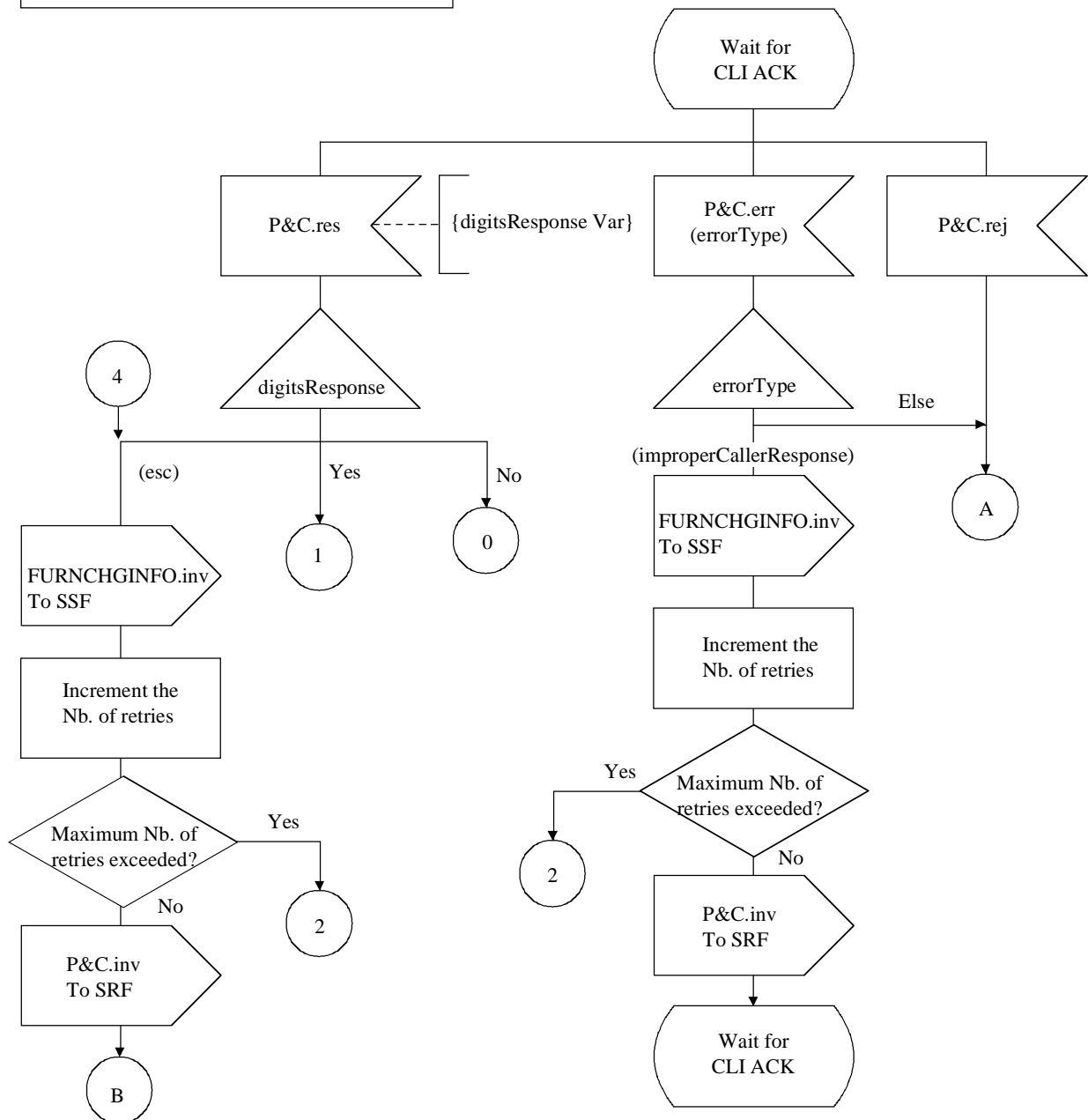
- the ModifyEntry operation is successful: the registration is accepted. The Counter2 is reset and the SCF formulates and sends FURNCHGINFO to the SSF. Afterwards the SCF sends a P&C operation in order to prompt the UPT user that the registration has been successfully executed and ask to terminate or request another procedure (this is done by the FRI procedure);
- an error has occurred for the ModifyEntry operation (MODIFY.err):
 - if the error is "Insufficient Access Rights", the registration is denied and the processing continues as described in subclause "Unsuccessful registration";
 - if the error is "Busy", the ModifyEntry operation is sent again to the same SDF after a time-out (timer) and the "Update of UPT current location" subclause is repeated. Note that Figure 11 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for any other error or when COUNTER2 is exceeded the procedure is terminated with Exit = NOK.
- the ModifyEntry operation is rejected (MODIFY.rej) the procedure is terminated with Exit = NOK.



T1182940-96

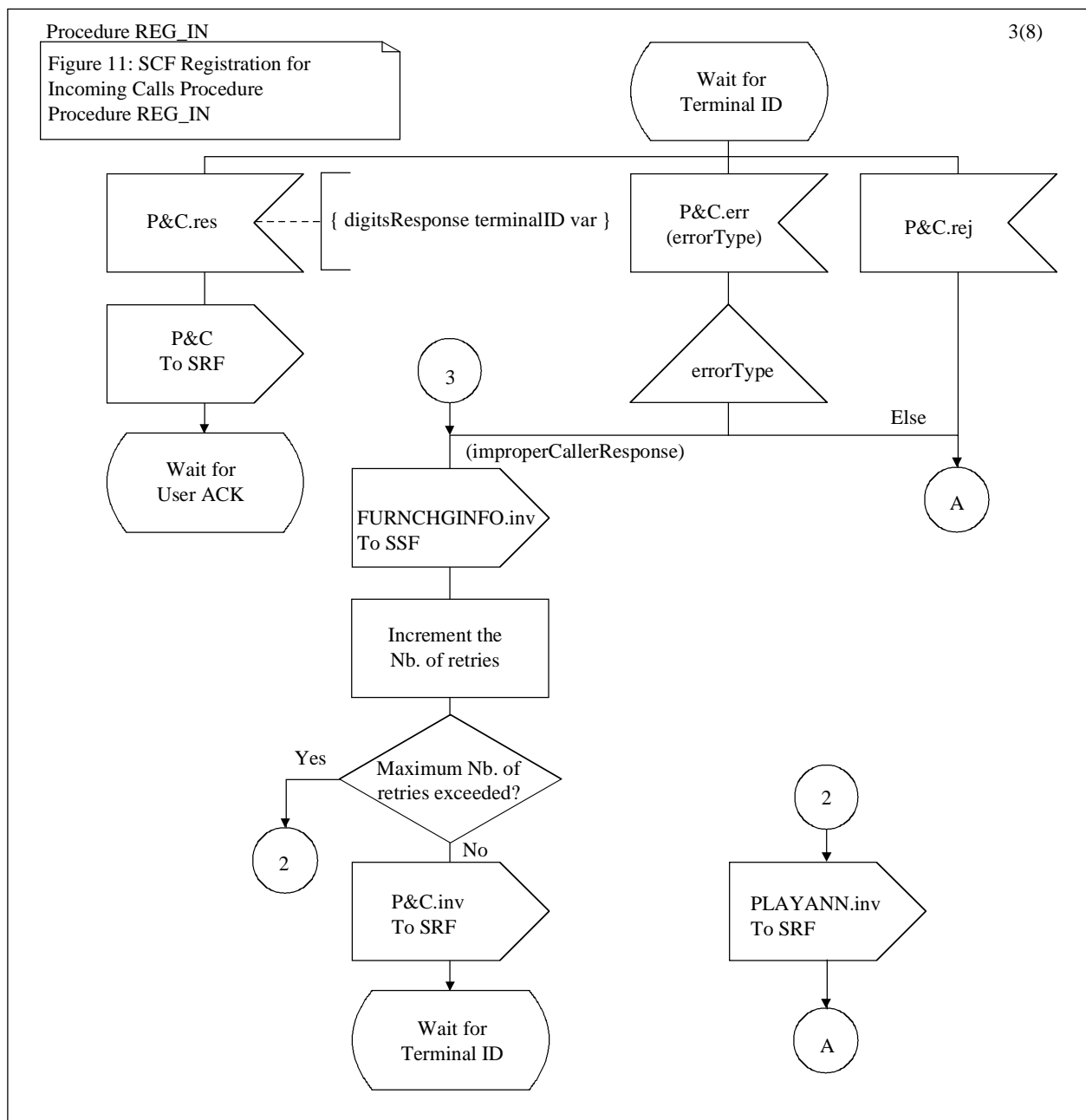
Figure 11/Q.1551 (sheet 1 of 8) – SCF Registration for Incoming Calls procedure

Figure 11: SCF Registration for Incoming Calls Procedure
Procedure REG_IN



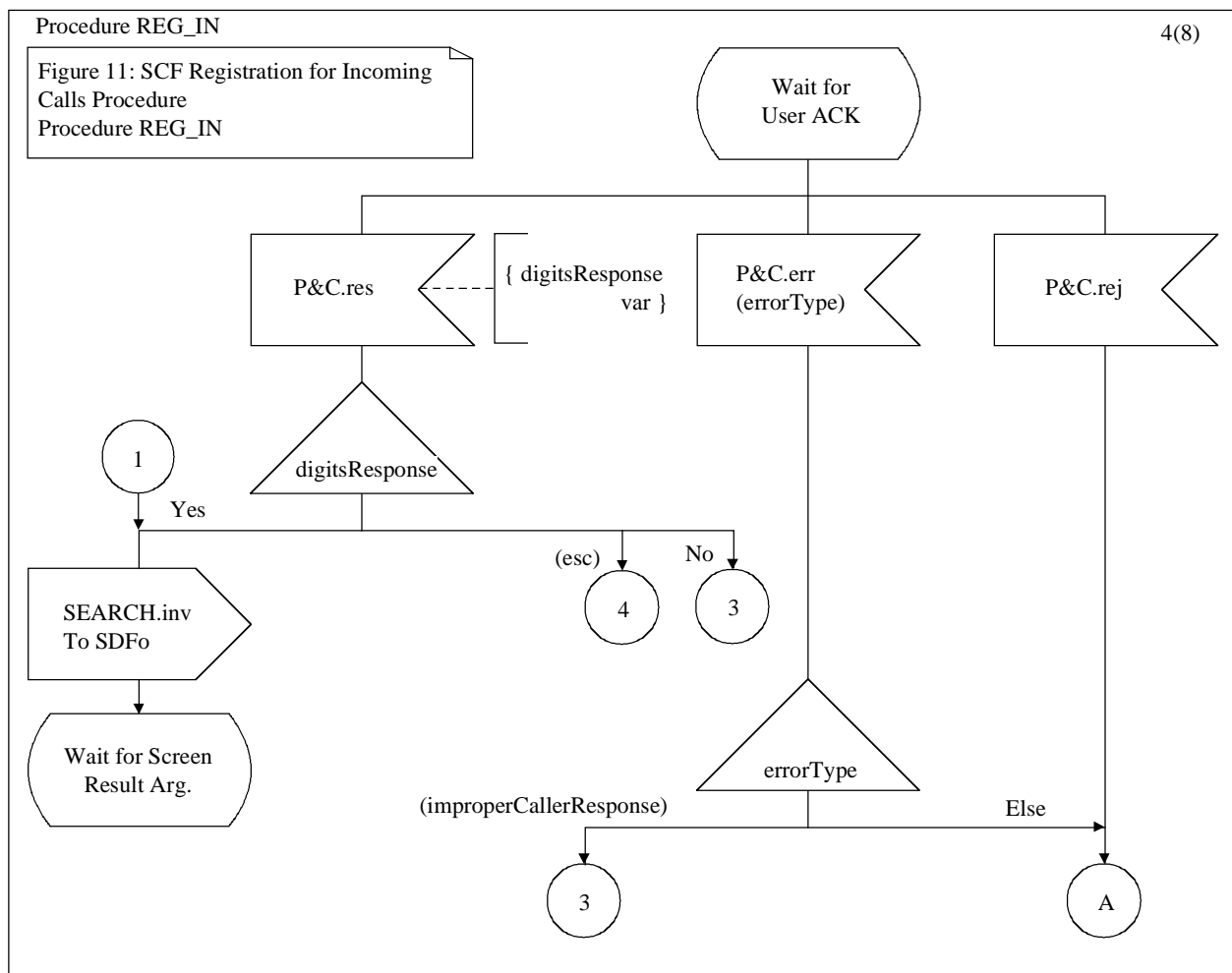
T1182950-96

Figure 11/Q.1551 (sheet 2 of 8) – SCF Registration for Incoming Calls procedure



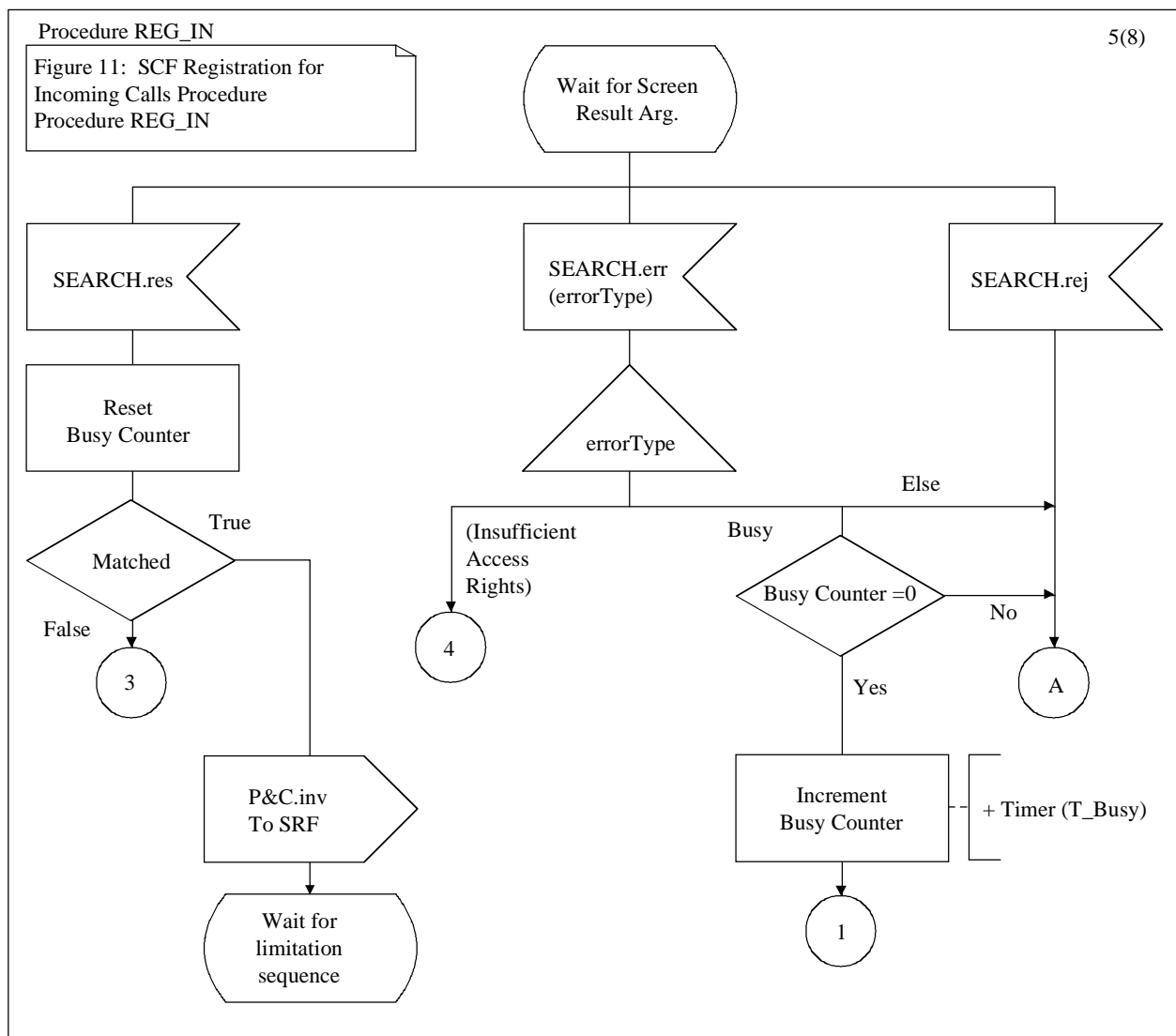
T1182960-96

Figure 11/Q.1551 (sheet 3 of 8) – SCF Registration for Incoming Calls procedure



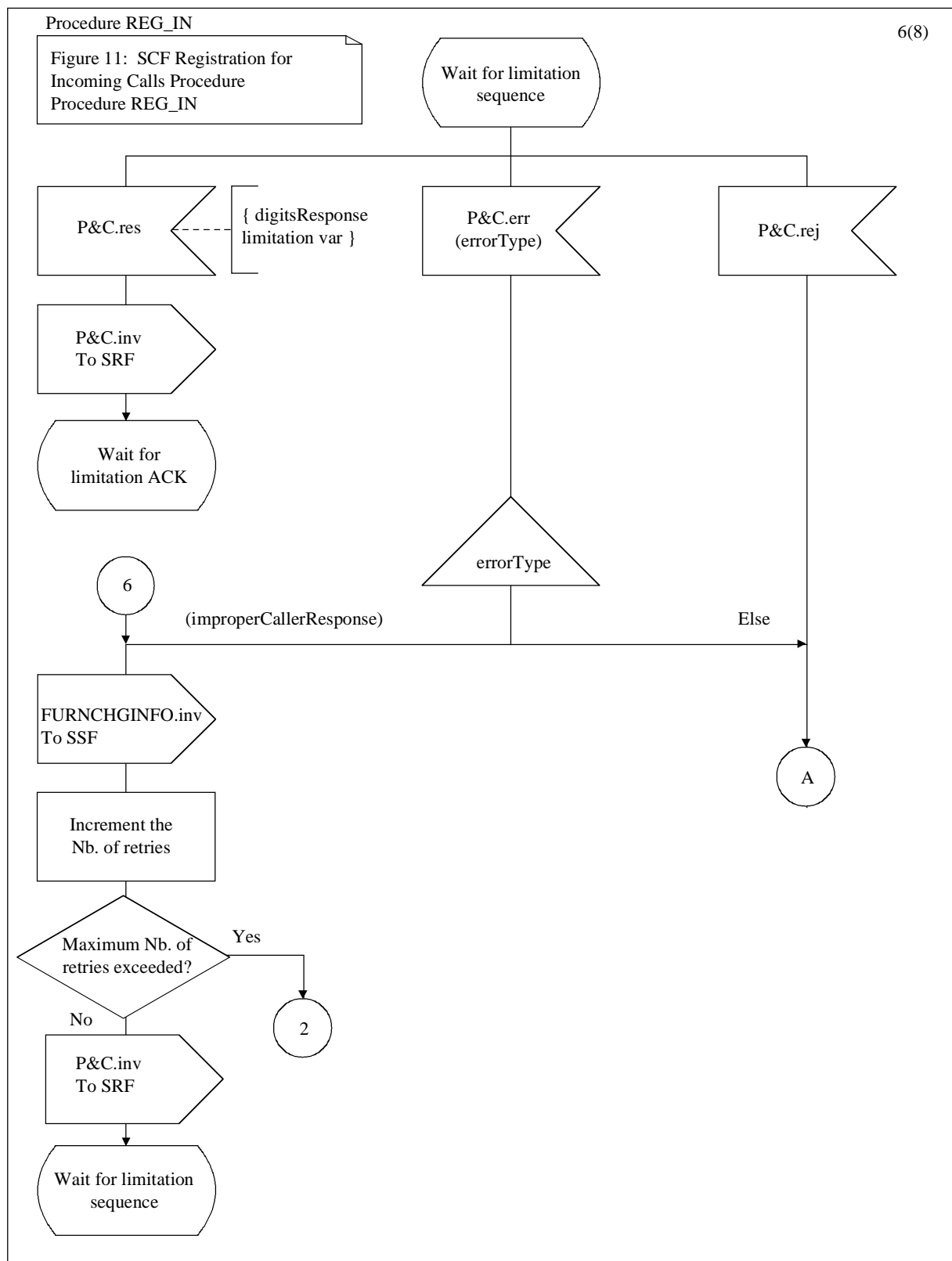
T1172970-95

Figure 11/Q.1551 (sheet 4 of 8) – SCF Registration for Incoming Calls procedure



T1182980-96

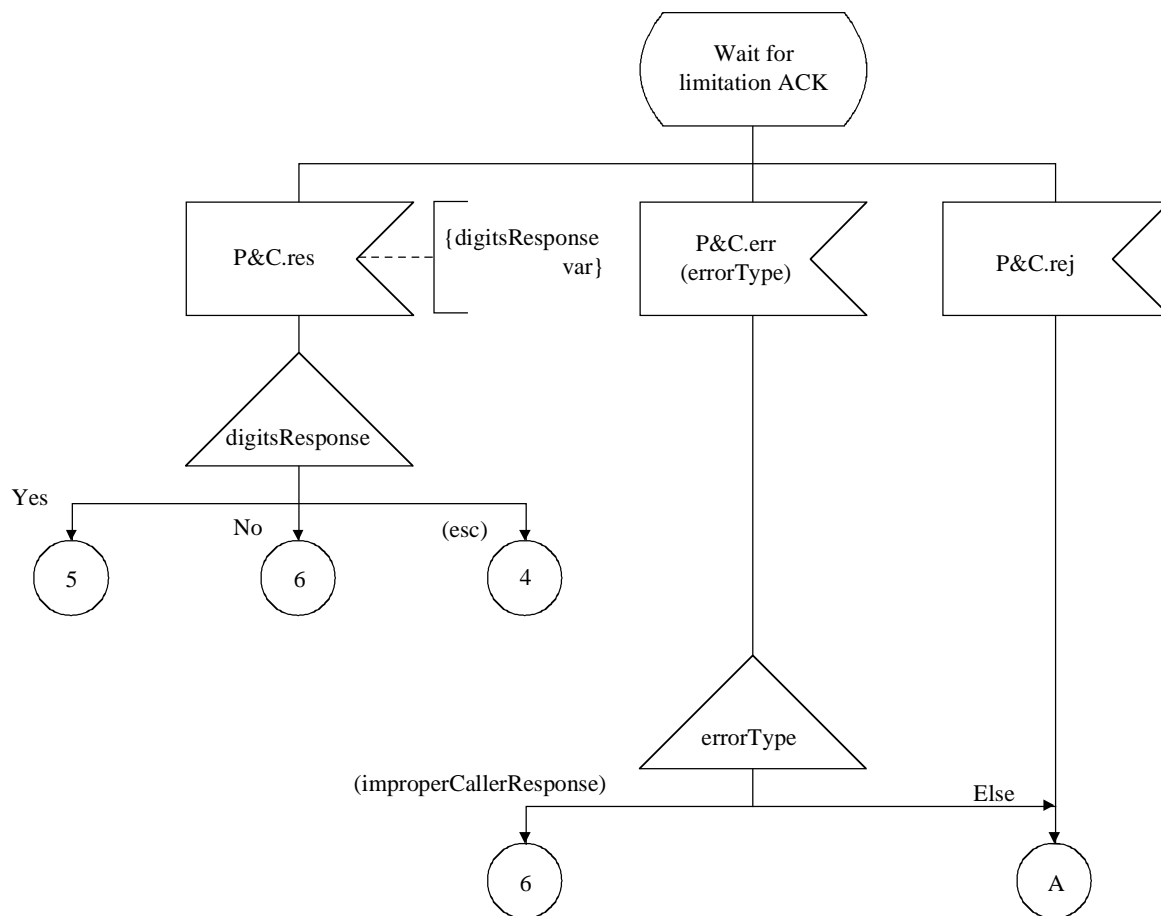
Figure 11/Q.1551 (sheet 5 of 8) – SCF Registration for Incoming Calls procedure



T1182990-96

Figure 11/Q.1551 (sheet 6 of 8) – SCF Registration for Incoming Calls procedure

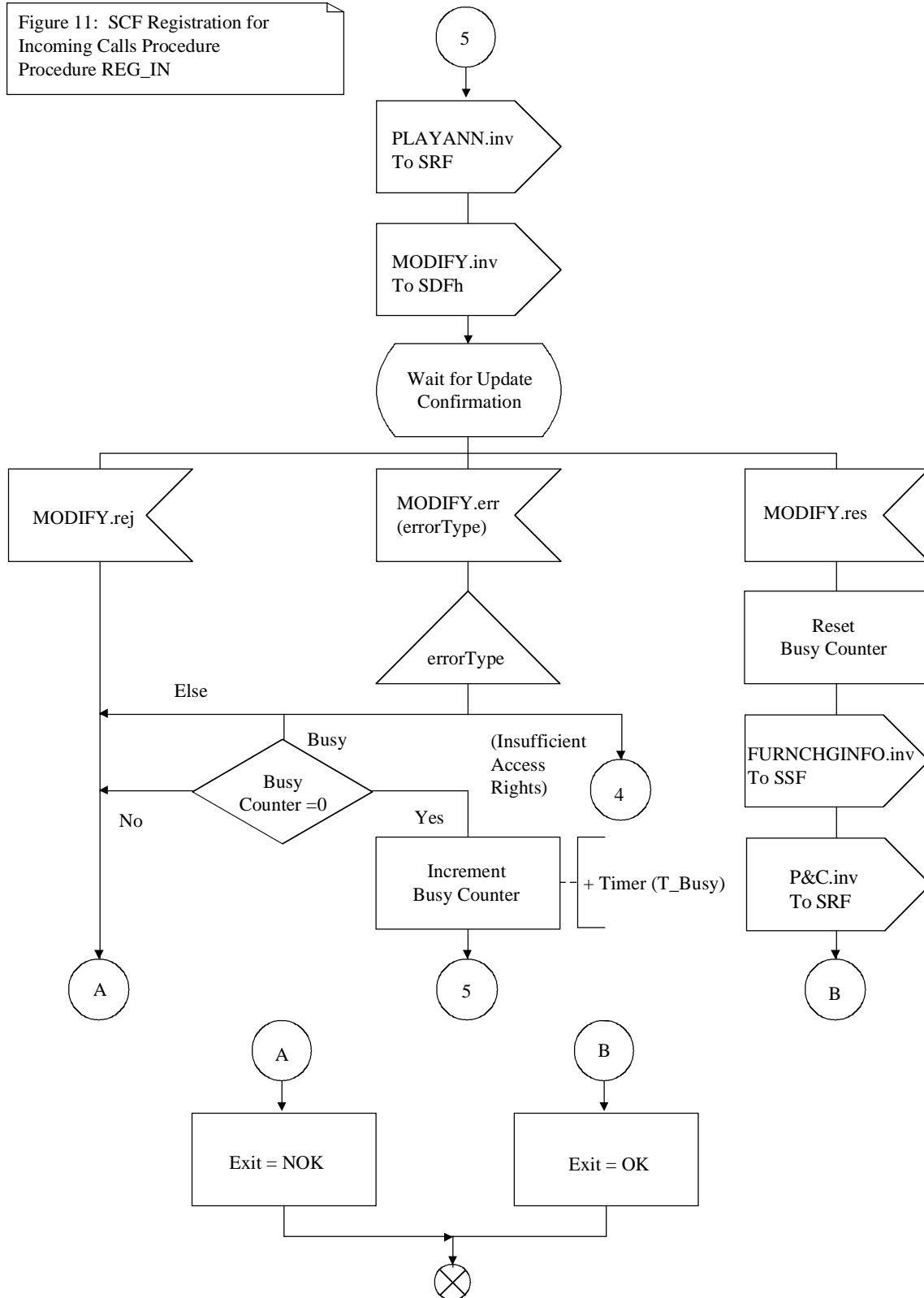
Figure 11: SCF Registration for Incoming Calls Procedure
Procedure REG_IN



T1183000-96

Figure 11/Q.1551 (sheet 7 of 8) – SCF Registration for Incoming Calls procedure

Figure 11: SCF Registration for Incoming Calls Procedure
Procedure REG_IN



T1183010-96

Figure 11/Q.1551 (sheet 8 of 8) – SCF Registration for Incoming Calls procedure

6.3.2 Deregistration for incoming calls

6.3.2.1 General

The Deregistration for Incoming Calls procedure is used when the UPT user explicitly deregisters.

The IA and FRI procedures must have been successfully completed before this procedure.

6.3.2.2 Detailed procedure

Figure 12 shows the Deregistration for Incoming Calls (DEREG_IN) procedure.

The DEREG_IN procedure is invoked by the UPT_SLP process when the UPT user answers with the Incall Deregistration Code during the FRI procedure.

Retrieve of registration information

As the retrieval of registration information is optional, two cases may occur:

- Option 1:** if the data retrieval is necessary for obtaining the current registration information, the SCF formulates and sends a Search operation to the home SDF in order to retrieve the terminal address where the user is currently registered for incoming calls. After that it waits for the SDF answer (state "Wait for Retrieve res.");
- Option 2:** otherwise the procedure continues as described in subclause "User Ack. for deregistration".

The following events move the SCF out of "Wait for Retrieve res." state:

- an error has occurred for the Search operation (SEARCH.err):
 - if the error is "Busy", the ModifyEntry operation may be sent again to the same SDF after a time-out (timer) and the "Retrieve of registration information" subclause is repeated. Note that Figure 12 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - if the error is "Insufficient Access Rights", the procedure is terminated as described in subclause "Unsuccessful deregistration";
 - for any other error or when COUNTER2 is exceeded the procedure is terminated with Exit = NOK.
- the Search operation is rejected (SEARCH.rej) the procedure is terminated with Exit = NOK;
- a response to the Search operation has been received. If a previous registration was done by the user (terminal ID available), the procedure continues as described in the subclause "User Ack. for deregistration". Otherwise the procedure is terminated as described in subclause "Unsuccessful deregistration".

User Ack. for deregistration

The SCF formulates and sends a P&C operation to SRF. The SRF receives and reacts to P&C in two different ways according to options described below:

- Option 1:** if the data retrieval of registration information has occurred, the SRF plays to the user to appeal for his acknowledgement of the deregistration from the terminal where the user is currently registered for incoming calls and waits for the answer (state "Wait for user dereg. Ack.");
- Option 2:** otherwise the SRF plays to the user for his acknowledgement of the deregistration without disclosing the terminal identity from where the user is currently registered for incoming calls and waits for the answer (state "Wait for user dereg. Ack.").

The following events move the SCF out of "Wait for user dereg. ack." state:

- the user has abandoned the procedure: the SCF receives an EVREPBCSM indicating the user release. The procedure is terminated with Exit = NOK. This is included in the "State *" of the SDL diagrams;
- an error has occurred for the P&C operation (P&C.err):
 - if the error is "ImproperCallerResponse", the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries is incremented and, if not exceeded, the prompt is repeated to the user that is allowed to make another attempt. If the number of retries is exceeded the processing continues as described in subclause "Maximum number of retries exceeded";
 - else for any other error the procedure is terminated with Exit = NOK.
- the P&C operation is rejected (P&C.rej): the procedure is terminated with Exit = NOK;
- the user answers "yes" to the prompt: the processing continues as described in subclause "Set to default registration address";
- the user answers "no" or "esc" to the prompt the procedure is terminated as described in subclause "Unsuccessful deregistration".

Set to default registration address

If the user accepts the deregistration, the SCF formulates and sends a P&C operation to SRF in order to inform the user that his request has been processed. The SCF formulates and sends an ModifyEntry operation to the SDFh to set the UPT user current location to the default value. The SCF waits for the SDFh answer (state "Wait for Update Confirmation").

The following events move the SDF out of this state:

- the ModifyEntry operation is successful: the deregistration is accepted. The SCF formulates and sends FURNCHGINFO to the SSF. Afterwards the SCF sends a P&C operation in order to prompt the UPT user that the deregistration has been successfully executed and ask to terminate or request another procedure (this is done by the FRI procedure);
- an error has occurred for the ModifyEntry operation (MODIFY.err):
 - if the error is "Insufficient Access Rights", the deregistration is denied and the processing continues as described in subclause "Denied update";
 - if the error is "Busy", the ModifyEntry operation may be sent again to the same SDF after a time-out (timer) and the "Set to default registration address" subclause is repeated. Note that Figure 12 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for any other error or when COUNTER2 is exceeded, the procedure is terminated with Exit = NOK.
- the ModifyEntry operation is rejected (MODIFY.rej) the procedure is terminated with Exit = NOK.

Denied update

The SCF formulates and sends FURNCHGINFO to request the SSF to send a record for all user requests and the number of retries counter is incremented. If the maximum number of retries is not exceeded, the SCF formulates and sends a P&C operation in order to inform the user that the request cannot be handled and to terminate or request another procedure (this is done by the FRI procedure). If the maximum number is exceeded, the processing continues as described in subclause "Maximum number of retries exceeded".

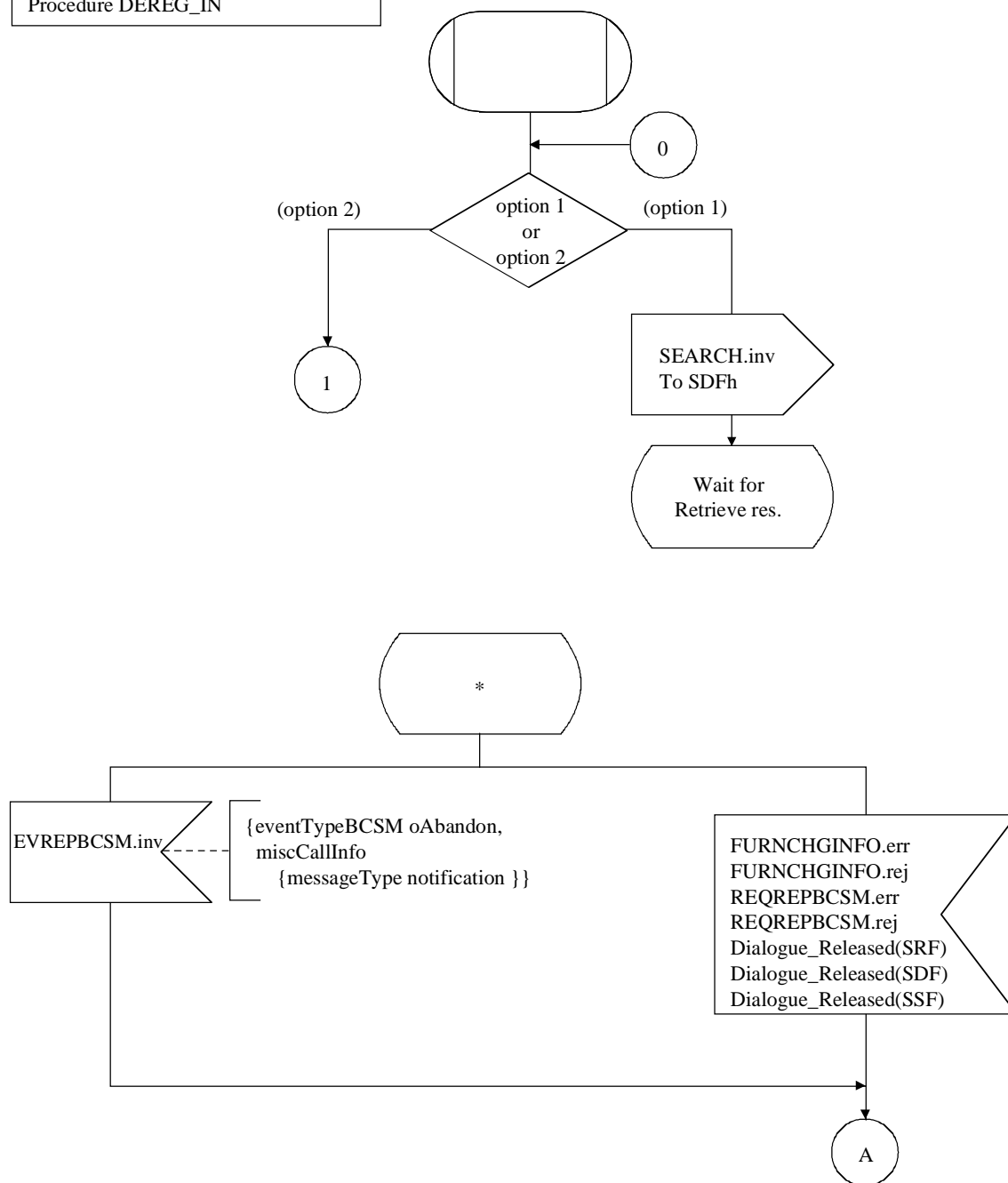
Unsuccessful deregistration

The SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries is incremented and, if not exceeded, the SCF formulates and sends a P&C operation allowing the user to return to the Feature Request Identification (FRI) (follow-on). If the number of retries is exceeded, the processing continues as described in subclause "Maximum number of retries exceeded".

Maximum number of retries exceeded

If the maximum number of retries is exceeded a PLAYANN operation is played to the user for asking him to hang up; the procedure is terminated with Exit = NOK and the calling user is released by the Release procedure.

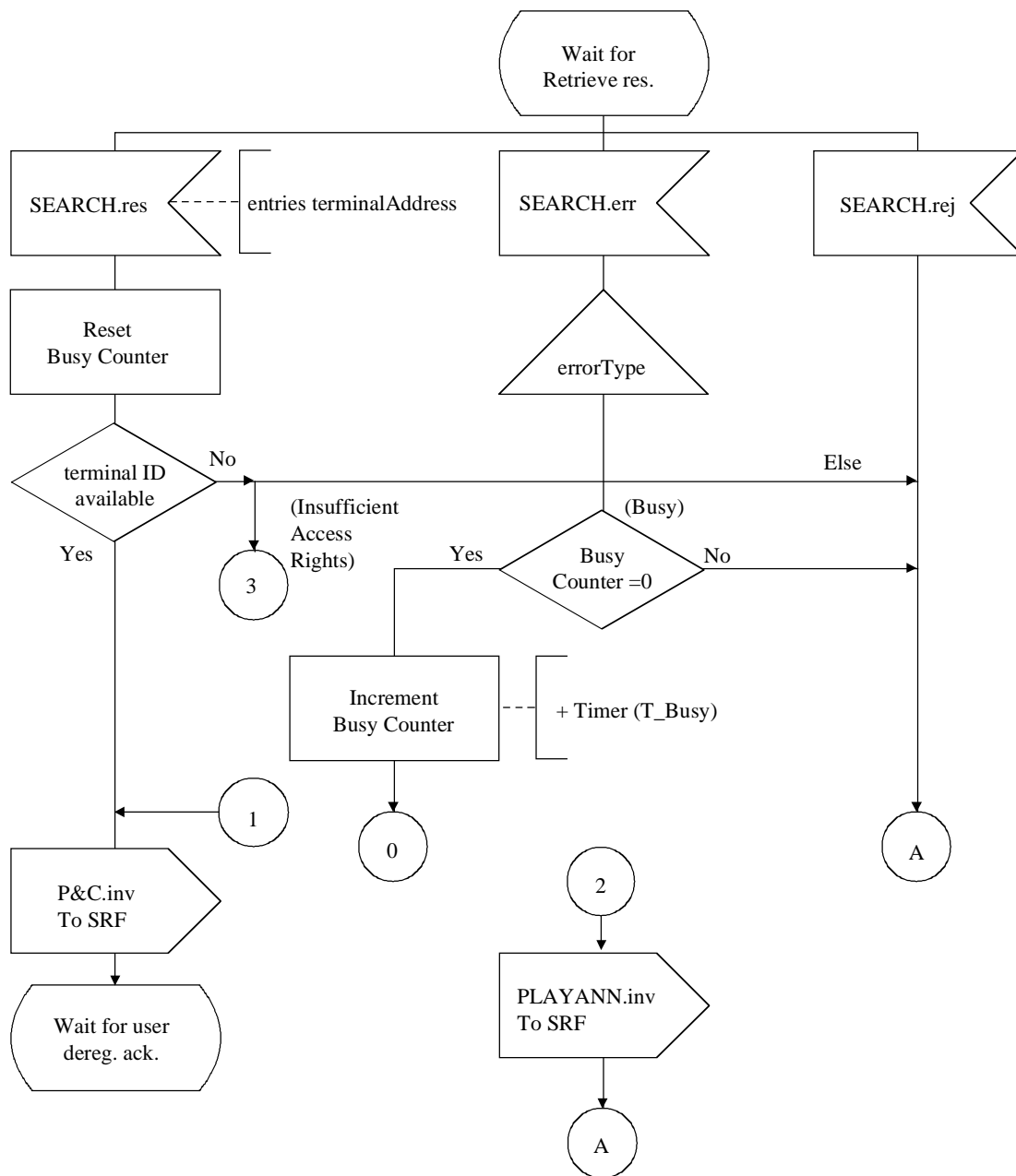
Figure 12 : SCF Deregistration for Incoming Calls Procedure
Procedure DEREG_IN



T1183020-95

Figure 12/Q.1551 (sheet 1 of 4) – SCF Deregistration for Incoming Calls procedure

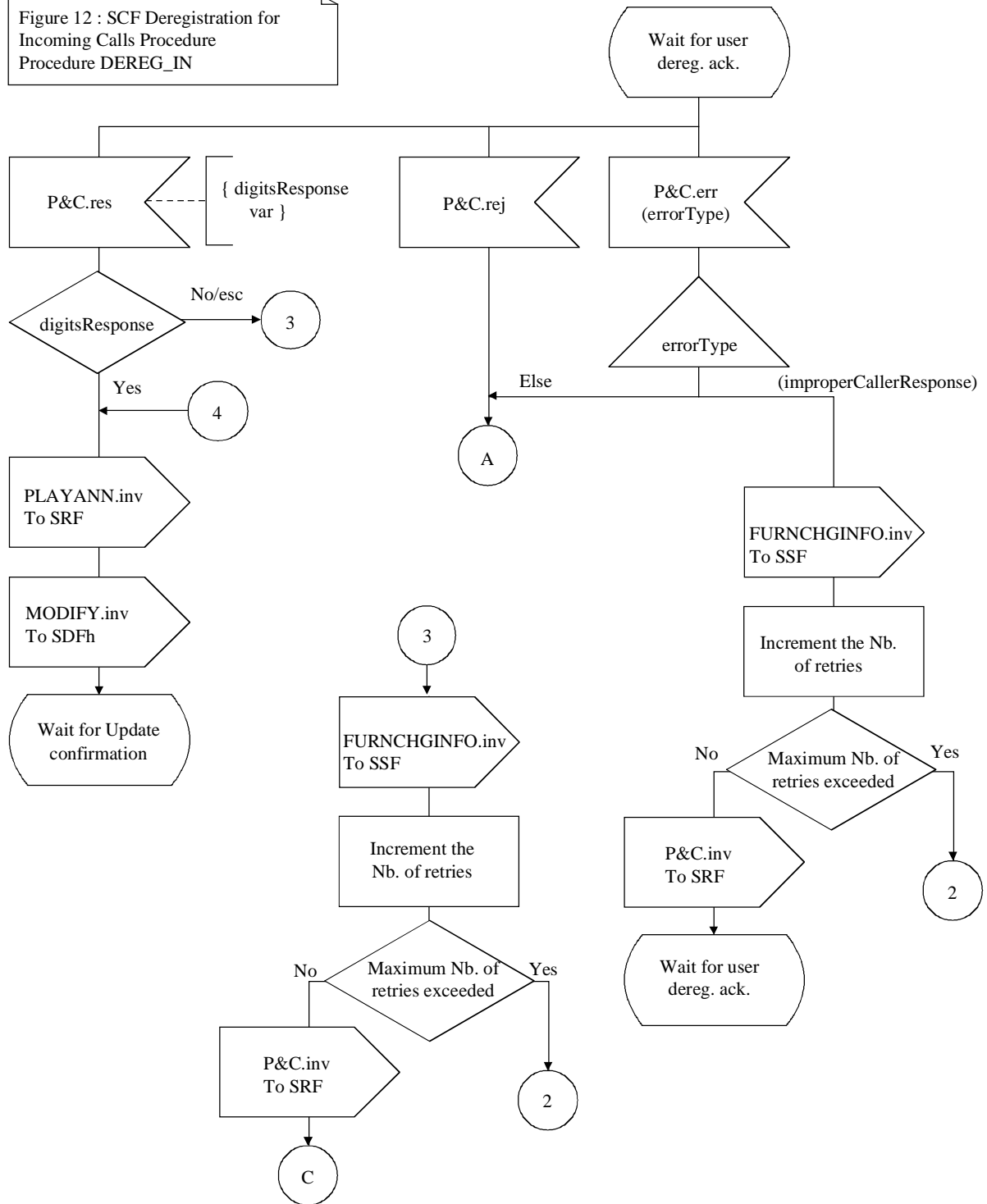
Figure 12 : SCF Deregistration for Incoming Calls Procedure
Procedure DEREG_IN



T1183030-96

Figure 12/Q.1551 (sheet 2 of 4) – SCF Deregistration for Incoming Calls procedure

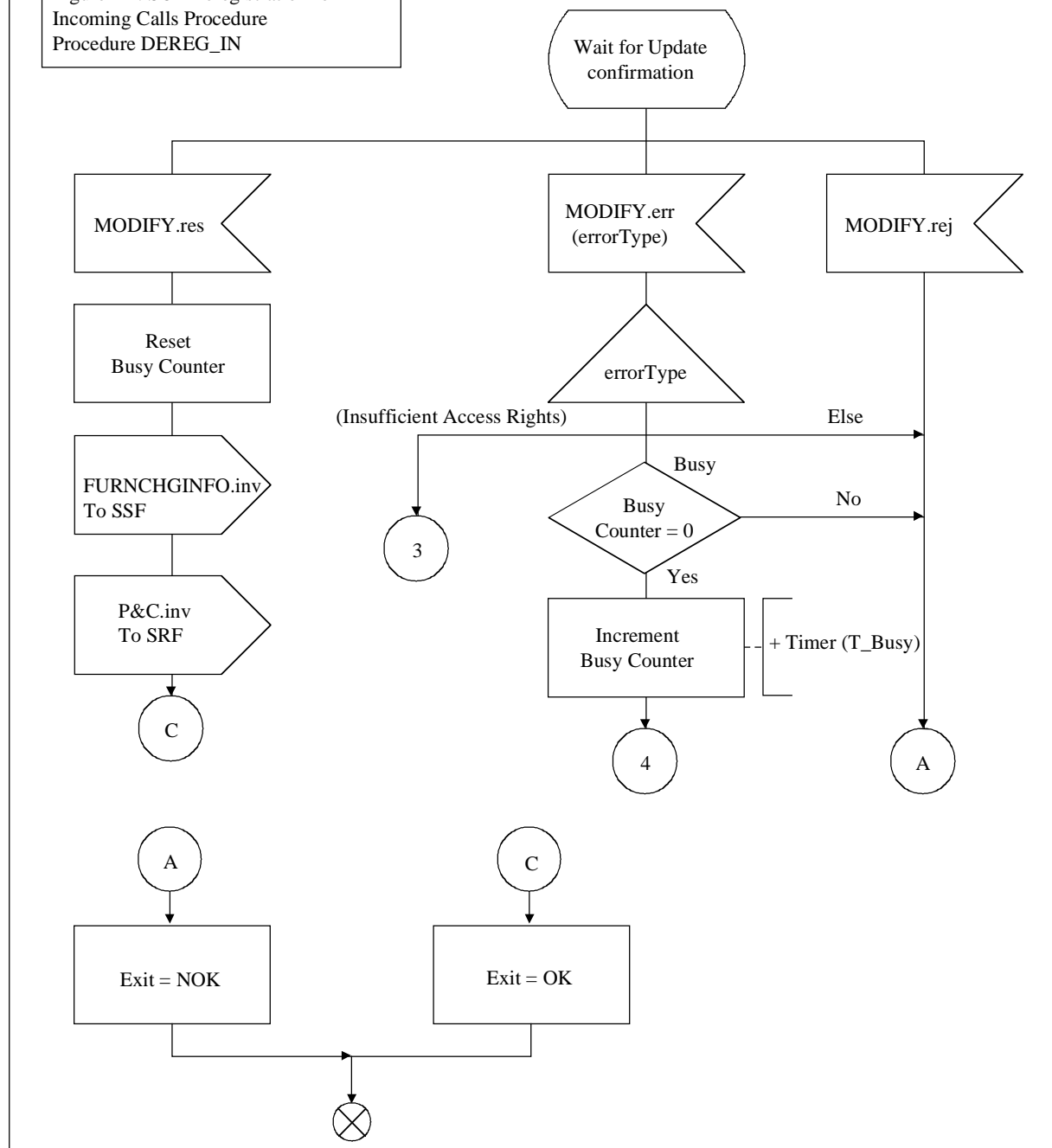
Figure 12 : SCF Deregistration for Incoming Calls Procedure
Procedure DEREG_IN



T1183040-96

Figure 12/Q.1551 (sheet 3 of 4) – SCF Deregistration for Incoming Calls procedure

Figure 12 : SCF Deregistration for Incoming Calls Procedure
Procedure DEREG_IN



T1183050-96

Figure 12/Q.1551 (sheet 4 of 4) – SCF Deregistration for Incoming Calls procedure

6.4 Call handling

UPT call handling procedures are the procedures related to the making and receiving of UPT calls. The description of the procedures assumes that:

- ISUP signalling will be used for the network signalling;
- the limit for the redirection counter is a network-provider option with an upper limit of 5 redirections;

- interaction with ISDN fixed network call diversion services and IN-based UPT call diversion services will follow the procedures described in ISUP/INAP interaction Recommendation Q.1600 [6];
- to enable call forwarding with a UPT number as the routing address (i.e. UPT to UPT call forwarding) to be detected, it will be possible to differentiate between a UPT and non UPT number;
- numbers not recognized as UPT numbers will result in the call being treated as a normal call;
- as the cost of forwarding a call is at the expense of the forwarding user, no service restriction or credit limit checks will be performed on the forwarded-to leg of the call;
- the use of BIND when interacting with the SDFhO is a local matter and is not shown in the call handling procedures.

6.4.1 Outgoing UPT call

This subclause will describe how the UPT user can make a single outgoing UPT call independent of any previous registrations by the UPT user or any other UPT user for incoming and/or outgoing calls to the used terminal access.

6.4.1.1 General

Outgoing calls from a UPT user may be single calls, in which the procedure terminates at the end of the call, or may allow follow-on. The follow-on may be of another call or of another UPT procedure. Follow-on will be offered to the UPT user after the B party disconnects at the end of a conversation or following call set-up failure. The procedures for follow-on are described in more detail in 6.1.3.

The IA and FRI procedures (see 6.2) must have been successfully completed before this procedure.

6.4.1.2 Detailed procedure

The procedure for outgoing call handling for UPT calls is described in Figure 13. The outgoing call procedure is called by the process UPT_SLP as described in 6.1.3.

If Redirection Information is provided in the INITIALDP operation, the Call Forwarding Counter (CFCOUNTER) is set to the same value as the Redirection Counter (see Recommendation Q.763 [7] for a description of this ISUP information element), otherwise the Call Forwarding Counter (CFCOUNTER) is set to zero.

This procedure uses two indicators, Call Forwarding Unconditional (CFUIND) and Charging Announcement (ANNIND). The Call Forwarding Unconditional indicator is used to indicate that Call Forwarding Unconditional service has occurred. The Charging Announcement indicator is used to indicate when the calling party is informed that UPT charging is applicable.

Destination number

If the calling UPT user has already provided a destination number, the Call Handling procedures continue with screening of the UPT users home database; otherwise, the Destination number is requested via invocation of the operation P&C which is sent to the SRF. The result of this operation will be one of the following events:

- dialogue released by IN node (P&C.rej or dialogue_released): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;

- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call set-up procedure as described below in the subclause "User Initiated Release";
- operation error returned (P&C.err): the possible error causes returned in the P&C operation (.err) are described in Recommendation Q.1218 [1]. If the error 'Improper Caller Response' is returned, the user is given another opportunity to enter the destination number, providing the users retry counter (COUNTER1) has not been exceeded. If the retry counter (COUNTER1) has been exceeded the calling user is informed that the maximum number of denied call attempts is reached and is asked to hang up, the SCF will instruct the SRF to send this announcement by invoking the PLAYANN operation. The call handling procedure is then terminated and the RELEASE procedure is activated, see 6.2.3. This also applies for all other error causes which may be returned by the P&C operation;
- successful result: a destination number is provided by the calling UPT user. The procedure continues as described in the subclause Screening of the home database.

Screening of the home database

The SCF checks the calling UPT users home database (SDFhA) for any restrictions which may apply to the dialled number. If the call has already been forwarded but has not been forwarded more times than permitted by the network (the maximum number of times a single call can be diverted is a network-provider option, there is an upper limit of 5, refer to [7] for further information), this check will not be performed. If the call has already been forwarded more times than permitted by the network, this call will be terminated and the calling UPT user will be given an opportunity to enter a different destination number as described below in the subclause "User retry".

To screen the dialled number the SCF will invoke the SEARCH operation. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) is described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the Release procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If there are no restrictions the procedure will continue as described below. If restrictions apply, the calling UPT user will be given an opportunity to enter a different destination number as described below in the subclause "User retry".

A further query may then be made on the UPT users home database (SDFhA) to check the Credit Limit to see if there is any credit available to make the call. This check will not be repeated for subsequent call set-up attempts for the same call (i.e. UPT to UPT Call Forwarding has occurred).

The SCF will invoke another SEARCH operation for this purpose. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) is described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If credit is available to make the call the procedure will continue as described below. If there is no credit available the calling UPT user will be given an opportunity to select another feature, the procedure will continue as described in the subclause "User retry".

For UPT to non-UPT user calls the call handling procedures continue with the Call Set-up procedures.

UPT to UPT call

For UPT to UPT user calls (i.e. the called number is recognized as a UPT number), the originating network database (SDFo) is interrogated to check if agreements between the local service provider and the called UPT users (or the forwarded-to UPT users) home provider exist for establishing outgoing calls. An analysis of the number provided by the calling user should be performed to avoid re-triggering the same SCF (the problem of interaction between IN and other services is to be solved). This interrogation is performed by invoking the SEARCH operation. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the Release procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;

- for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If the result is a match (i.e. agreement exists between the local service provider and the called UPT users home provider for establishing outgoing calls from the current location), then the procedure will continue as described below. In the case of no match (i.e. there is no agreement to establish outgoing calls) the call will be treated as a normal call and routed to another network for completion, for this the procedure continues as described in the subclause "Call set-up".

Following the successful service provider check, the database of the home network of the called or forwarded UPT user (SDFhB) is then interrogated to retrieve the location of the called user. The dialogue is opened with an "empty" Directory BIND operation (i.e. the Credentials parameter will not be present), the outcome of this operation can be one of the following events:

- dialogue released by IN node (Dialogue_released or BIND.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (BIND.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1] regardless of the reason for the error the call handling procedure is terminated and the calling UPT user will be given an opportunity to enter a different destination number or select another feature, the procedure will continue as described in the subclause "User retry";
- successful result (BIND.res): this means that SDFhB accepts the dialogue and the procedure continues as described below.

To retrieve the location of the called user the operation SEARCH is invoked by the SCF, the outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation result (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with EXIT = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDFhB. The result is the routing address(es) applicable at the time the

request is made. If more than one routing address is returned, the SCF will select the address to use based on the following priority:

- 1) the routing address for the Call Forwarding Unconditional service, if this service is active;

NOTE – If this option is selected the Call Forwarding Counter (CFCOUNTER) will be incremented.

- 2) if the Call Forwarding Unconditional service is not active but the registration is still valid, the routing address used will be the registration address;
- 3) if the Call Forwarding Unconditional service is not active but the Variable Routing service is active, the address used will depend on the time or on the calling user;
- 4) default registration address if none of the above criteria apply.

Retrieve default charging reference point

If the call is being forwarded the retrieval of the default charging reference point is not required. The charging for the forwarded leg of the call is a matter for the original called user and is not described here.

To retrieve the default charging reference point the SDFhB is interrogated by the SCF using a SEARCH operation, the outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation result (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDFhB. The procedure then continues as described below.

If split charging [8] is not to be applied, the procedure continues with the "UPT charging notification" subclause below. If split charging is to be applied then the called UPT user's credit limit is checked to determine if there is sufficient credit available to receive the call. The SCF will invoke another SEARCH operation for this purpose. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;

- calling party released: the SCF is notified of the release by the EVREPBSCM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation result (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDFhB. If credit is available to make the call, the procedure will continue as described in subclause "UPT charging notification". If there is no credit available the calling UPT user will be given an opportunity to select another feature, the procedure will continue as described in the subclause "User retry".

UPT charging notification

The calling user is informed that "UPT Charging is Applicable" via invocation of the PLAYANN operation. This notification will not be repeated for subsequent call set-up attempts (i.e. UPT to UPT call forwarding has occurred and the calling user was notified that UPT charging was applicable from a previous attempt to set up this call). Following the instruction for the PLAYANN operation the macro SRF_Disconnect is called, this macro will handle the operation errors and the disconnection of the SRF.

Retrieve supplementary service information

If the original called party was another UPT user but as a result of Call Forwarding Unconditional, the forwarded-to party is not a UPT user this part of the procedure is not required. Instead, the procedure will continue with Call set-up.

The purpose of this part of the procedure is to query the called UPT users (or forwarded-to UPT user if Call Forwarding has occurred) home database (SDFhB) for the status of supplementary services and for those call forwarding services which are active, retrieve the relevant conditional forwarding parameters (e.g. No Reply Condition Timer).

For the case where Call Forwarding Unconditional service has been activated the new forwarded-to UPT Users home database (SDFhB) needs to be queried, it has been assumed that the SDF of the original called party and the new forwarded-to party are not the same. If this is not the case, neither the provider check nor the BIND procedure are required as the same dialogue will be used and the procedure continues with the SEARCH to retrieve supplementary service information described below.

Prior to establishing a dialogue with the new forwarded-to SDFhB, the originating database (SDFo) is searched to check if agreements exist between the local service provider and the new forwarded-to UPT users home provider for establishing outgoing calls. This interrogation is performed by invoking the SEARCH operation. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;

- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If the result is a match (i.e. agreement exists between the local service provider and the forwarded-to UPT users home provider for establishing outgoing calls from the current location), then the procedure will continue as described below. In the case of no match (i.e. there is no agreement to establish outgoing calls) the call will be treated as a normal call and routed to another network for completion, for this the procedure continues as described in the subclause "Call set-up".

Following the successful service provider check, the database of the home network of the new forwarded-to UPT user (SDFhB) is then interrogated to retrieve the supplementary service information. The dialogue is opened with an "empty" Directory BIND operation (i.e. the Credentials parameter will not be present), the outcome of this operation can be one of the following events:

- dialogue released by IN node (Dialogue_released or BIND.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (BIND.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1] regardless of the reason for the error the call handling procedure is terminated and the calling UPT user will be given an opportunity to enter a different destination number or select another feature, the procedure will continue as described in the subclause "User retry";
- successful result (BIND.res): this means that SDFhB accepts the dialogue and the procedure continues as described below.

To retrieve the supplementary service information the SCF will invoke a SEARCH operation for this. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation result (.err) are described in Recommendation Q.1218 [1];

- in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 13 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDFhB. The call forwarding parameters retrieved (if any) are stored for future use. The procedure continues as described below.

Call set-up

The SCF can now instruct the SSF to set up the call, several operations are invoked for this purpose:

- DISCFWDCONN: this instructs the SSF to release the SRF, this operation will not be repeated for subsequent call set-up attempts (i.e. UPT to UPT Call Forwarding has occurred). This operation is invoked from within the macro SRF_Disconnect;
- REQREPBSCSM: this requests the SSF to monitor for a call-related event (e.g. busy, no answer, release) and report back to the SCF when the event has been detected. If a No Reply Condition Timer value was retrieved from the called UPT users home database (SDFhB) when searched for supplementary service information, the value will be provided with this operation;
- FURNCHGINFO: this requests the SSF to generate call record information for the following event;
- APPLYCHG: this operation requests the SSF to report back to the SCF when a charging related event has been detected;
- CONNECT: this instructs the SSF to set up the call (i.e. generate the IAM). If any redirection information was provided in the INITIALDP operation the information will be returned in this operation, the redirection counter may have been updated as a result of further call forwarding. If no redirection information was provided in the INITIALDP operation but call forwarding has occurred, then the redirection information will be constructed by the SCF. Redirection information will not be provided if no call forwarding has occurred. The SSF will determine how to handle this information.

The outcome of this procedure will be one of the following events:

- dialogue released by IN node (Dialogue_released or <operation_name>.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- called Party busy: this state is reported to the SCF by the SSF returning the operation EVREPBCSM. There are three possible outcomes:
 - if call forwarding on busy service is not active, the calling UPT user is informed that the call cannot be connected and is provided with an opportunity to enter another destination number or select another feature. The SCF will first ask the SSF to establish a temporary connection to the SRF by invoking the SRF_Connect procedure (see 6.2.4). The procedure continues as described in the subclause "User retry";

- if call forwarding on busy service is active but the Call Forwarding Counter (CFCOUNTER) has exceeded the network redirection limit (note that the upper limit for this counter is 5), then the calling UPT user is informed that the call cannot be connected and provides the UPT user with an opportunity to enter another destination number or select another feature. The SCF will first ask the SSF to establish a temporary connection to the SRF by invoking the SRF_Connect procedure (see 6.2.4). The procedure continues as described in subclause "User retry";
- if call forwarding on busy service is active, and the Call Forwarding Counter (CFCOUNTER) has not exceeded the network redirection limit, the call can be forwarded. The Call Forwarding Counter (CFCOUNTER) is incremented and the outgoing call procedure restarted.
- called Party no answer: this state is reported to the SCF by the SSF returning the operation EVREPBCSM. There are three possible outcomes:
 - if call forwarding on no reply service is not active, the calling UPT user is informed that the call cannot be connected and is provided with an opportunity to enter another destination number or select another feature. The SCF will first ask the SSF to establish a temporary connection to the SRF by invoking the SRF_Connect procedure (see 6.2.4). The procedure continues as described in subclause "User retry";
 - if call forwarding on no reply service is active but the Call Forwarding Counter (CFCOUNTER) has exceeded the network redirection limit (note that the upper limit for this counter is 5), then the calling UPT user is informed that the call cannot be connected and provides the UPT user with an opportunity to enter another destination number or select another feature. The SCF will first ask the SSF to establish a temporary connection to the SRF by invoking the SRF_Connect procedure (see 6.2.4). The procedure continues as described in the subclause "User retry";
 - if call forwarding on no reply service is active and the Call Forwarding Counter (CFCOUNTER) has not exceeded the network redirection limit, the call can be forwarded. The Call Forwarding Counter (CFCOUNTER) is incremented and the outgoing call procedure restarted.
- route select failure: this state will be reported to the SCF by the SSF returning the operation EVREPBCSM. This indicates that it was not possible to complete the set-up of the call due to either congestion, unsubscribed number or number blocked. The calling UPT user is informed that the call cannot be connected and provides the UPT user with an opportunity to enter another destination number or select another feature. The SCF will first ask the SSF to establish a temporary connection to the SRF by invoking the SRF_Connect procedure (see 6.2.4). The procedure continues as described in subclause "User retry";
- operation error returned (CONNECT.err, REQREPBCSM.err, FURNCHGINFO.err, APPLYCHGRPT.err or DISCFWDCONN.err): the possible error causes returned by these operations are described in Recommendation Q.1218 [1]. Regardless of the reason for the error, the call set-up procedure is terminated and the calling user is then given an opportunity to select another feature, the procedure continues as described in subclause "User retry";
- called party answers: the SCF is notified of this event by an EVREPBCSM operation.

Call release and follow-on

The set-up having been completed, the SCF waits for the release of the call. The SCF is notified of the release by the SSF sending an EVREPBCSM operation. If the B party releases the call and the A party does not release the call immediately, this means that a follow-on call is required.

Before invoking the follow-on procedure, the SCF sends the FURNCHGINFO operation to the SSF, this requests the SSF to update the call record. The SCF then waits for the call record from the SSF, during this period the calling UPT user may choose to release the call, the SCF is notified of this event by an EVREPBCSM operation from the SSF.

The SSF will provide the call record in the APPLYCHGRPT operation. This information is then used to update the Calling Party's home database (SDFhA) by the SCF invoking the MODIFY operation to store the call record in the SDF. It should be noted that it is not possible to directly use the call record to modify the users credit as the SDF is not able to calculate the charge that corresponds to the call record. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or MODIFY.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure as described below in the subclause "User initiated release";
- operation error returned (MODIFY.err): the possible error causes returned in the MODIFY operation (.err) are described in Recommendation Q.1218 [1]:
 - if the error is "Busy", the MODIFY operation can be attempted again, after a time delay. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed but to avoid the loss of the call record an implementation-dependent action could be performed at this stage;
 - for any other error the procedure is terminated. To avoid loss of the call record an implementation-dependent action could be performed at this stage;
 - Successful result (MODIFY.res): the SDF will report the update of the database with MODIFY.res.

In the case of UPT to UPT calls the original called UPT user's home database (SDFhB) will also be updated in the same manner as described above.

To offer the follow-on to the user the SCF will instruct the SSF to reconnect an SRF through the SRF_Connect procedure described in 6.2.4.

Once the connection to the SRF is confirmed the SCF will instruct the SRF to inform the UPT user that another request can be made or terminate, by sending the operation P&C. The follow-on procedure then continues with the feature request identification procedures as described in 6.2.2.

The dialogue to the original SDFhB is released. There may be more than one SDFhB dialogues open, these should also be released.

User retry

Firstly, the SCF will request the SSF to create a call record for the following call event by invoking the operation FURNCHGINFO. The retry counter (COUNTER1) is incremented.

If the retry counter (COUNTER1) has been exceeded the calling user is informed that the maximum number of denied call attempts is reached and is asked to hang up, the SCF will instruct the SRF to send this announcement by invoking the PLAYANN operation. The call handling procedure is then terminated and the Release procedure is activated, see 6.2.3.

If the retry counter (COUNTER1) has not been exceeded the calling user is informed that the request is denied and is requested to either hang up or to make another feature request. The SCF will instruct the SRF to send this announcement by invoking the PLAYANN operation. The procedure then

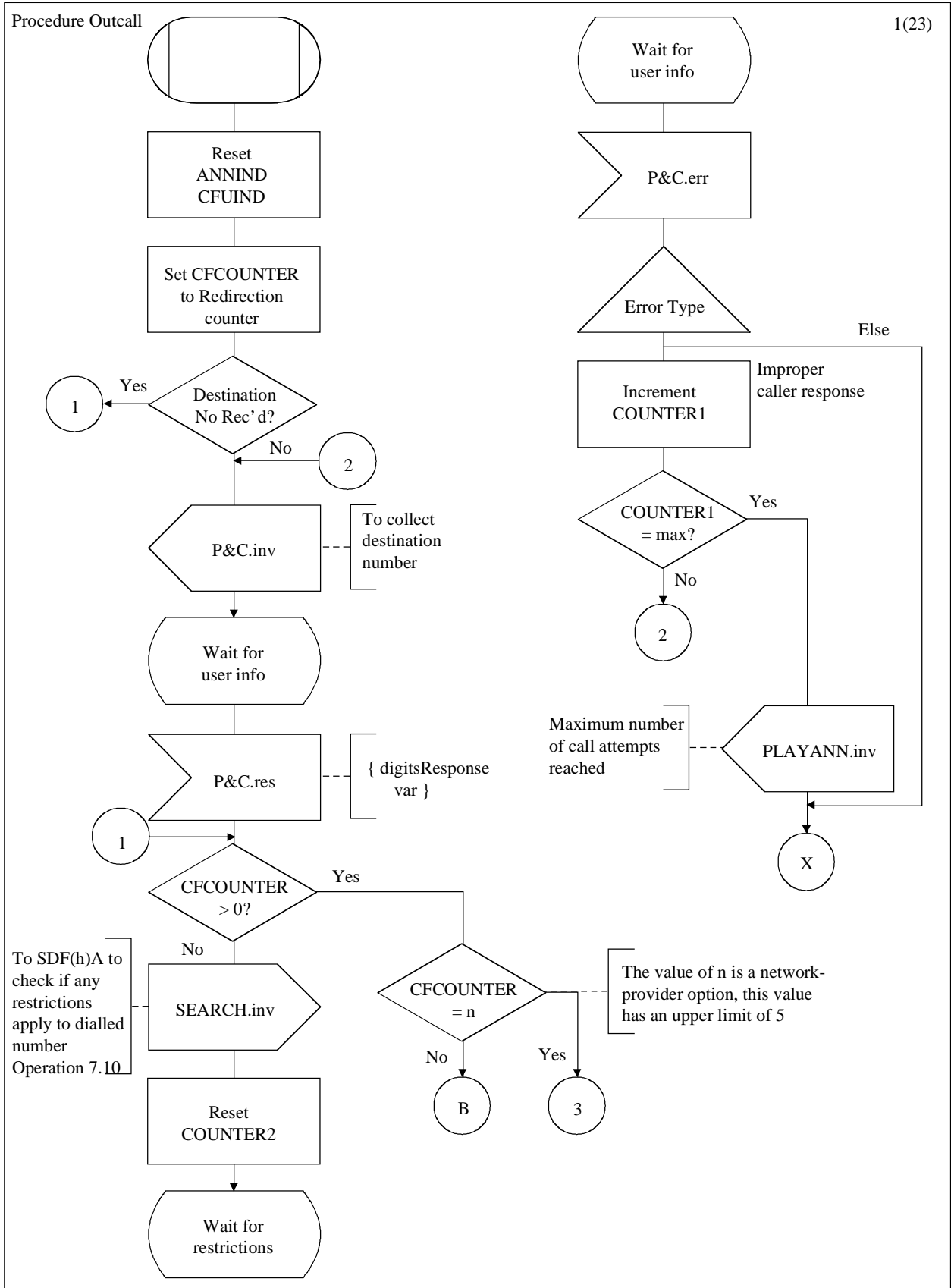
continues with the Feature Request Identification procedure as described in 6.2.2, call handling procedures are terminated.

User initiated release

The SCF, following notification of early release of the call by the calling party, waits for the APPLYCHGRPT operation which returns to the SCF the call record. Once this message is received, the SCF sends a MODIFY operation to SDFhA to store the call record in the SDF. It should be noted that it is not possible to directly use the call record to modify the user credit as the SDF is not able to calculate the charge that corresponds to the call record. The outcome of this operation will be one of the following events:

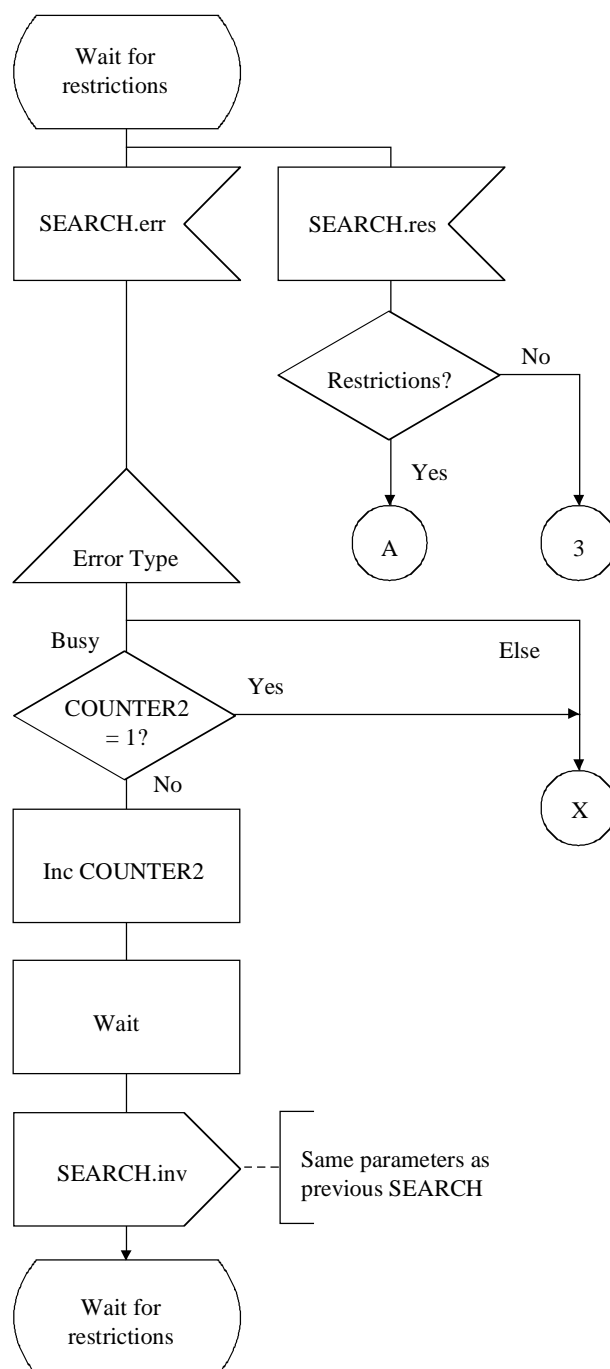
- dialogue released by IN node (Dialogue_released or MODIFY.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- operation error returned (MODIFY.err): the possible error causes returned in the MODIFY operation rasion (.err) are described in Recommendation Q.1218 [1]:
 - if the error is "Busy", the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries is exceeded (COUNTER2) the procedure is terminated. To avoid loss of the call record an implementation-dependent operation could be performed at this stage;
 - for any other error the procedure is terminated. To avoid loss of the call record an implementation-dependent operation could be performed at this stage.
- successful result (MODIFY.res): this confirms that the SDF has been successfully updated.

If the call is a UPT to UPT call the same procedure as described for SDFhA will be performed on the original SDFhB.



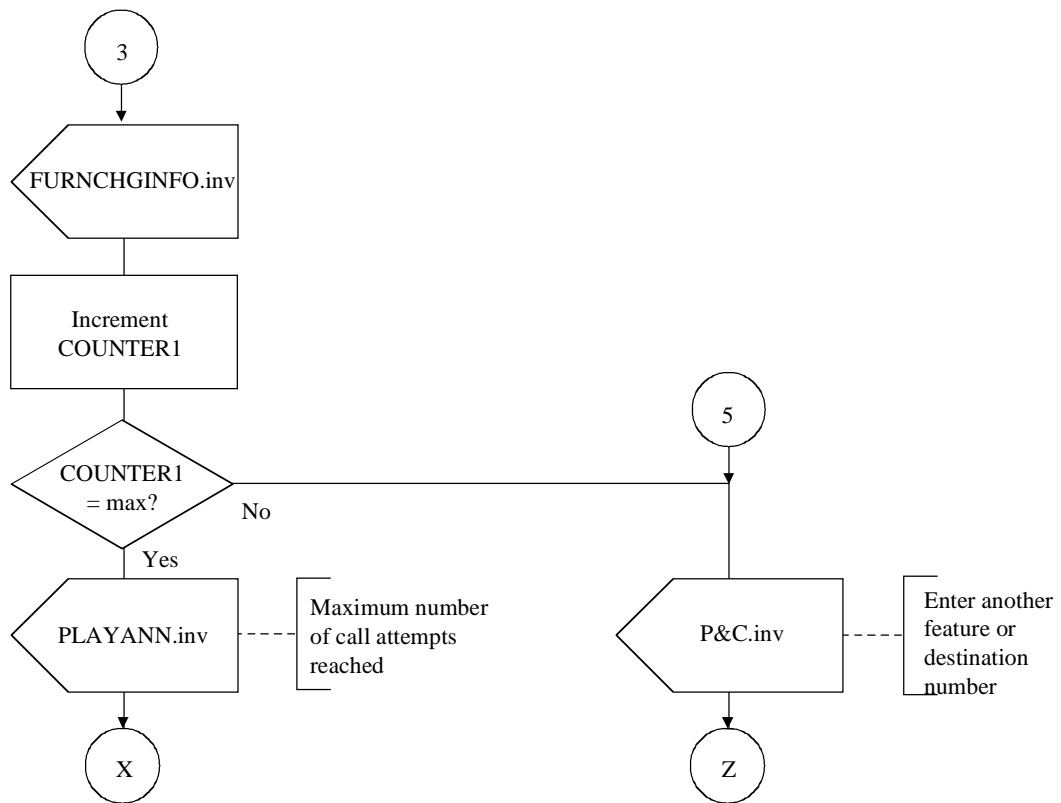
T1183060-96

Figure 13/Q.1551 (sheet 1 of 23) – Outgoing UPT Call procedure



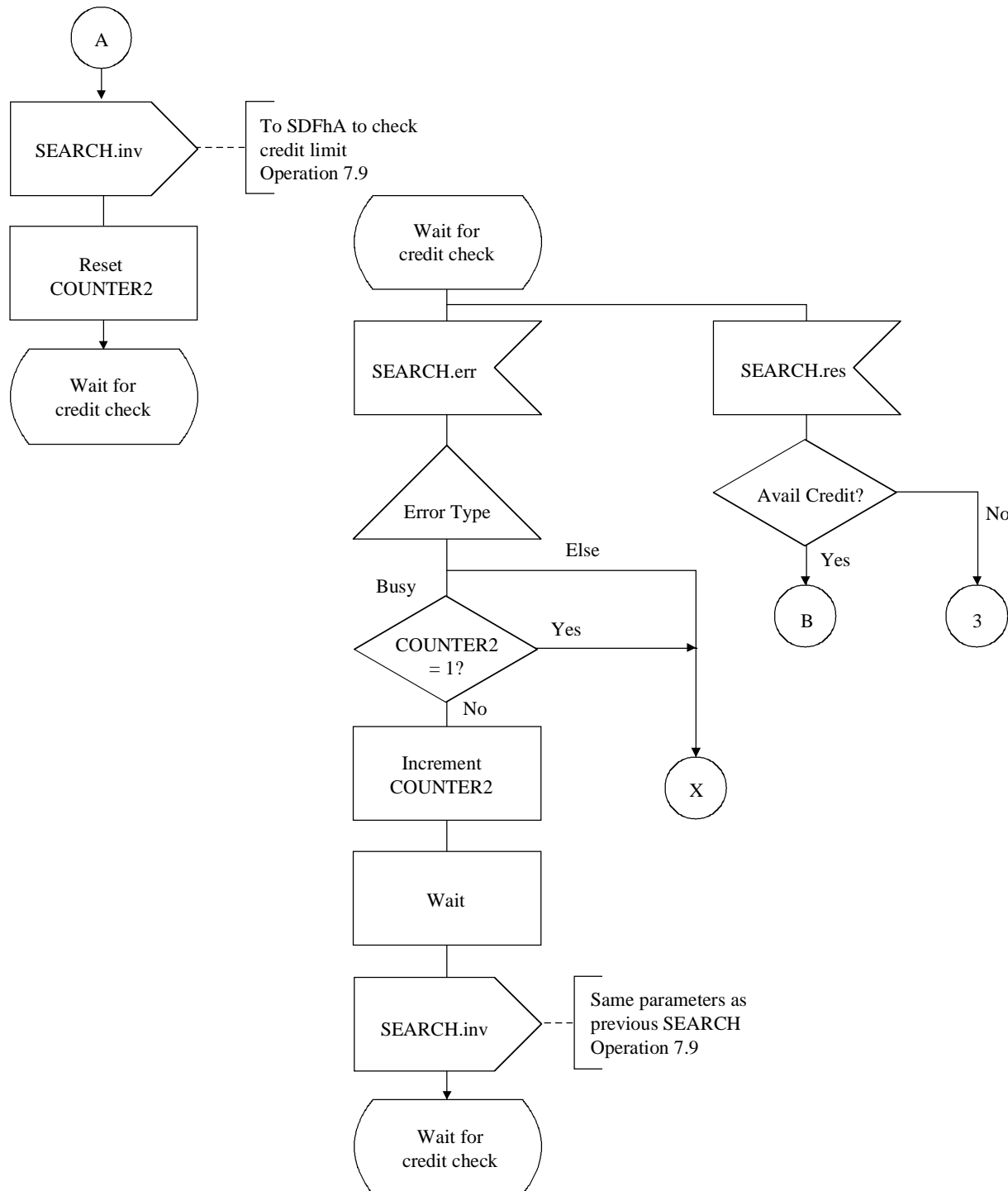
T1183070-96

Figure 13/Q.1551 (sheet 2 of 23) – Outgoing UPT Call procedure



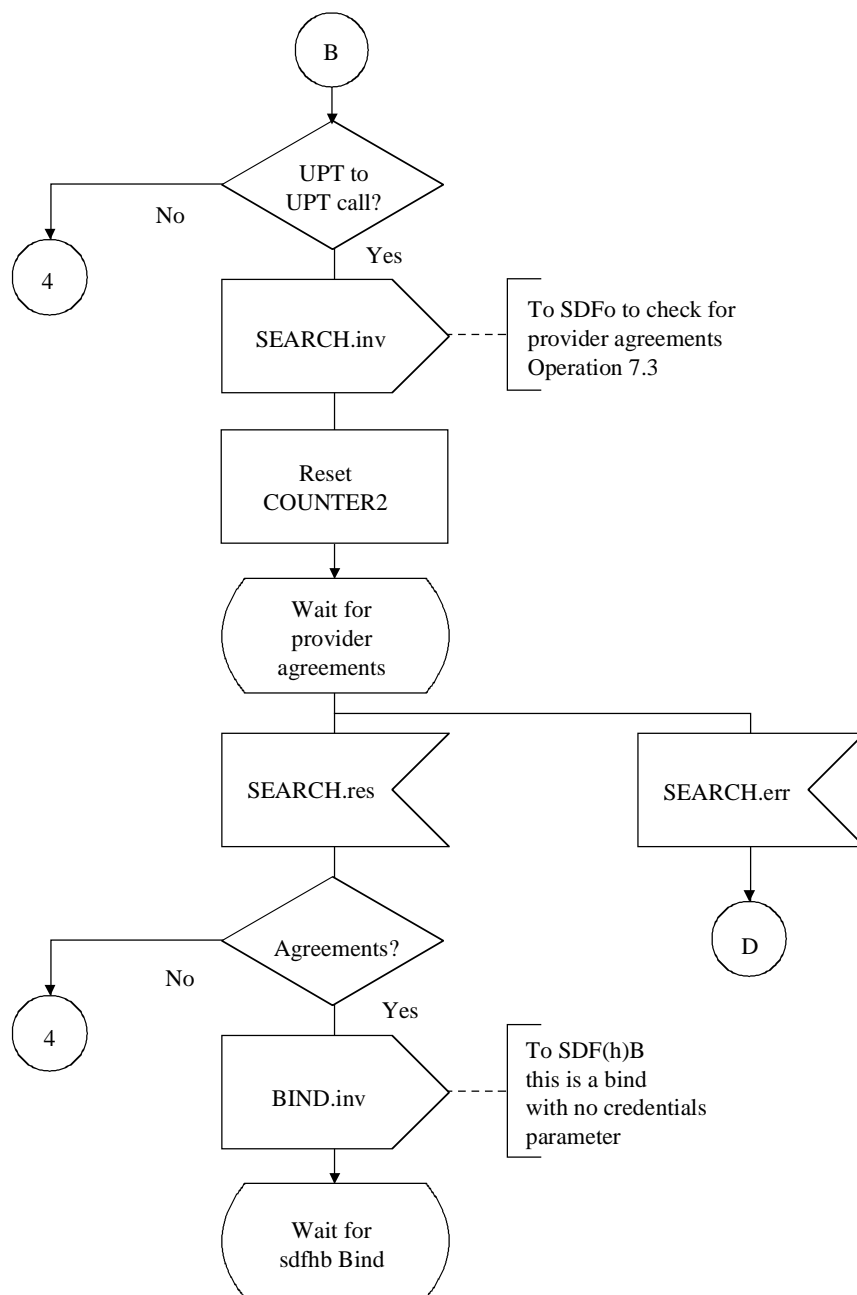
T1183080-96

Figure 13/Q.1551 (sheet 3 of 23) – Outgoing UPT Call procedure



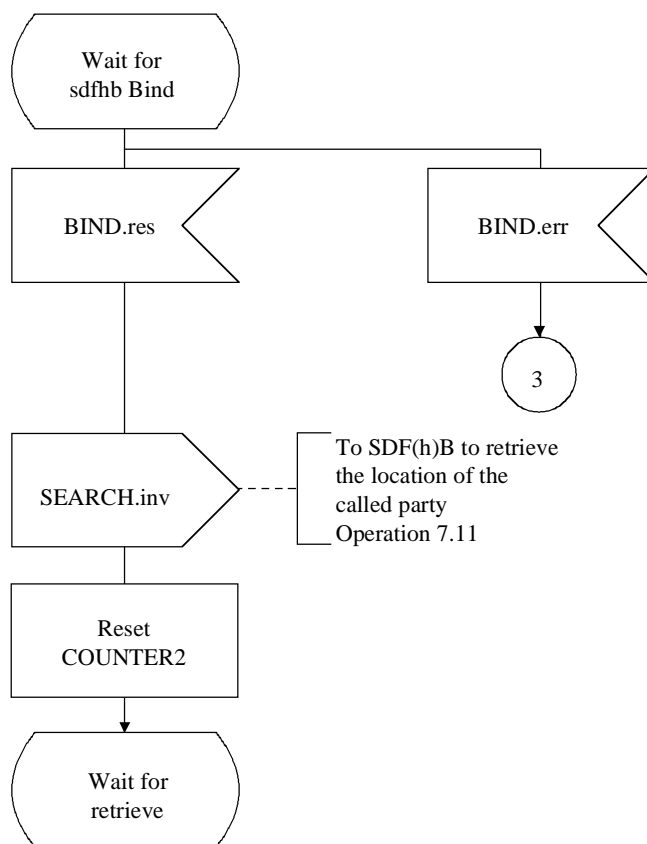
T1183090-96

Figure 13/Q.1551 (sheet 4 of 23) – Outgoing UPT Call procedure



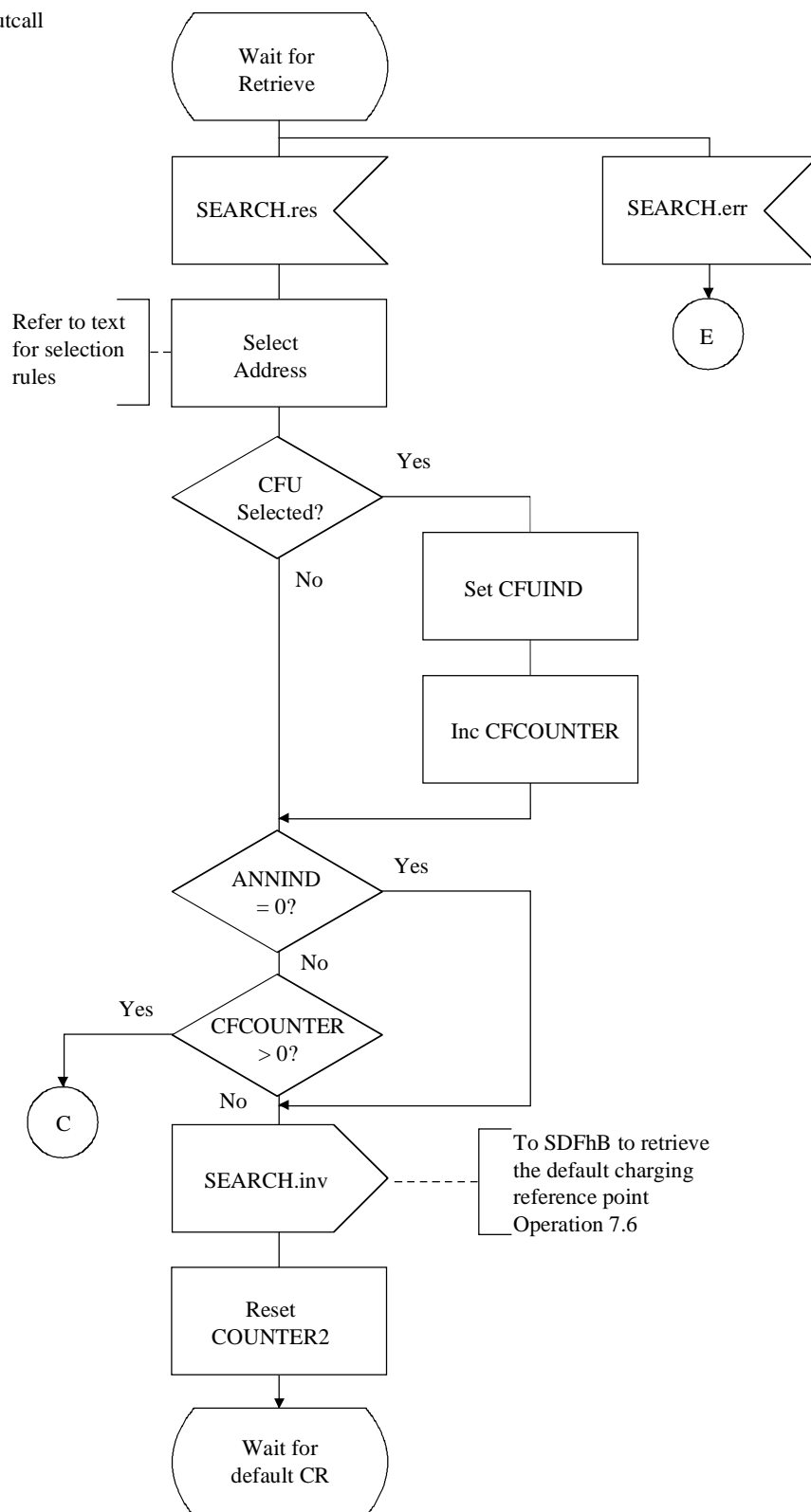
T1183100-96

Figure 13/Q.1551 (sheet 5 of 23) – Outgoing UPT Call procedure



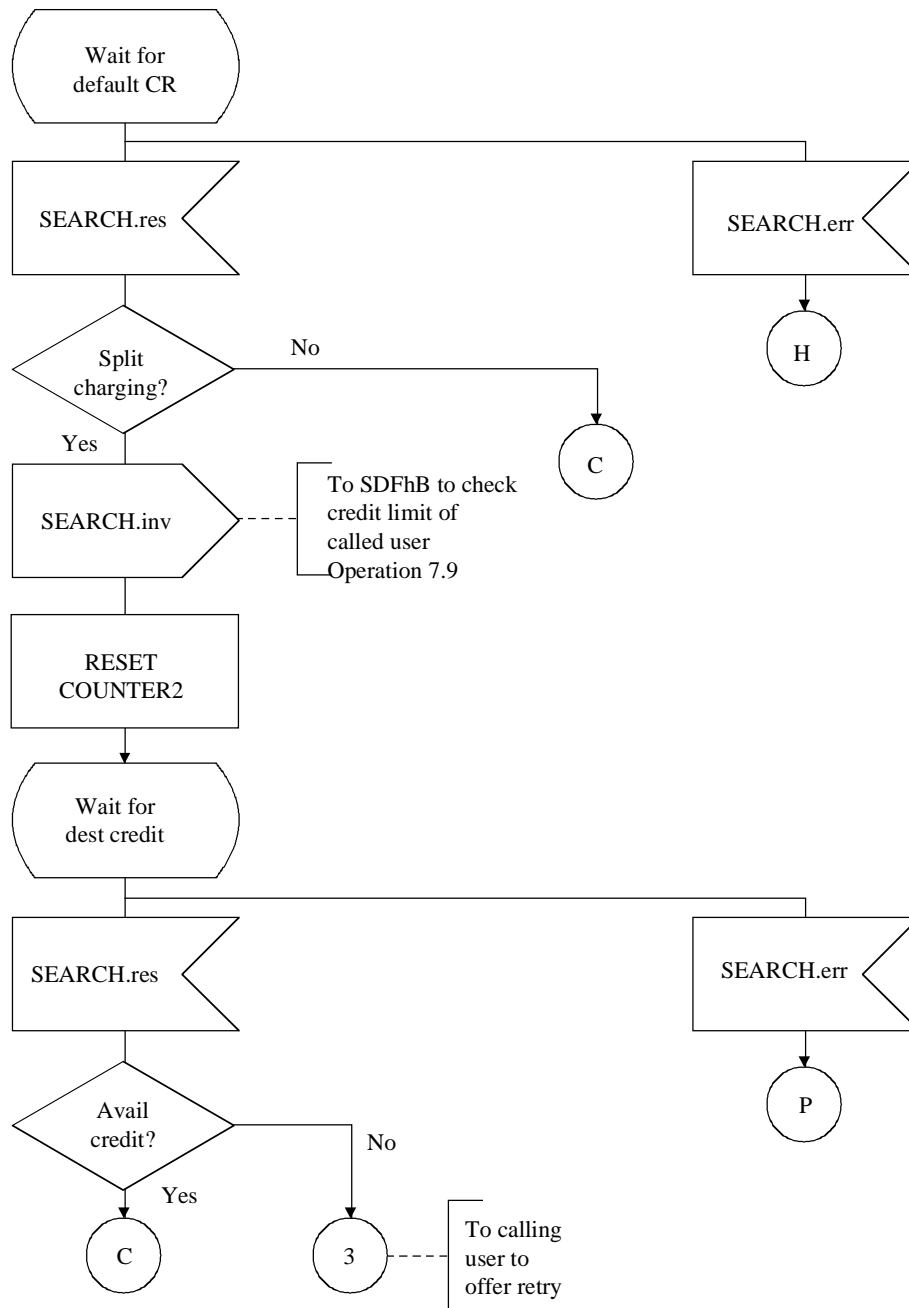
T1183110-96

Figure 13/Q.1551 (sheet 6 of 23) – Outgoing UPT Call procedure



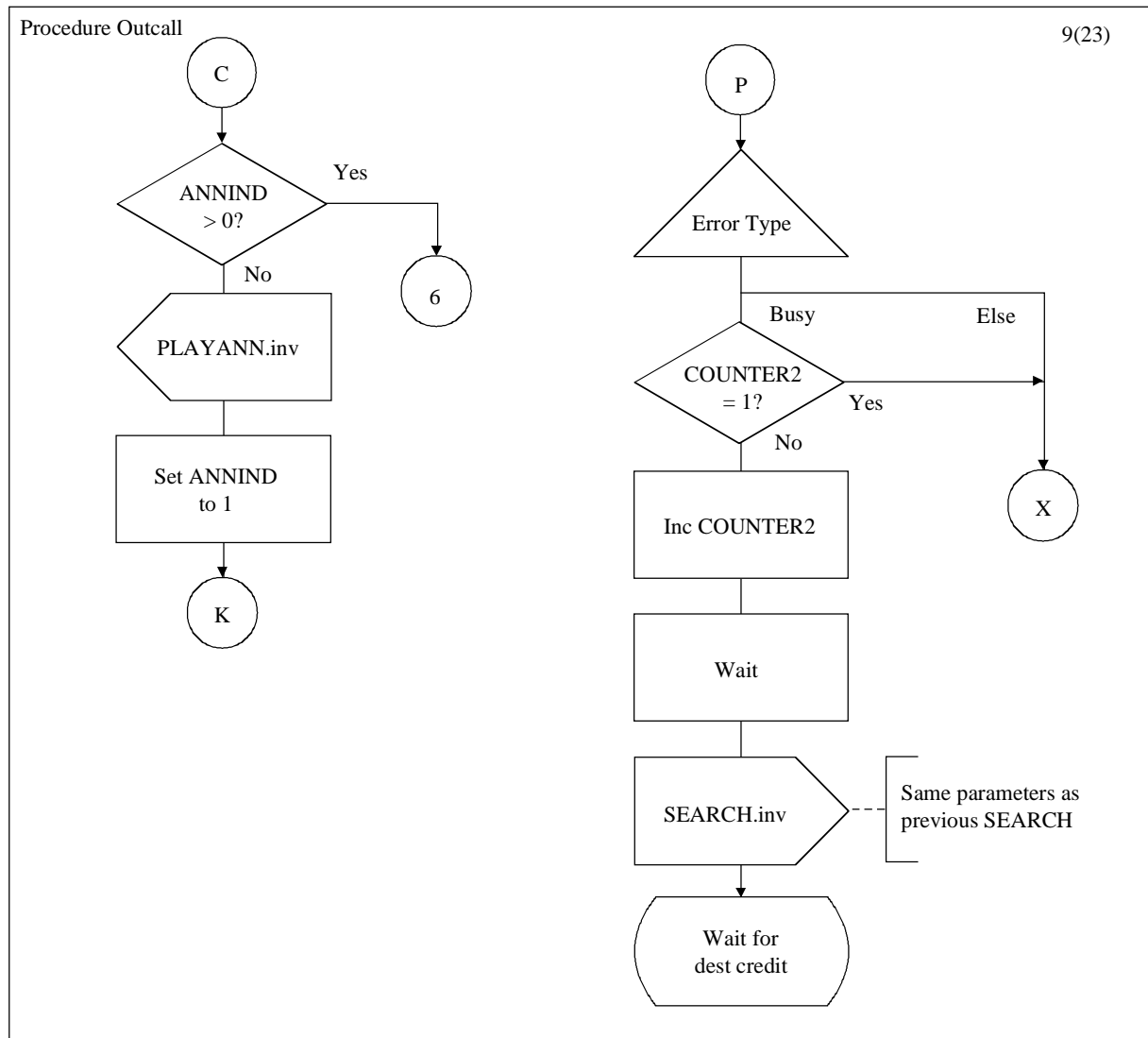
T1183120-96

Figure 13/Q.1551 (sheet 7 of 23) – Outgoing UPT Call procedure



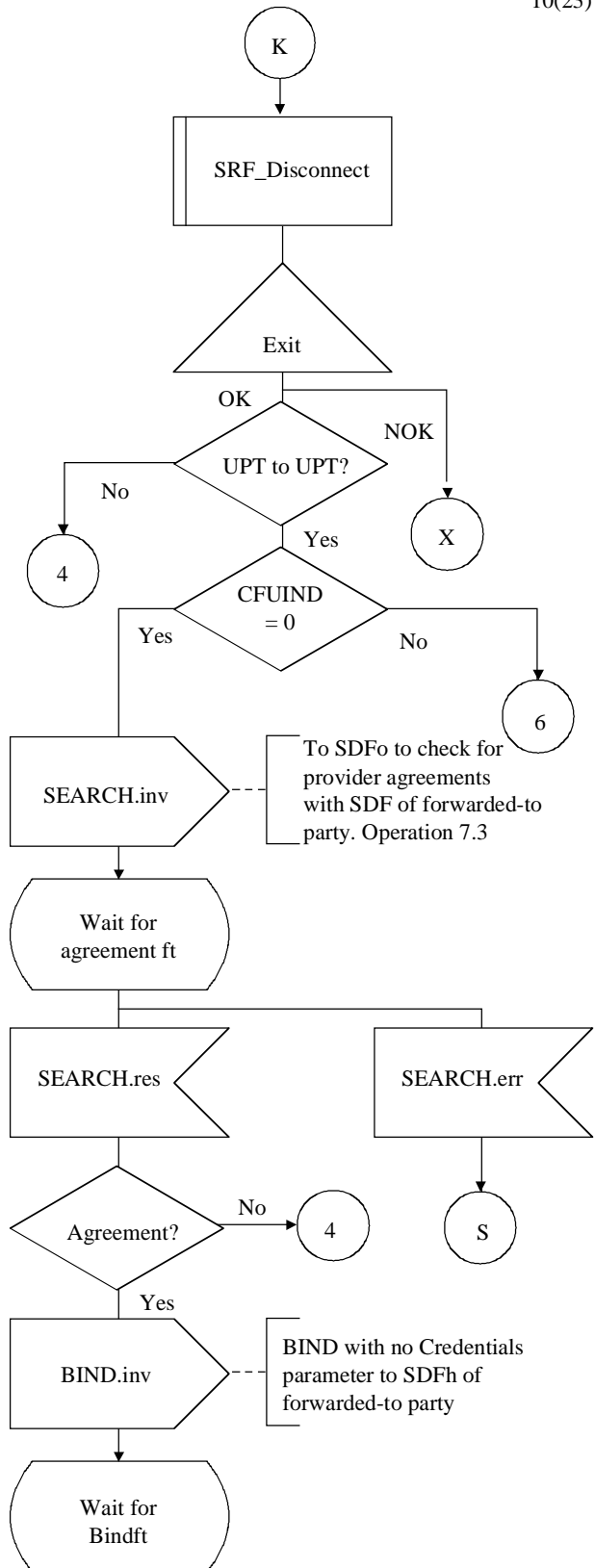
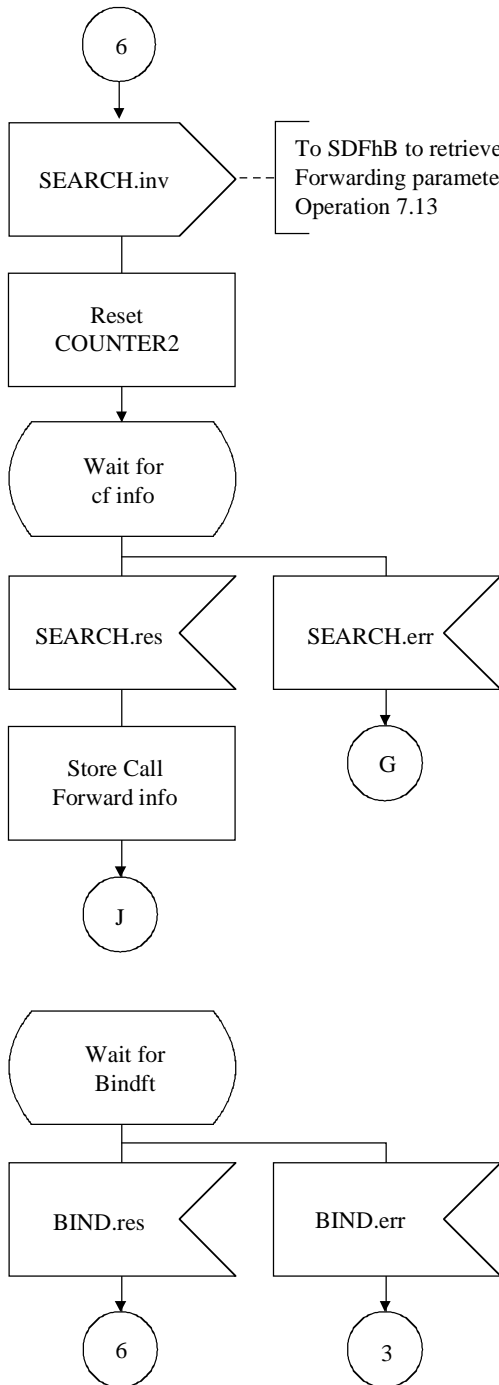
T1183130-96

Figure 13/Q.1551 (sheet 8 of 23) – Outgoing UPT Call procedure



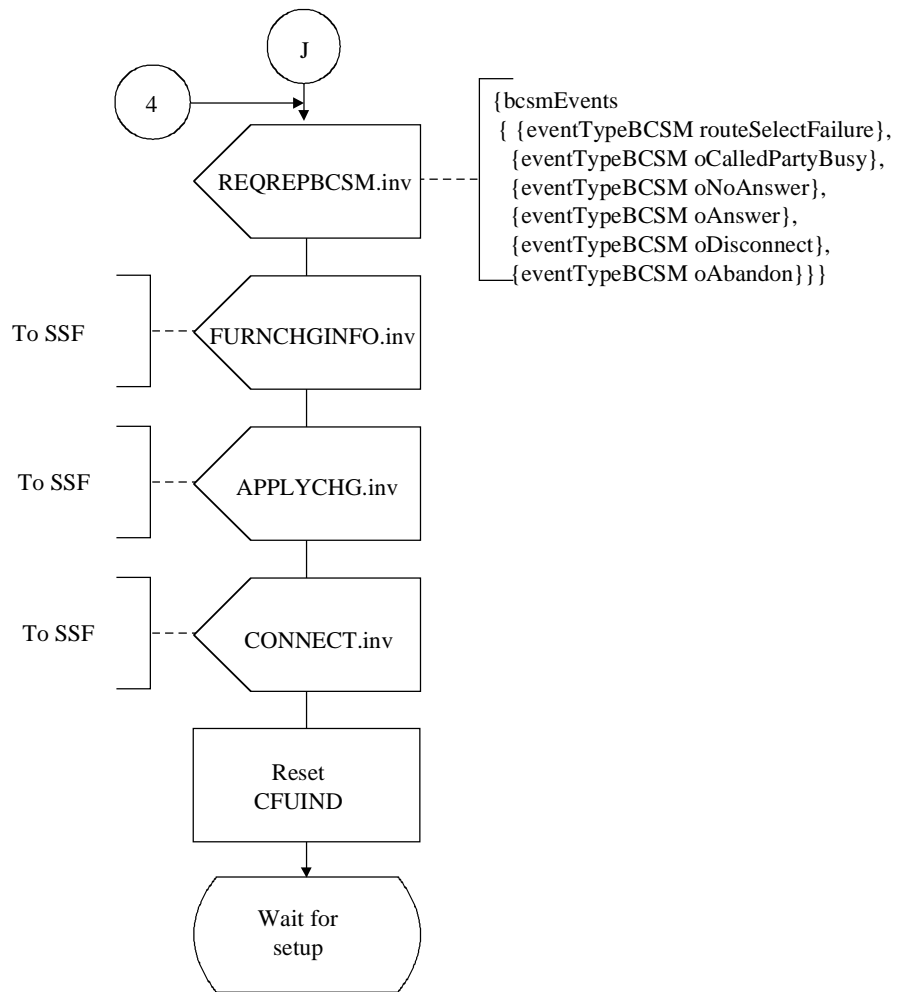
T1183140-96

Figure 13/Q.1551 (sheet 9 of 23) – Outgoing UPT Call procedure



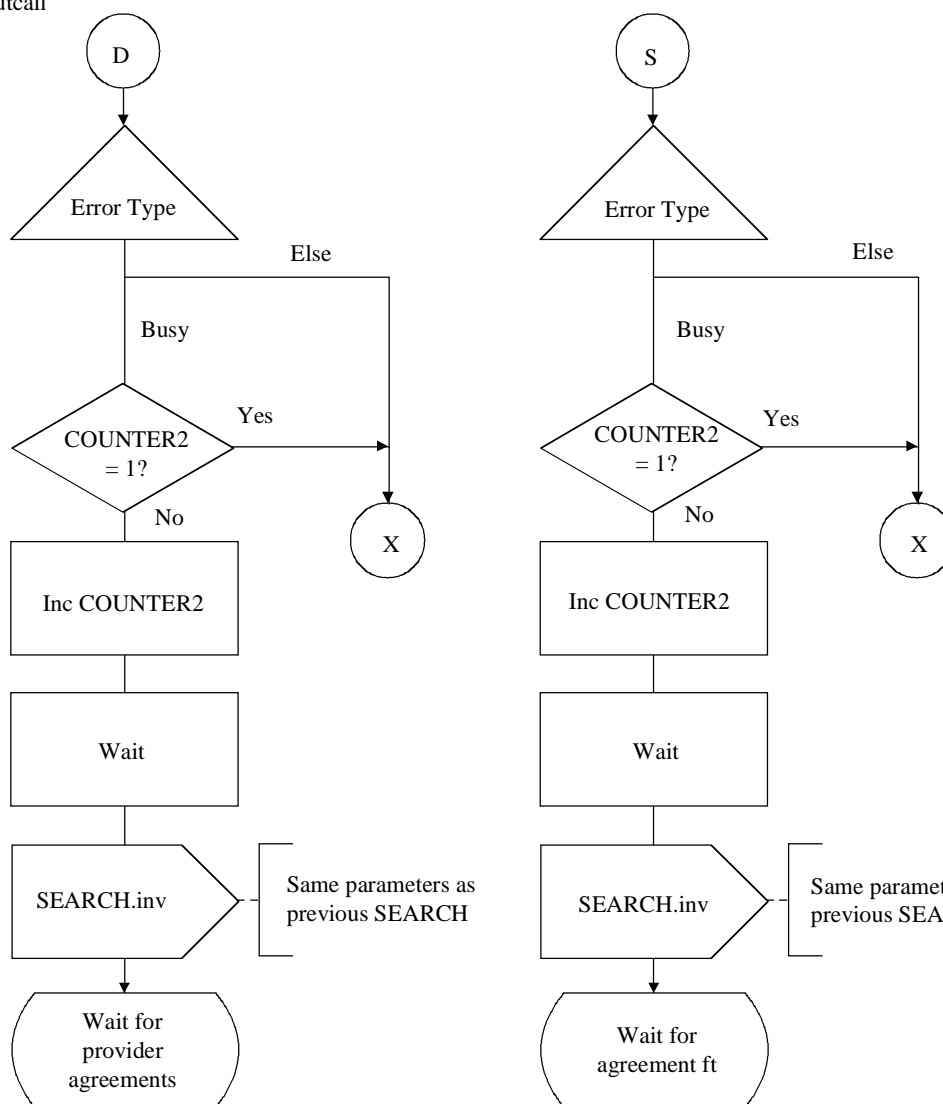
T1183150-96

Figure 13/Q.1551 (sheet 10 of 23) – Outgoing UPT Call procedure



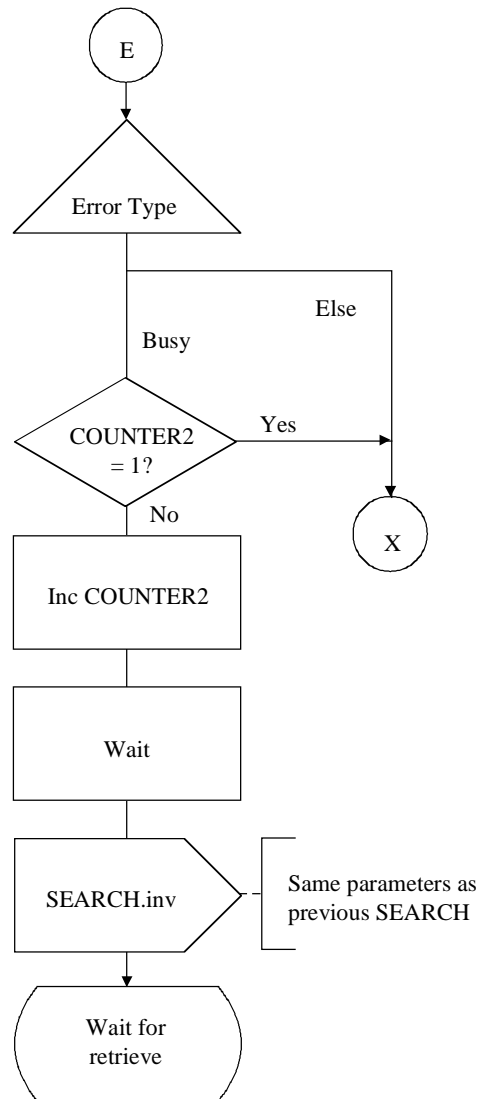
T1183160-96

Figure 13/Q.1551 (sheet 11 of 23) – Outgoing UPT Call procedure



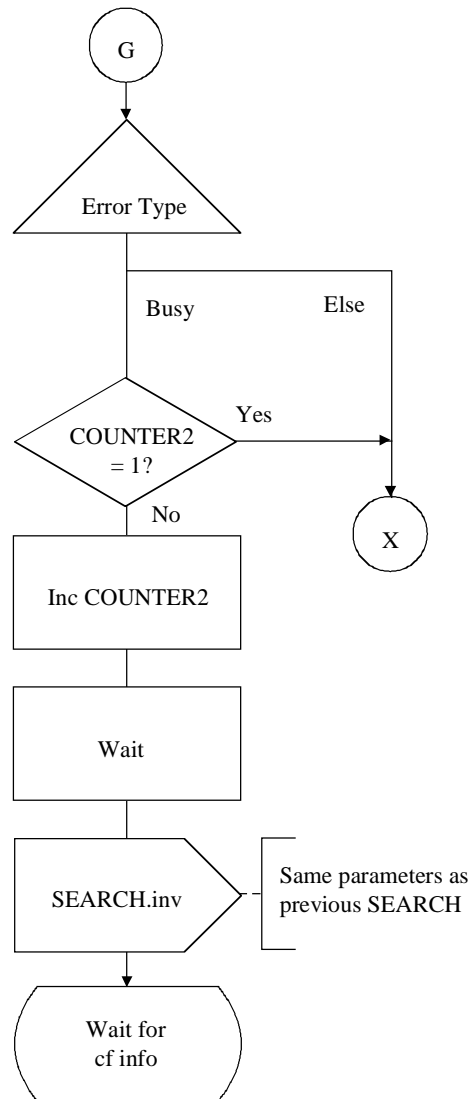
T1183170-96

Figure 13/Q.1551 (sheet 12 of 23) – Outgoing UPT Call procedure



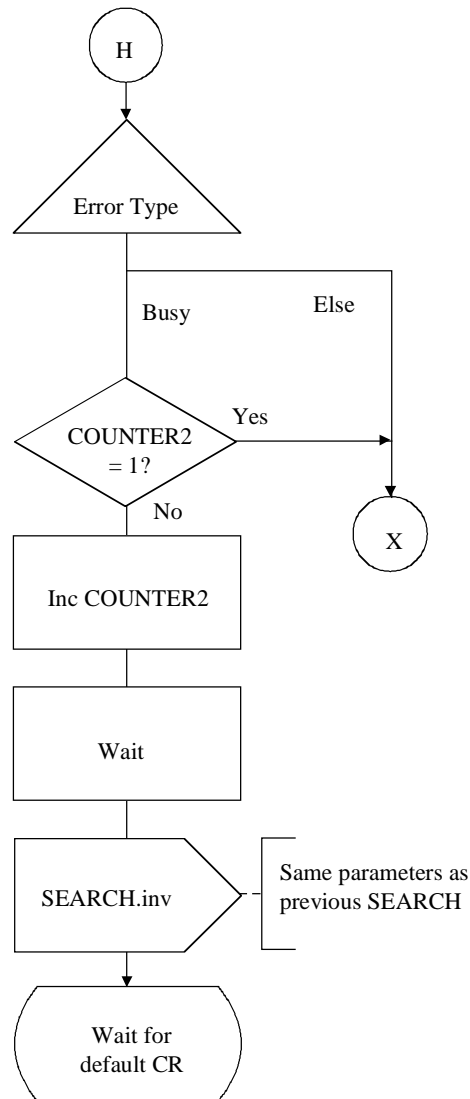
T1183180-96

Figure 13/Q.1551 (sheet 13 of 23) – Outgoing UPT Call procedure



T1183190-96

Figure 13/Q.1551 (sheet 14 of 23) – Outgoing UPT Call procedure



T1183200-96

Figure 13/Q.1551 (sheet 15 of 23) – Outgoing UPT Call procedure

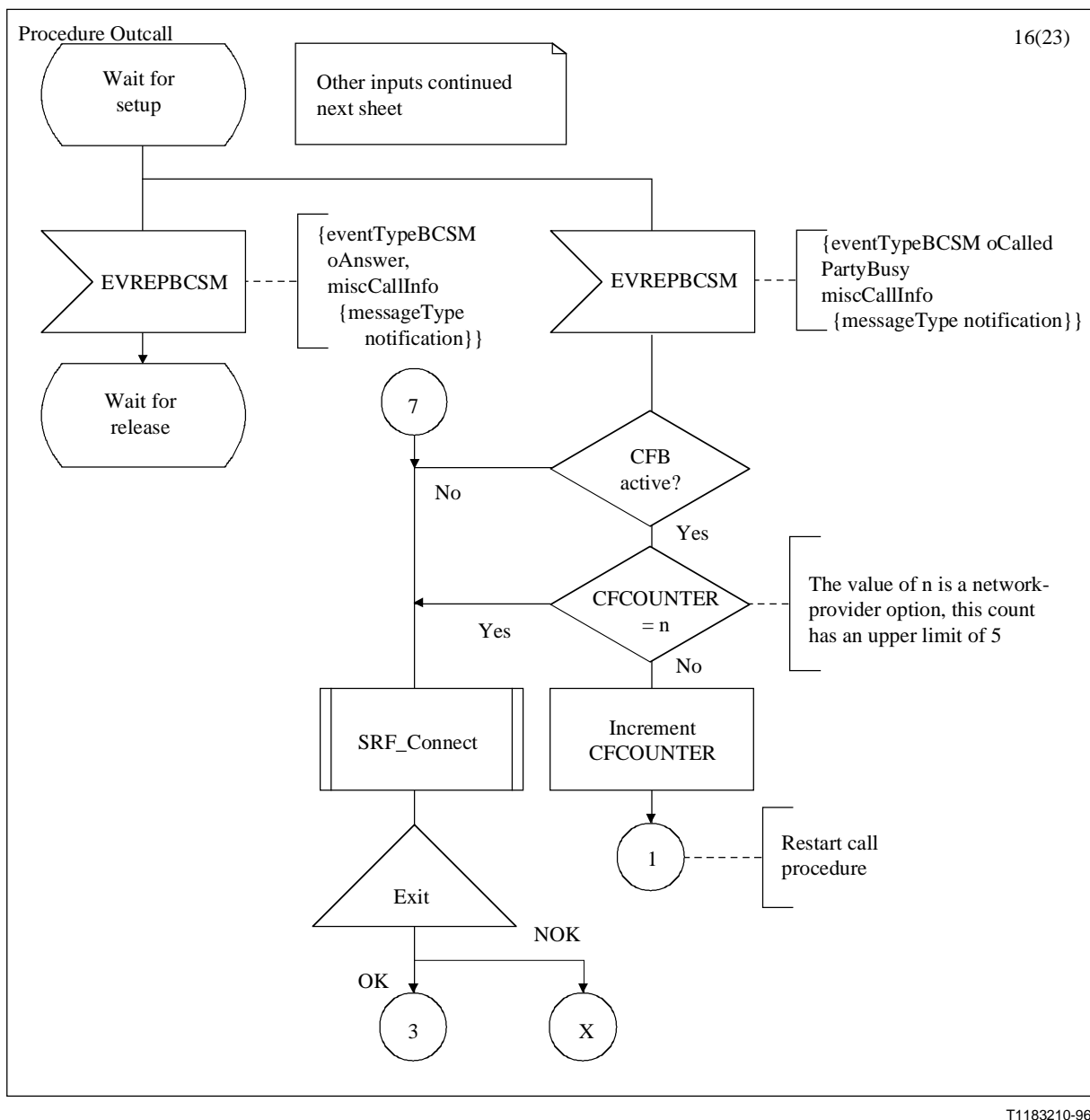


Figure 13/Q.1551 (sheet 16 of 23) – Outgoing UPT Call procedure

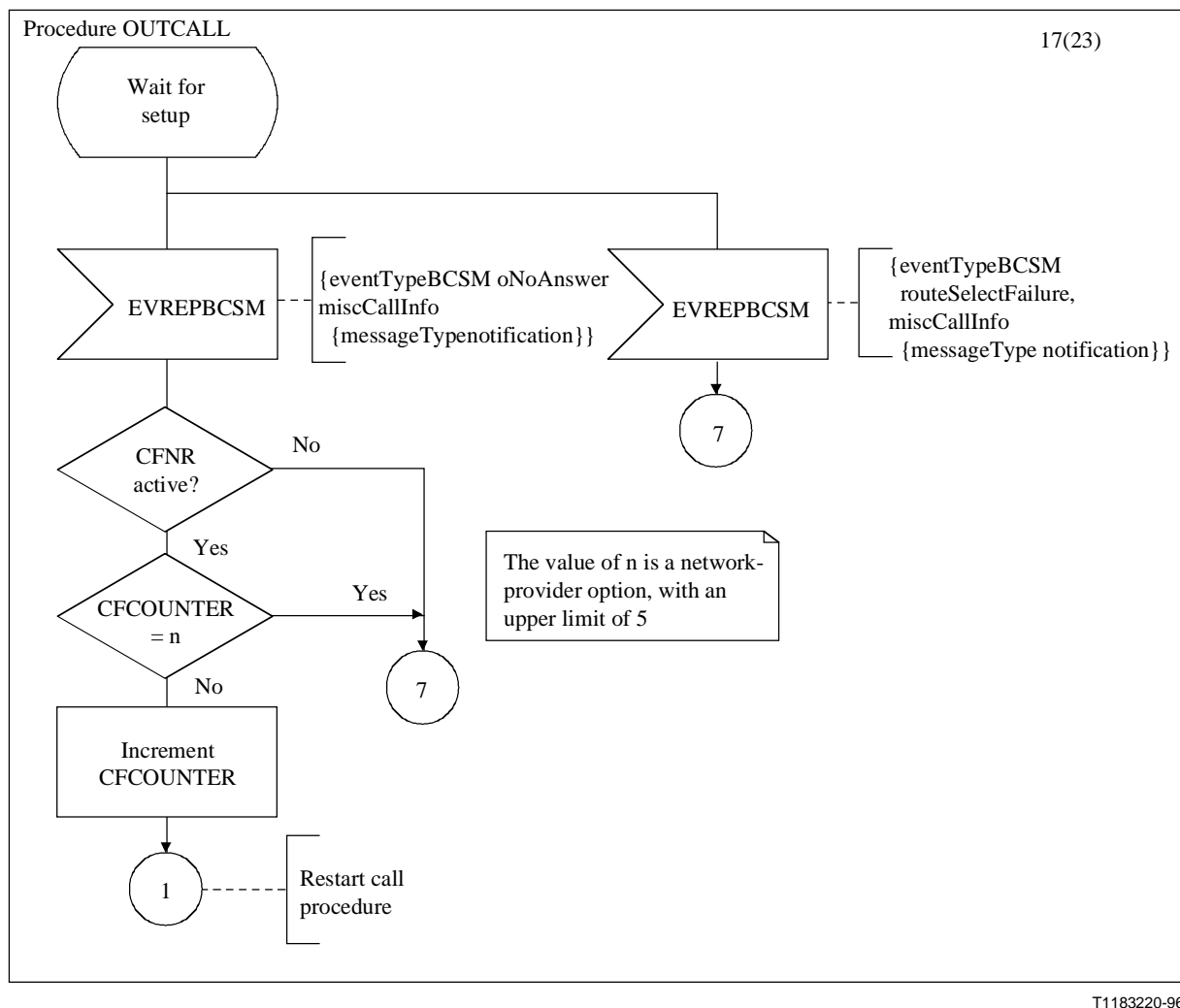


Figure 13/Q.1551 (sheet 17 of 23) – Outgoing UPT Call procedure

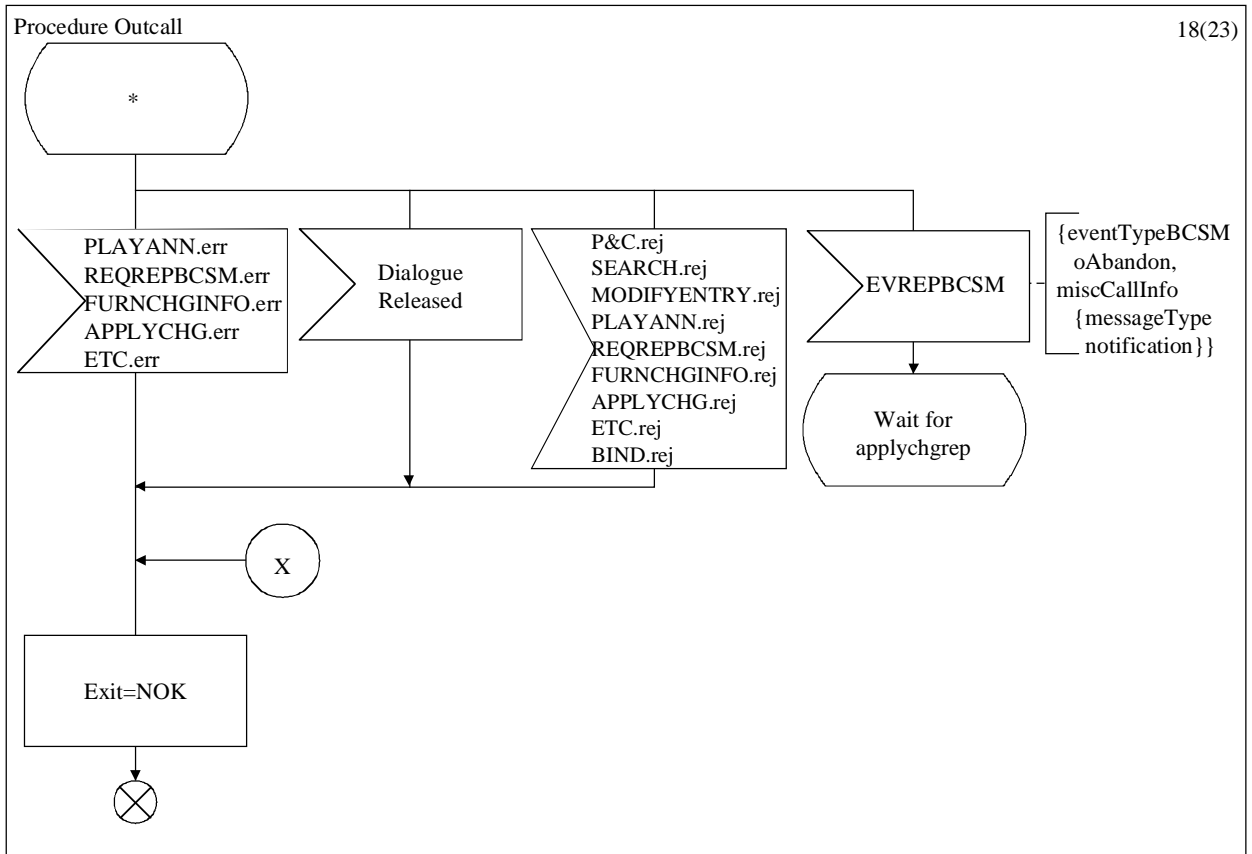


Figure 13/Q.1551 (sheet 18 of 23) – Outgoing UPT Call procedure

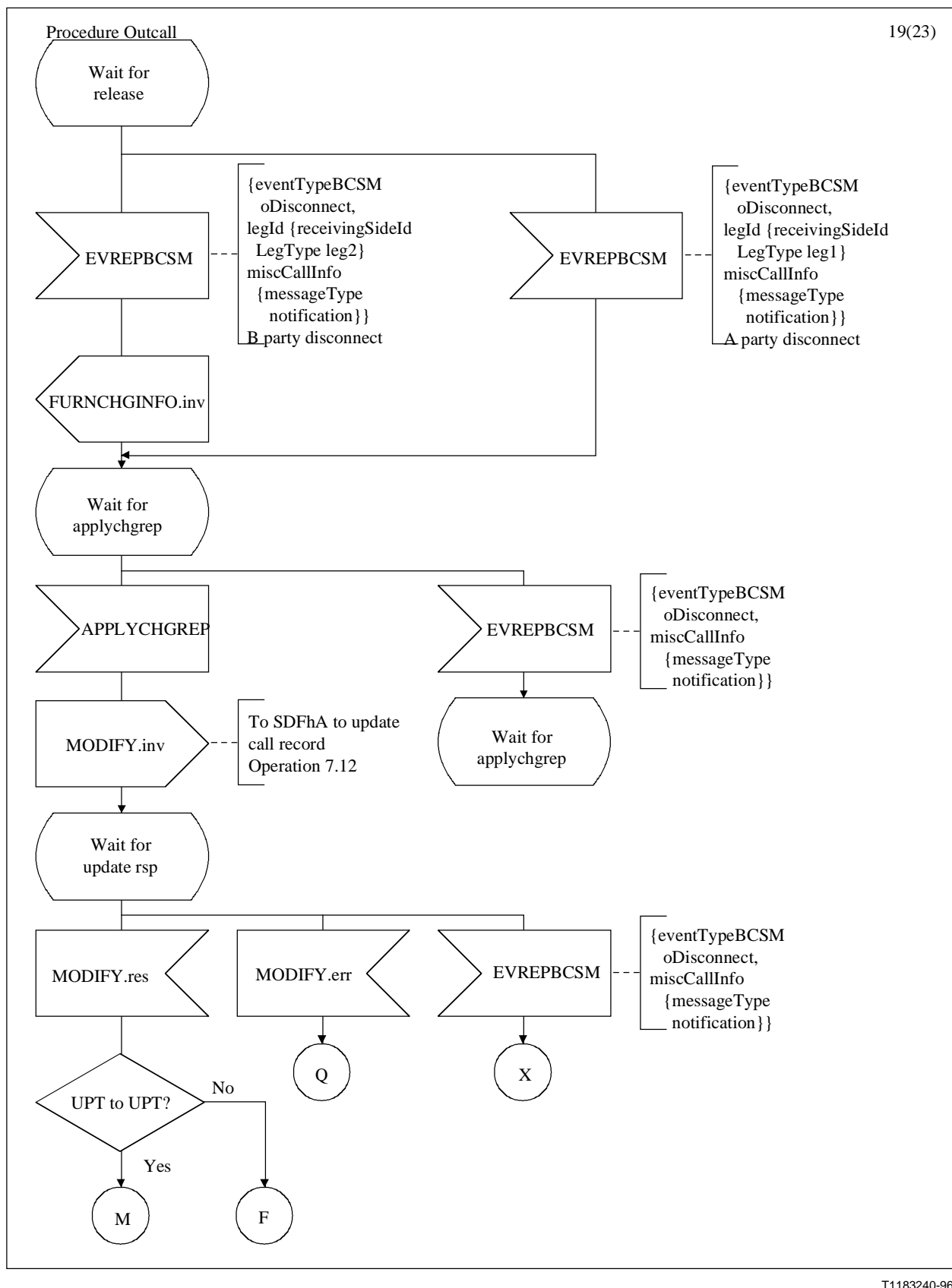
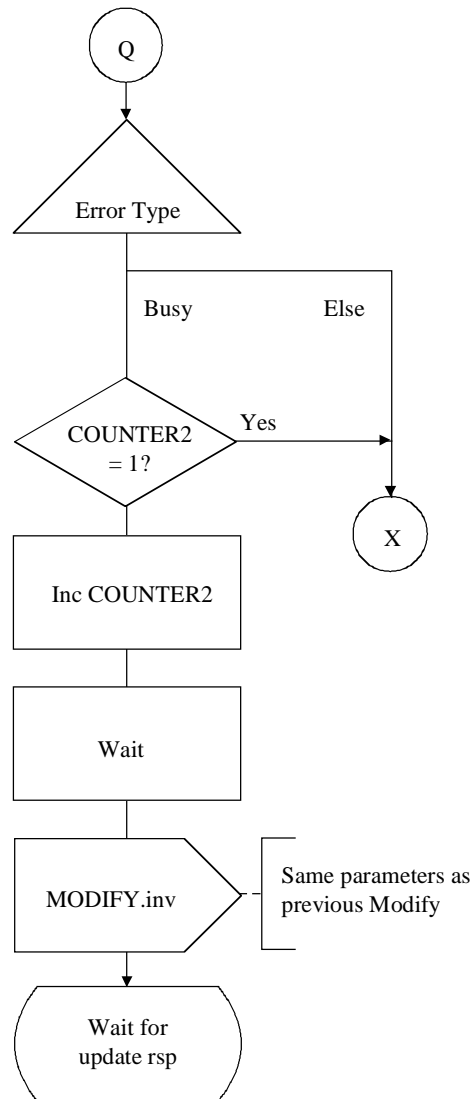
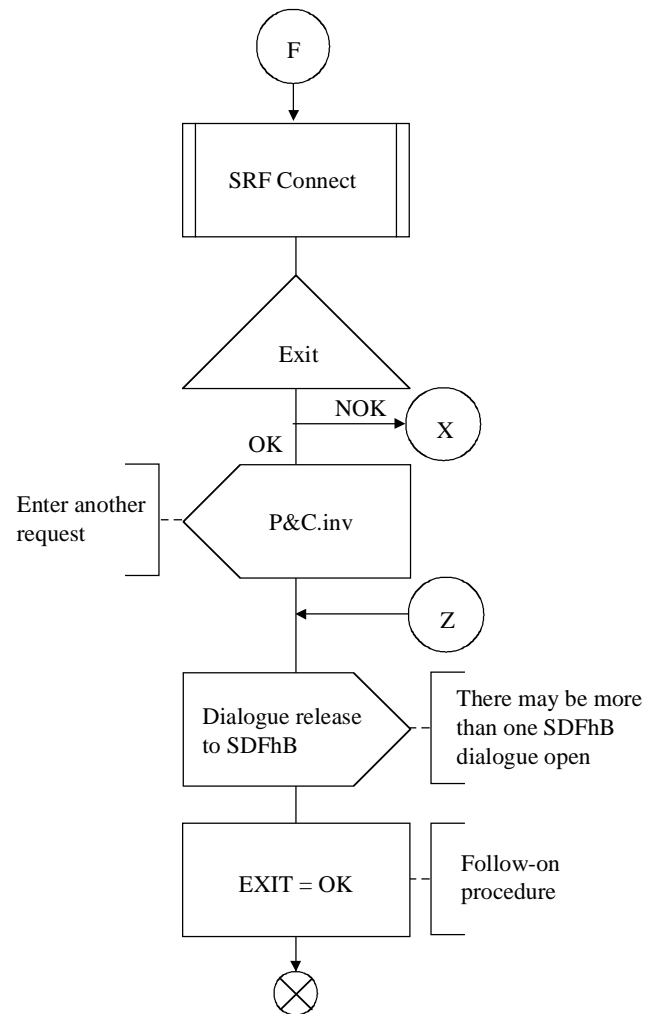


Figure 13/Q.1551 (sheet 19 of 23) – Outgoing UPT Call procedure



T1183250-96

Figure 13/Q.1551 (sheet 20 of 23) – Outgoing UPT Call procedure



T1183260-96

Figure 13/Q.1551 (sheet 21 of 23) – Outgoing UPT Call procedure

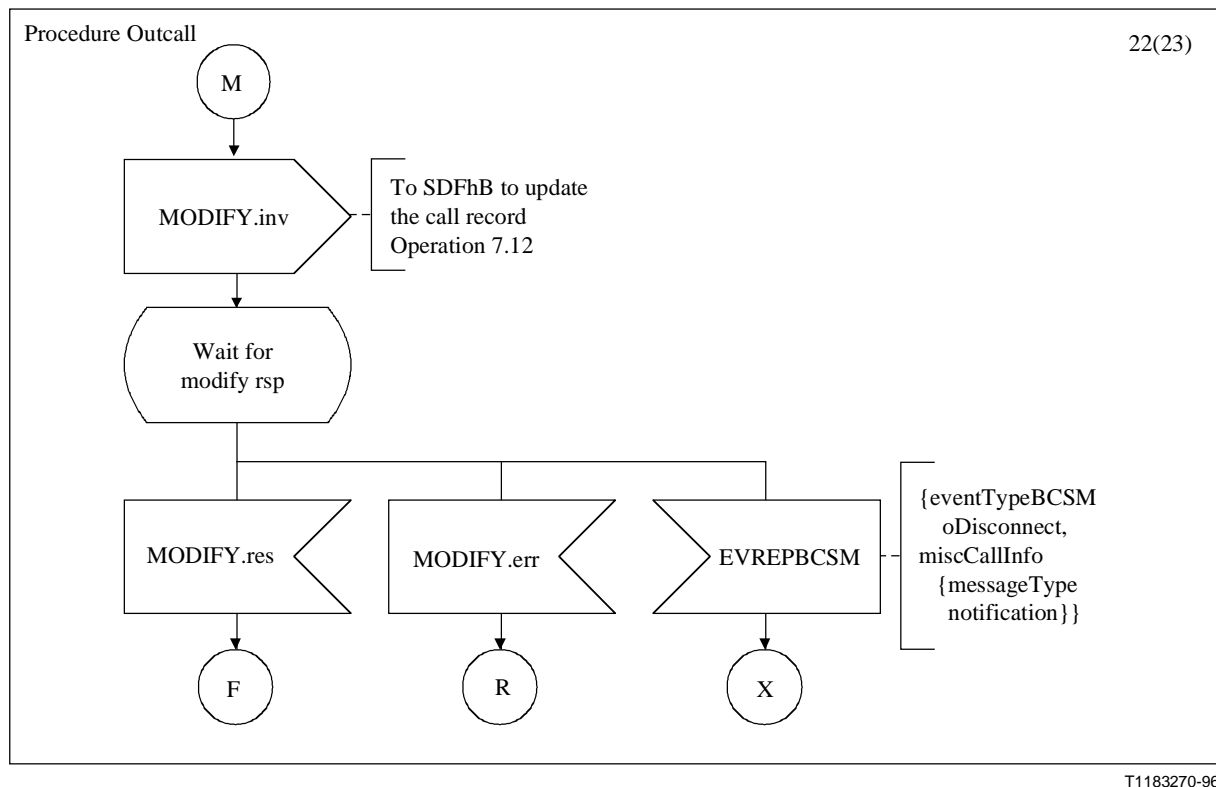
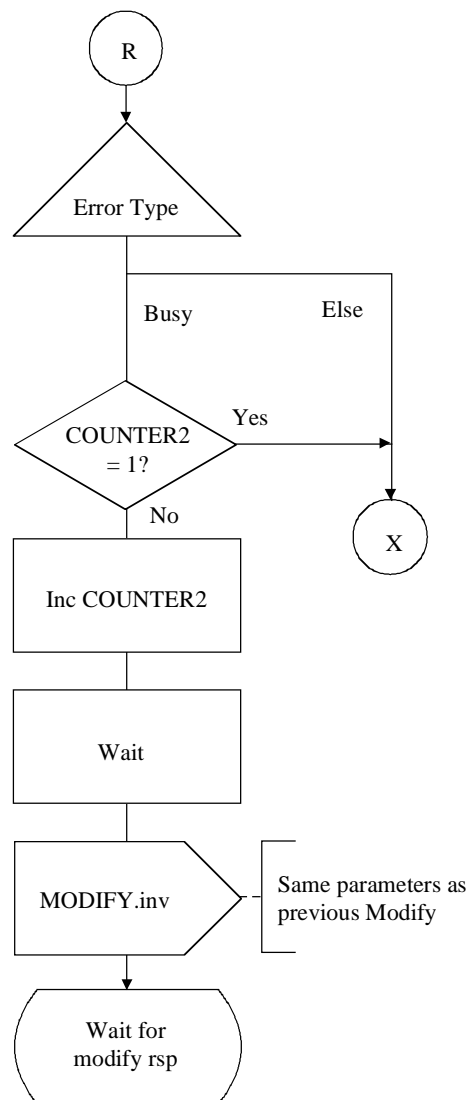


Figure 13/Q.1551 (sheet 22 of 23) – Outgoing UPT Call procedure



T1183280-96

Figure 13/Q.1551 (sheet 23 of 23) – Outgoing UPT Call procedure

6.4.2 Incoming UPT call

6.4.2.1 General

This subclause only describes the case of a non-UPT user calling a UPT user. The case of UPT user calling another UPT user is covered in the previous subclause (6.4.1). It is assumed that a call to a UPT user is always controlled from the "first" network with UPT capabilities. This network may be either the originating network or the called user's home network or another network.

6.4.2.2 Detailed procedure

The procedure for incoming call handling for UPT calls is described in Figure 14. The incoming call procedure is called by the process UPT_SLP as described in 6.1.3.

If Redirection Information is provided in the INITIALDP operation, the Call Forwarding Counter (CFCOUNTER) is set to the same value as the Redirection Counter (see Recommendation Q.763 [7])

for a description of this ISUP information element); otherwise, the Call Forwarding Counter (CFCOUNTER) is set to zero.

This procedure uses two indicators, Call Forwarding Unconditional (CFUIND) and Charging Announcement (ANNIND). The Call Forwarding Unconditional indicator is used to indicate that Call Forwarding Unconditional service has occurred. The Charging Announcement indicator is used to indicate when the calling party is informed that UPT charging is applicable.

Provider agreements

The purpose of the first SEARCH operation is to check the SDFo to determine whether agreement exists between the local service provider and the called user's home provider for establishing incoming calls.

NOTE 1 – If the local Service Provider and the called UPT user's home provider are the same, this part of the procedure is not required; instead the procedure continues with the retrieval of the location of the called user.

The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 14 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the call handling procedure is terminated and the call released.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If the result is a match (i.e. agreement exists between the local service provider and the called UPT user's home provider for establishing incoming calls), then the procedure will continue as described below (Screening of the home database). If there is no agreement to establish incoming calls, the call will be treated as a normal call and the procedure continues with "Call set-up" as described in the subclause below.

Screening of the home database

Having established that agreements exist between the local and home service providers, the SCF can now proceed to check the called UPT user's home database to retrieve the current location of the called user. The dialogue is opened with an "empty" Directory BIND operation (i.e. the Credentials parameter will not be present), the outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or BIND.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;

- operation error returned (BIND): the possible error causes returned in the BIND operation (.err) are described in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- successful result (SEARCH): this means that the operation has been successfully executed by the SDF. The dialogue with the SDFhB can now proceed.

Retrieve location of the called UPT user

To retrieve the location of the called user the SCF will invoke a SEARCH operation. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBSCM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (SEARCH): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 14 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the call handling procedure is terminated and the call released.
- successful result (SEARCH): this means that the operation has been successfully executed by the SDF. The result is one or more valid routing addresses. The SCF will select the routing address based on the following priority:
 - 1) the routing address for the Call Forwarding Unconditional service, if this service is active;

NOTE 2 – If this option is selected the Call Forwarding Counter (CFCOUNTER) will be incremented.
 - 2) if the Call Forwarding Unconditional service is not active but the registration is still valid, the routing address used will be the registration address;
 - 3) if the Call Forwarding Unconditional service is not active but the Variable Routing service is active, the address used will depend on the time or on the calling user;
 - 4) default registration address if none of the above criteria apply.

Default charging reference point

If the call is being forwarded the retrieval of the default charging reference point is not required, the charging for the forwarded leg of the call is a matter for the original called user and is not described here.

To retrieve the default charging reference the SDFhB is interrogated by the SCF using a SEARCH operation, the outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;

- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation result (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 14 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDFhB and the default charging reference point retrieved.

If split charging [8] is not to be applied the procedure continues with the "UPT charging notification" subclause below. If split charging is to be applied then the procedure continues with the "Credit limit check".

Credit limit check

The SCF will also check the called users Credit Limit to determine that there is sufficient credit available to receive the call. This check will not be repeated for subsequent call set-up attempts (i.e. UPT to UPT Call Forwarding has occurred). The SCF sends a SEARCH operation, the outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 14 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the call handling procedure is terminated and the call released.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If credit is available to receive the call, the procedure will continue as described below. If there is no credit available the call handling procedures are terminated and the SSF is instructed to release the call.

UPT charging notification

The SCF instructs the SSF to establish a connection with an SRF. This is done with the SRF_Connect procedure described in 6.2.4. If the above procedure was completed successfully, the SCF then instructs the SRF to play an announcement to the calling user to inform that UPT charging is applicable by invoking the PLAYANN operation. This notification will not be repeated for subsequent call set-up attempts for this call (i.e. UPT to UPT call forwarding has occurred and the calling user was notified that UPT charging was applicable from a previous attempt to set up this

call). Following the instruction for the PLAYANN operation the macro SRF_Disconnect is called, this macro will handle the operation errors and the disconnection of the SRF.

Retrieve supplementary service information

If the original called party was another UPT user but as a result of Call Forwarding Unconditional, the forwarded-to party is not a UPT user, this part of the procedure is not required. Instead, the procedure will continue with Call Set-up.

The purpose of this part of the procedure is to query the called UPT users (or forwarded-to UPT user if Call Forwarding has occurred) home database (SDFhB) for the status of supplementary services and for those call forwarding services which are active, retrieve the relevant conditional forwarding parameters (e.g. No Reply Condition Timer).

For the case where the Call Forwarding Unconditional service has been activated the new forwarded-to UPT Users home database (SDFhB) needs to be queried, it has been assumed that the SDF of the original called party and the new forwarded-to party are not the same. If this is not the case, neither the provider check nor the BIND procedure are required as the same dialogue will be used and the procedure continues with the SEARCH to retrieve supplementary service information described below.

Prior to establishing a dialogue with the new forwarded-to SDFhB, the originating database (SDFo) is searched to check if agreements exist between the local service provider and the new forwarded-to UPT user's home provider for establishing incoming calls. This interrogation is performed by invoking the SEARCH operation. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 14 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDF. If the result is a match (i.e. agreement exists between the local service provider and the forwarded-to UPT user's home provider for establishing incoming calls from the current location), then the procedure will continue as described below. In the case of no match (i.e. there is no agreement to establish incoming calls), the call will be treated as a normal call and routed to another network for completion; for this, the procedure continues as described in the subclause "Call set-up".

Following the successful service provider check, the database of the home network of the new forwarded-to UPT user (SDFhB) is then interrogated to retrieve the supplementary service information. The dialogue is opened with an "empty" Directory BIND operation (i.e. the Credentials parameter will not be present), the outcome of this operation can be one of the following events:

- dialogue released by IN node (Dialogue_released or BIND.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (BIND.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]. Regardless of the reason for the error, the call handling procedure is terminated and the SSF is instructed to release the call;
- successful result (BIND.res): this means that SDFhB accepts the dialogue and the procedure continues as described below.

To retrieve the supplementary service information the SCF will invoke a SEARCH operation for this. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or SEARCH.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation result (.err) are described in Recommendation Q.1218 [1]:
 - in the case of 'Service Error' type Busy the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed. Note that Figure 14 only shows the possibility of one retry, the maximum number of retries is Network Specific;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- successful result (SEARCH.res): this means that the operation has been successfully executed by the SDFhB. The call forwarding parameters retrieved (if any) are stored for future use. The procedure continues as described below.

Call set-up

On successful completion of the above procedures the SCF can now instruct the SSF to set up the call, several operations are invoked for this purpose:

- FURNCHGINF0: this requests the SSF to generate call record information for the following event;
- APPLYCHG: the purpose of the operation is to request the SSF to report back to the SCF when a charging-related event has been detected (i.e. if UPT user is charged);
- REQREPBSCM: this requests the SSF to monitor for a call-related event (e.g. busy, no answer, release, etc.) and report back to the SCF when the event has been detected. If a No Reply Condition Timer value was retrieved from the called UPT user's home database (SDFhB) when searched for supplementary service information, the value will be provided with this operation;
- CONNECT: this instructs the SSF to set up the call (i.e. generate the IAM). If any redirection information was provided in the INITIALDP operation the information will be

returned in this operation, the redirection counter may have been updated as a result of further call forwarding. If no redirection information was provided in the INITIALDP operation but call forwarding has occurred, then the redirection information will be constructed by the SCF. Redirection information will not be provided if no call forwarding has occurred, The SSF will determine how to handle this information.

The outcome of this procedure will be one of the following events:

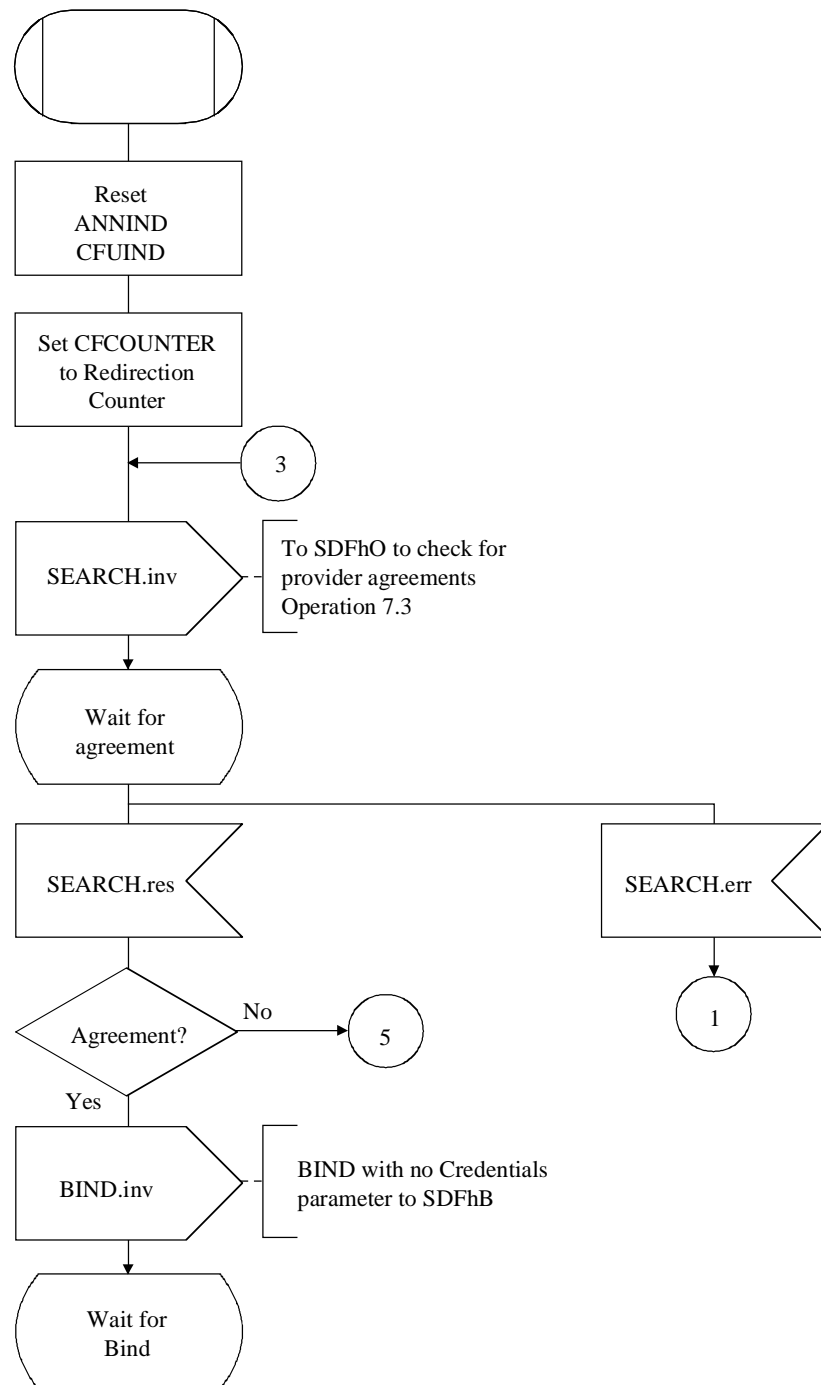
- dialogue released by IN node (Dialogue_released or <operation_name>.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling party released: the SCF is notified of the release by the EVREPBCSM operation from the SSF. The SCF will terminate the call handling procedure;
- called party busy: this state will be reported to the SCF by the SSF returning the operation EVREPBCSM. There are three possible outcomes:
 - if call forwarding on busy service is not active, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
 - if call forwarding on busy service is active but the Call Forwarding Counter (CFCOUNTER) has exceeded the network redirection limit (note that the upper limit for this counter is 5), the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
 - if call forwarding on busy service is active and the Call Forwarding Counter (CFCOUNTER) has not exceeded the network redirection limit, the call can be forwarded. The Call Forwarding Counter (CFCOUNTER) is incremented and the outgoing call procedure restarted.
- called party no answer: this state will be reported to the SCF by the SSF returning the operation EVREPBCSM. There are three possible outcomes:
 - if call forwarding on no reply service is not active, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
 - if call forwarding on no reply service is active but the Call Forwarding Counter (CFCOUNTER) has exceeded the network redirection limit (note that the upper limit for this counter is 5), the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
 - if call forwarding on no reply service is active and the Call Forwarding Counter (CFCOUNTER) has not exceeded the network redirection limit, the call can be forwarded. The Call Forwarding Counter (CFCOUNTER) is incremented and the outgoing call procedure restarted.
- route select failure: this state will be reported to the SCF by the SSF returning the operation EVREPBCSM. This indicates that it was not possible to complete set-up of the call due to either congestion, unsubscribed number or number blocked. The call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- operation error returned (FURNCHGINF0.err, APPLYCHG.err or CONNECT.err): the possible error causes returned in these operations are described in Recommendation Q.1218 [1]. Regardless of the reason for the error, the call handling procedure is terminated and the SSF is instructed to release the call;
- calling Party Answers: the SCF is notified of this event by an EVREPBCSM operation.

Call release

The SCF is informed of the release of the call by the EVREPBCSM operation. The SCF will then send a FURNCHGINFO operation to the SSF to update the call record. Once the APPLYCHGRPT containing the call record is received, the SCF sends a MODIFY operation to the called parties SDFhB to store the call record. It should be noted that it is not possible to directly use the call record to modify the user credit as the SDF is not able to calculate the charge that corresponds to the call record. The outcome of this operation will be one of the following events:

- dialogue released by IN node (Dialogue_released or MODIFY.rej): the reasons for releasing the dialogue prematurely are described in 6.1 and are further elaborated in Recommendation Q.1218 [1]. Regardless of the reason for the release, the call handling procedure is terminated and the RELEASE procedure (as described in 6.2.3) is invoked;
- operation error returned (MODIFY.err): the possible error causes returned in the MODIFY operation (.err) are described in Recommendation Q.1218 [1]:
 - if the error is "Busy", the SCF can make further attempts to retry the same SDF, after a time delay for each retry. If the number of retries is exceeded (COUNTER2) the procedure is terminated. To avoid loss of the call record, an implementation-dependent operation could be performed at this stage;
 - for any other error the procedure is terminated. To avoid loss of the call record an implementation-dependent operation could be performed at this stage.
- successful result (MODIFY.res): this confirms that the SDF has been successfully updated.

NOTE 3 – If this incoming call was a result of UPT to UPT call forwarding from another IN network but the forwarded-to leg of the call could not be recognized as another UPT call by the original SCF (i.e. no provider agreements exist), then the original called parties SDFhB may not be the same SDFhB accessed during this incoming call procedure, as this SDFhB will be the forwarded-to UPT user's SDFh. This may result in discrepancies in the charging information, the problem cannot be solved at present.



T1183290-96

Figure 14/Q.1551 (sheet 1 of 16) – Incoming UPT Call procedures

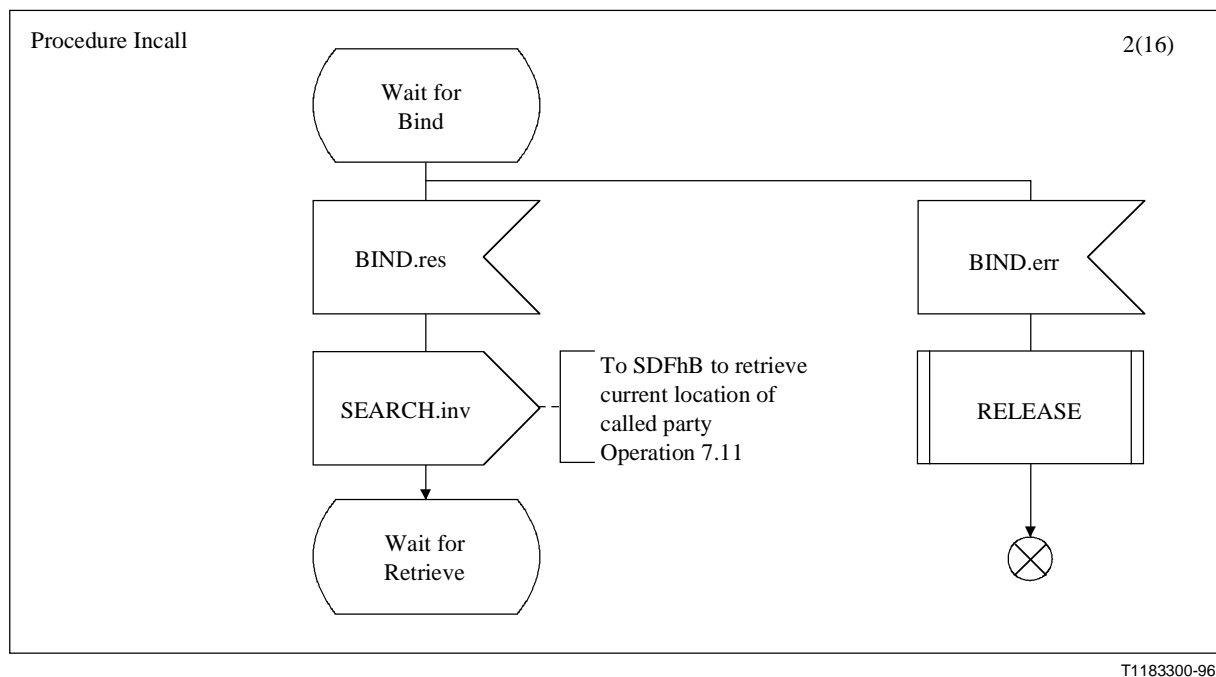
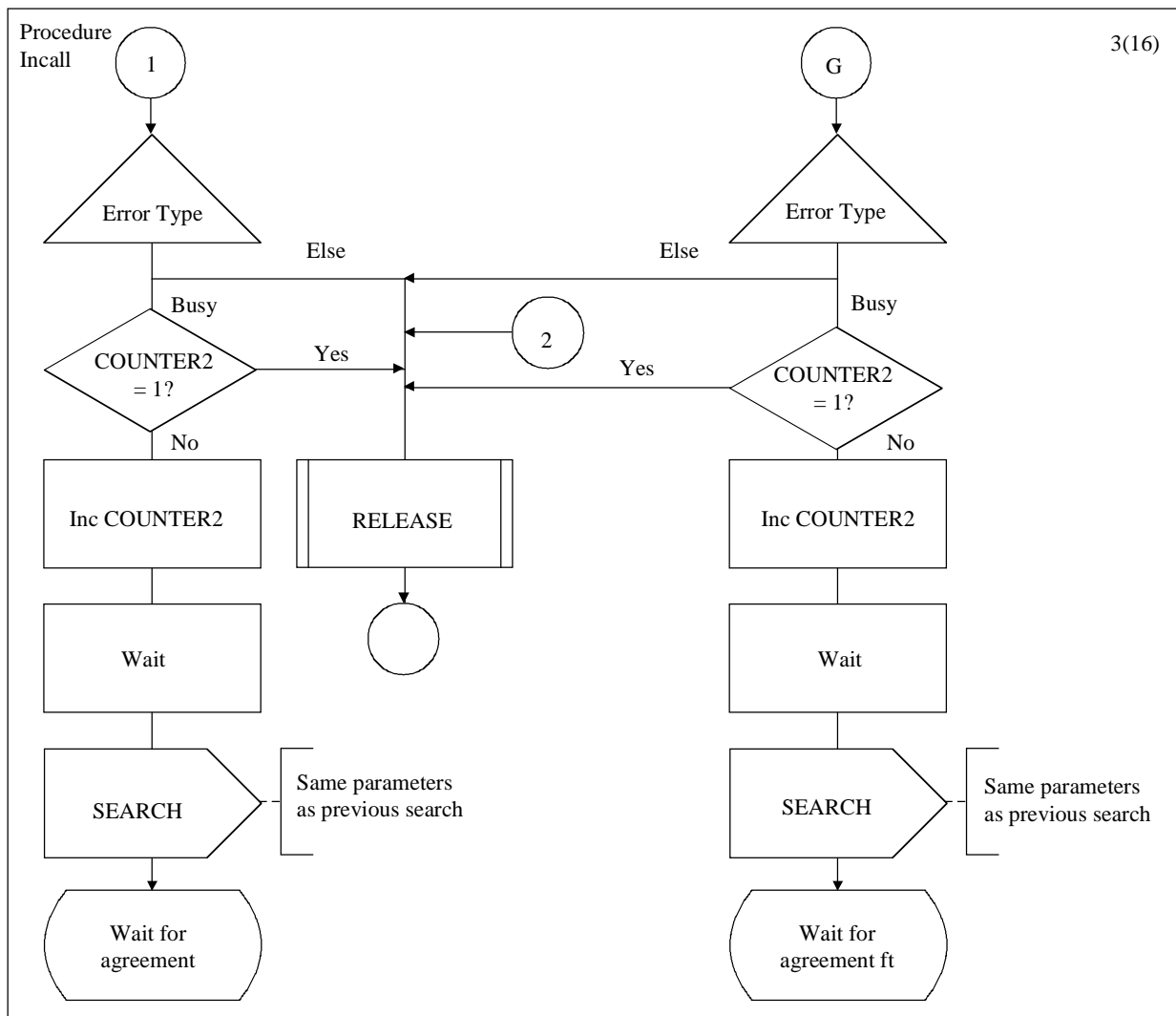
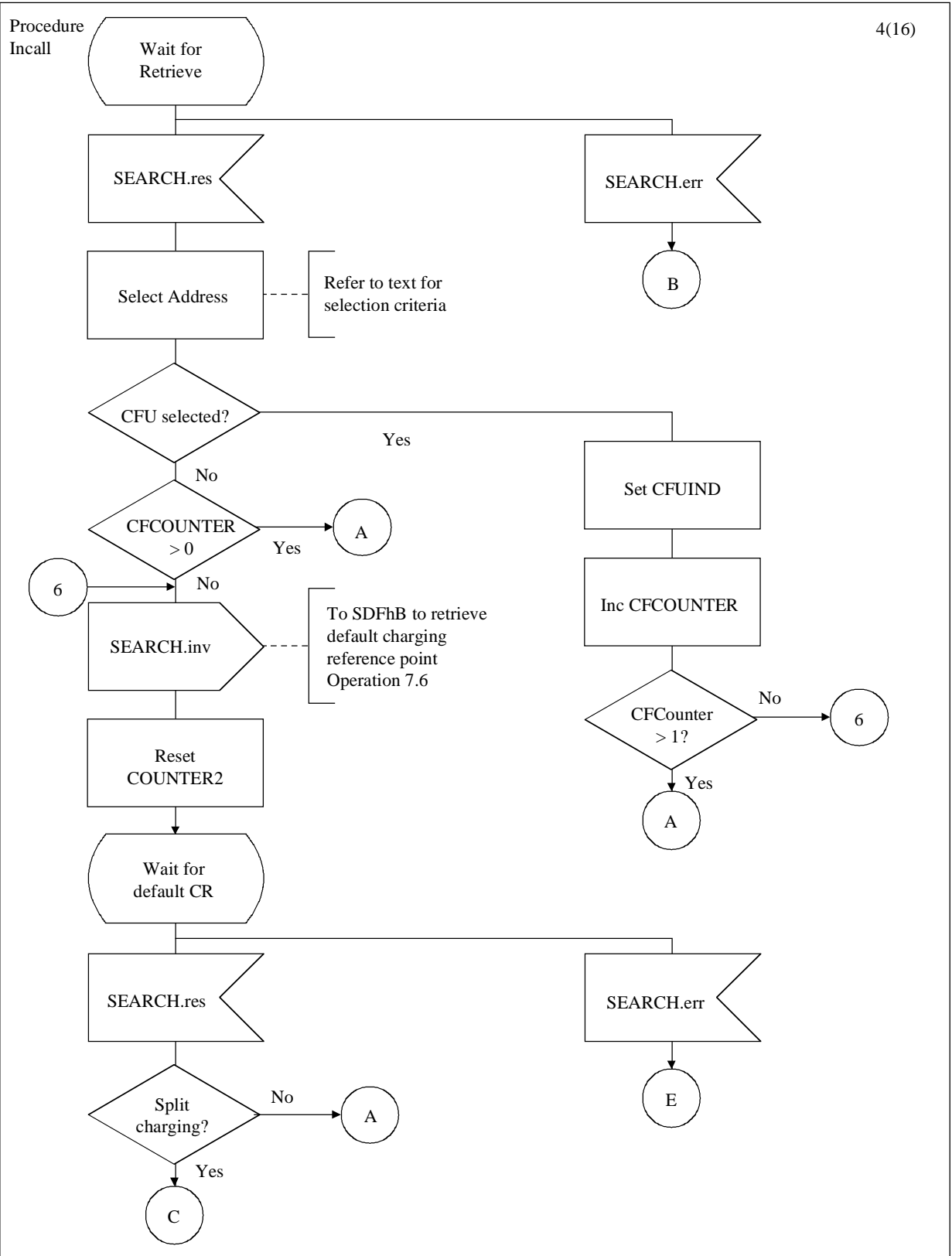


Figure 14/Q.1551 (sheet 2 of 16) – Incoming UPT Call procedures



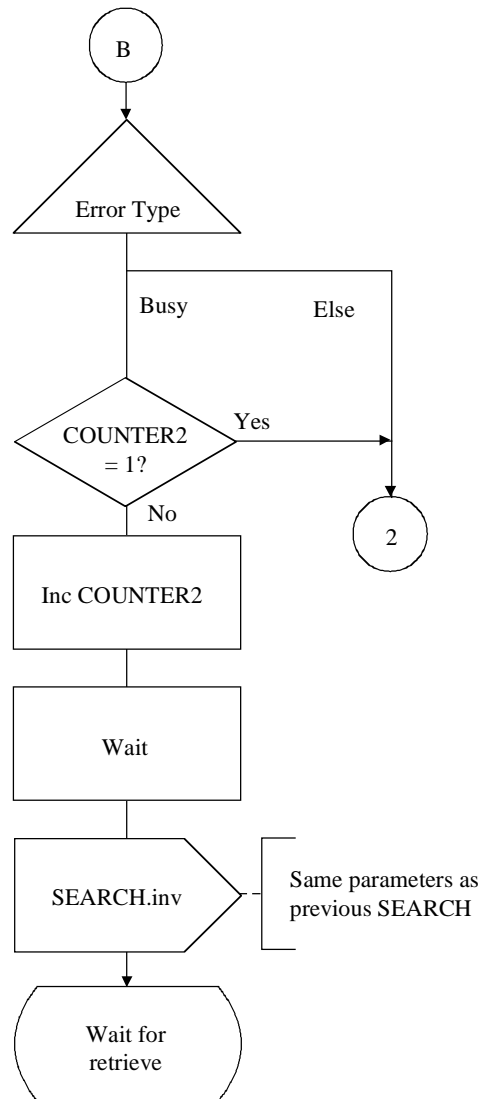
T1183310-96

Figure 14/Q.1551 (sheet 3 of 16) – Incoming UPT Call procedures



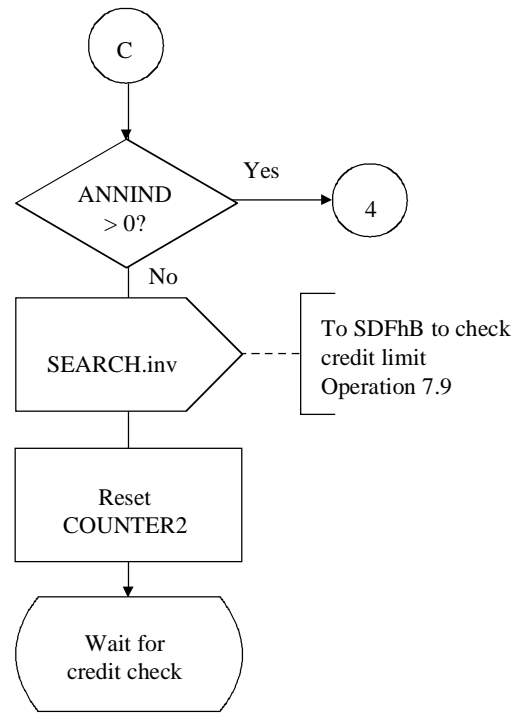
T1183320-96

Figure 14/Q.1551 (sheet 4 of 16) – Incoming UPT Call procedures



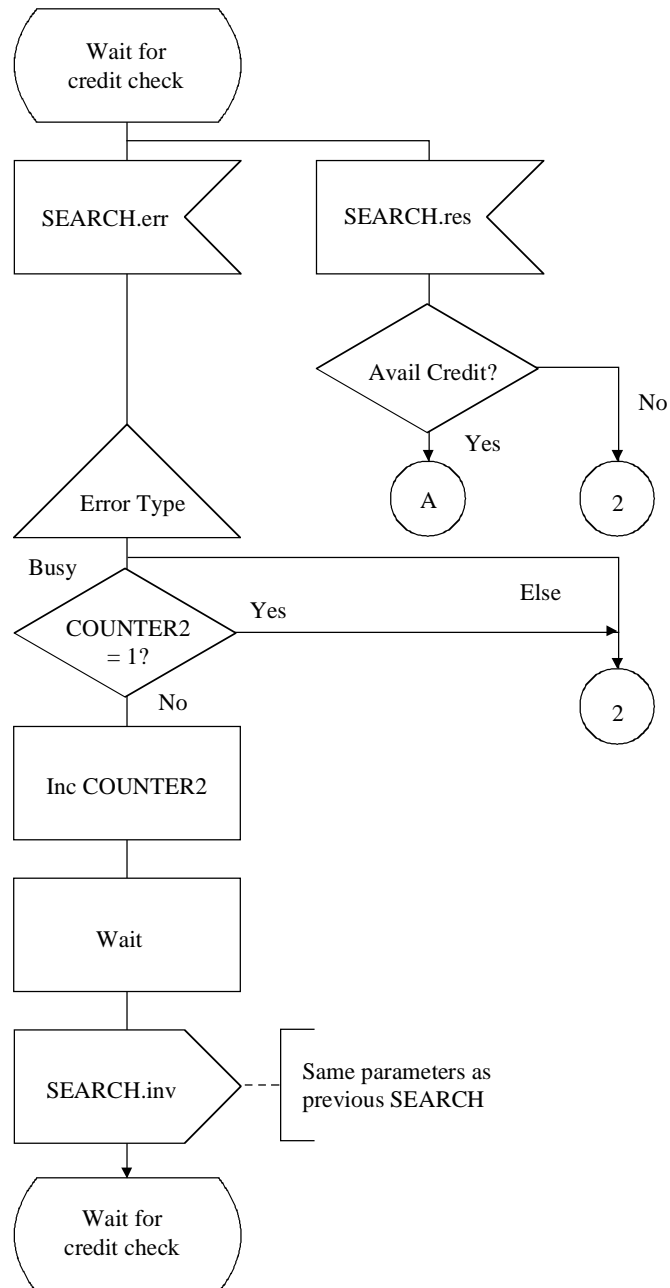
T1183330-96

Figure 14/Q.1551 (sheet 5 of 16) – Incoming UPT Call procedures



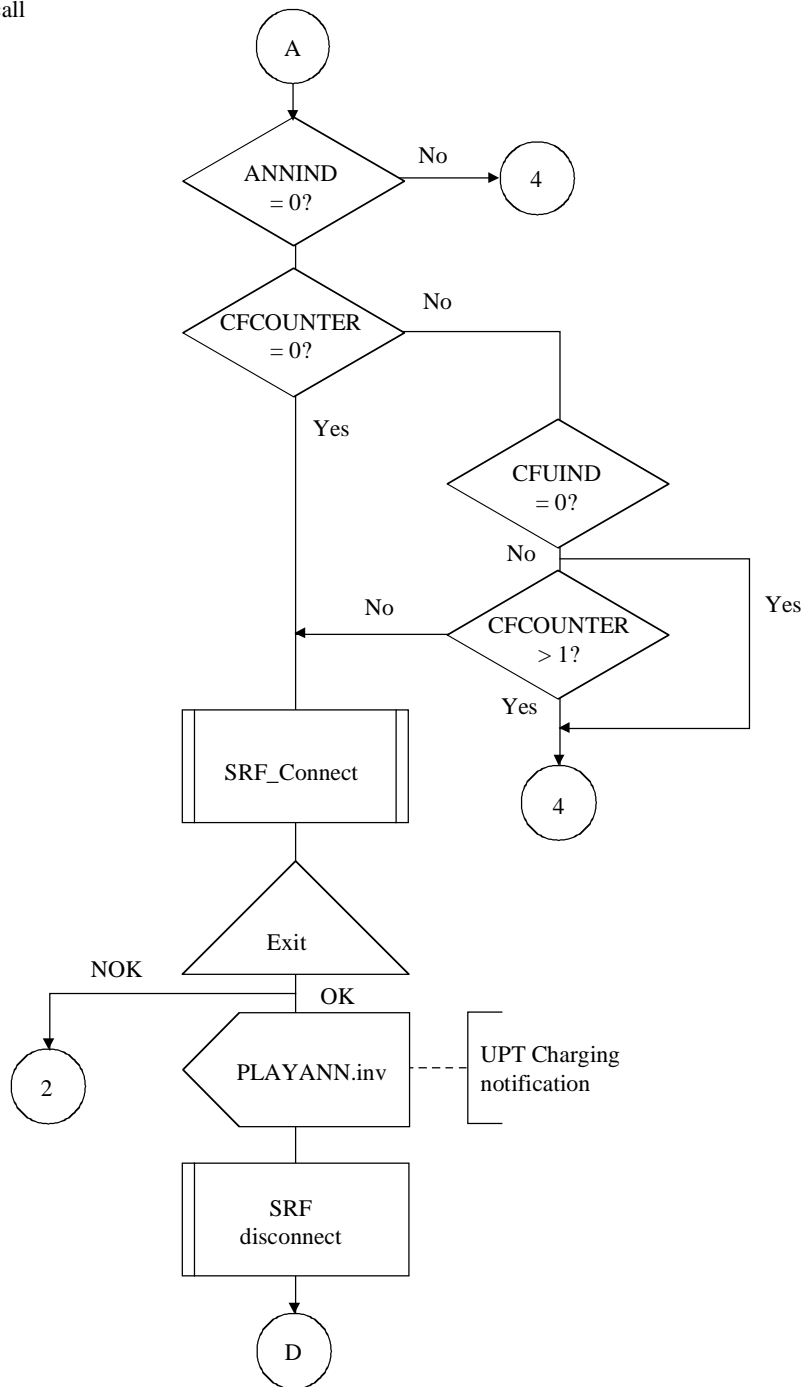
T1183340-96

Figure 14/Q.1551 (sheet 6 of 16) – Incoming UPT Call procedures



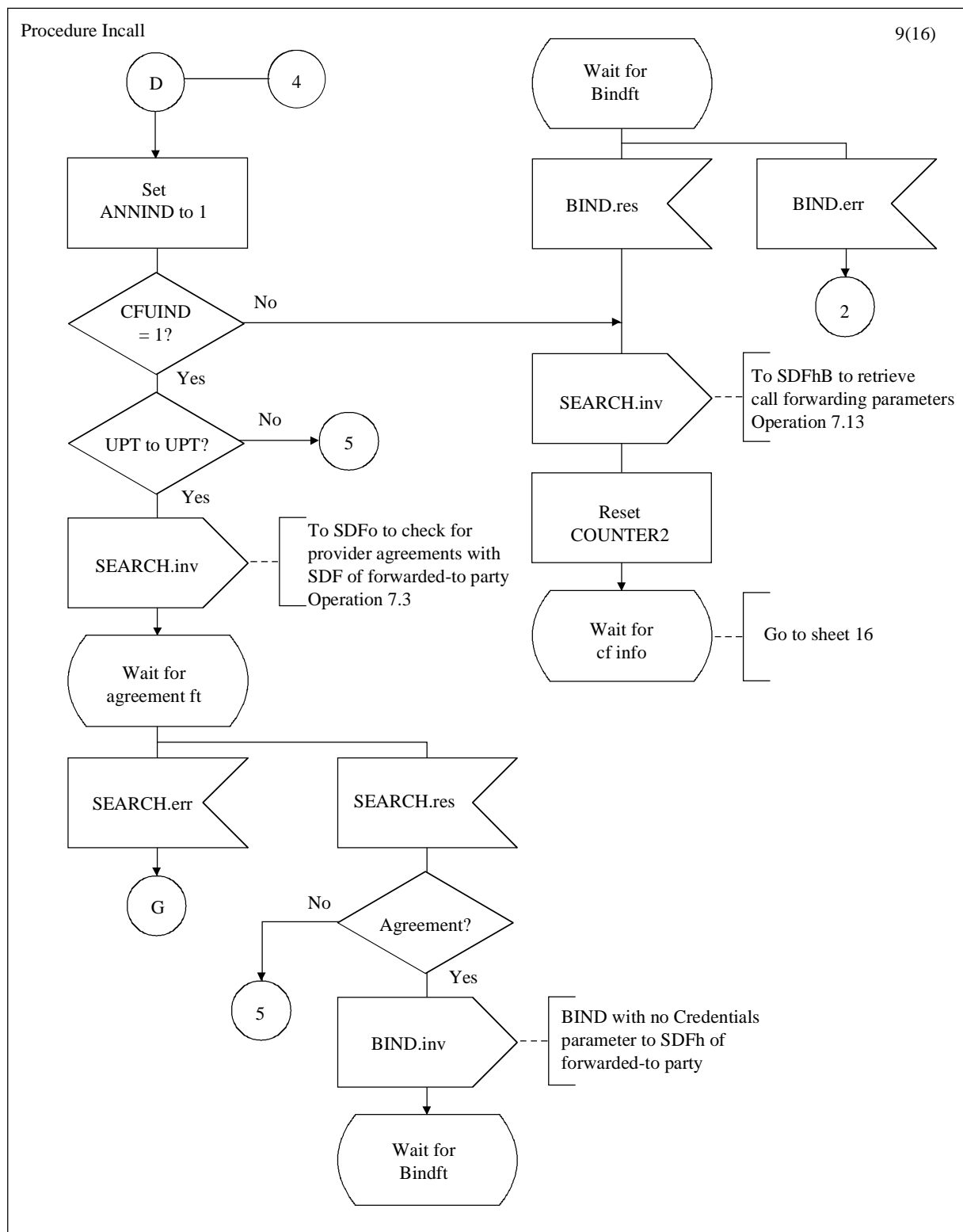
T1183350-96

Figure 14/Q.1551 (sheet 7 of 16) – Incoming UPT Call procedures



T1183360-96

Figure 14/Q.1551 (sheet 8 of 16) – Incoming UPT Call procedures



T1183370-96

Figure 14/Q.1551 (sheet 9 of 16) – Incoming UPT Call procedures

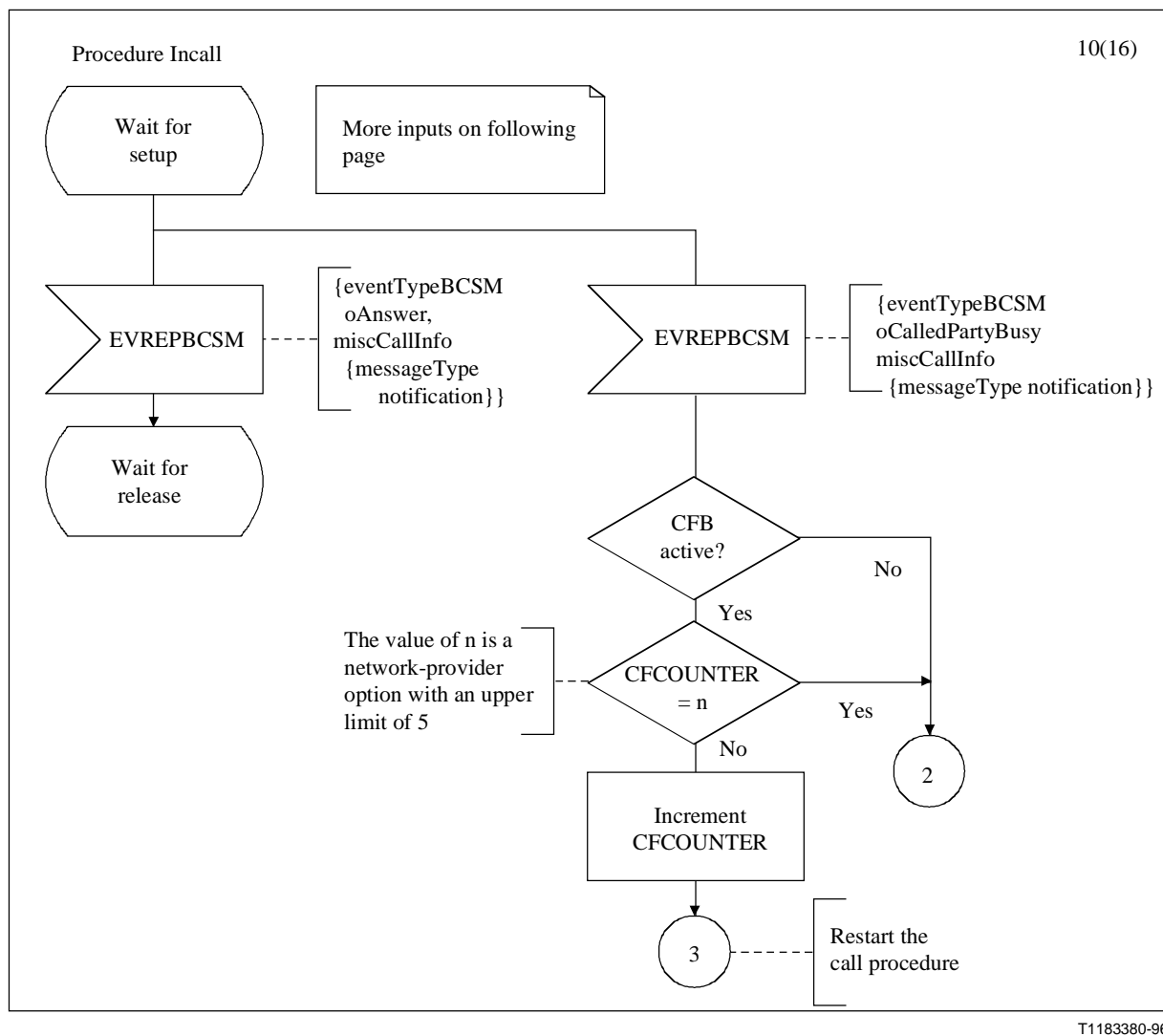
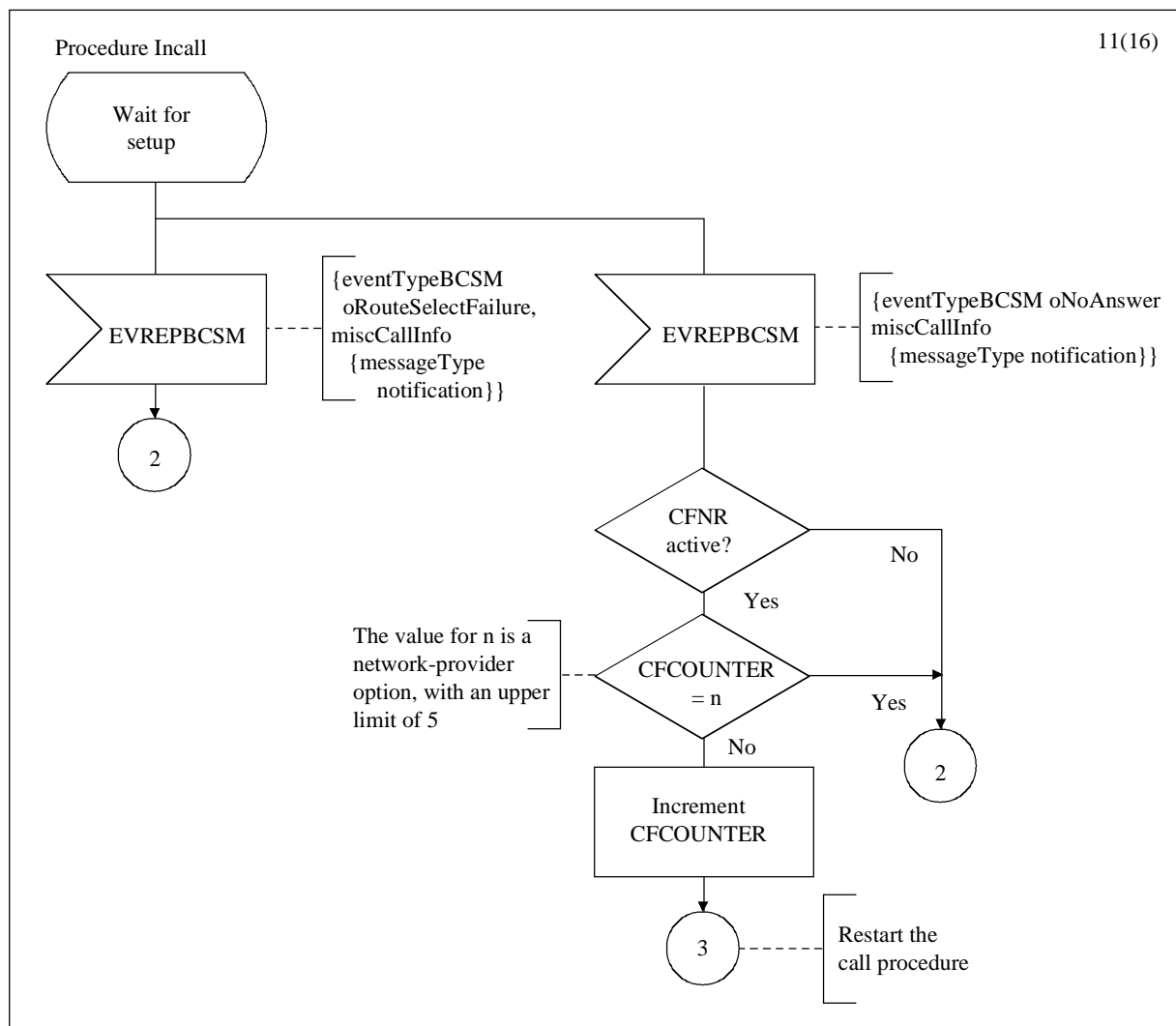


Figure 14/Q.1551 (sheet 10 of 16) – Incoming UPT Call procedures



T1183390-96

Figure 14/Q.1551 (sheet 11 of 16) – Incoming UPT Call procedures

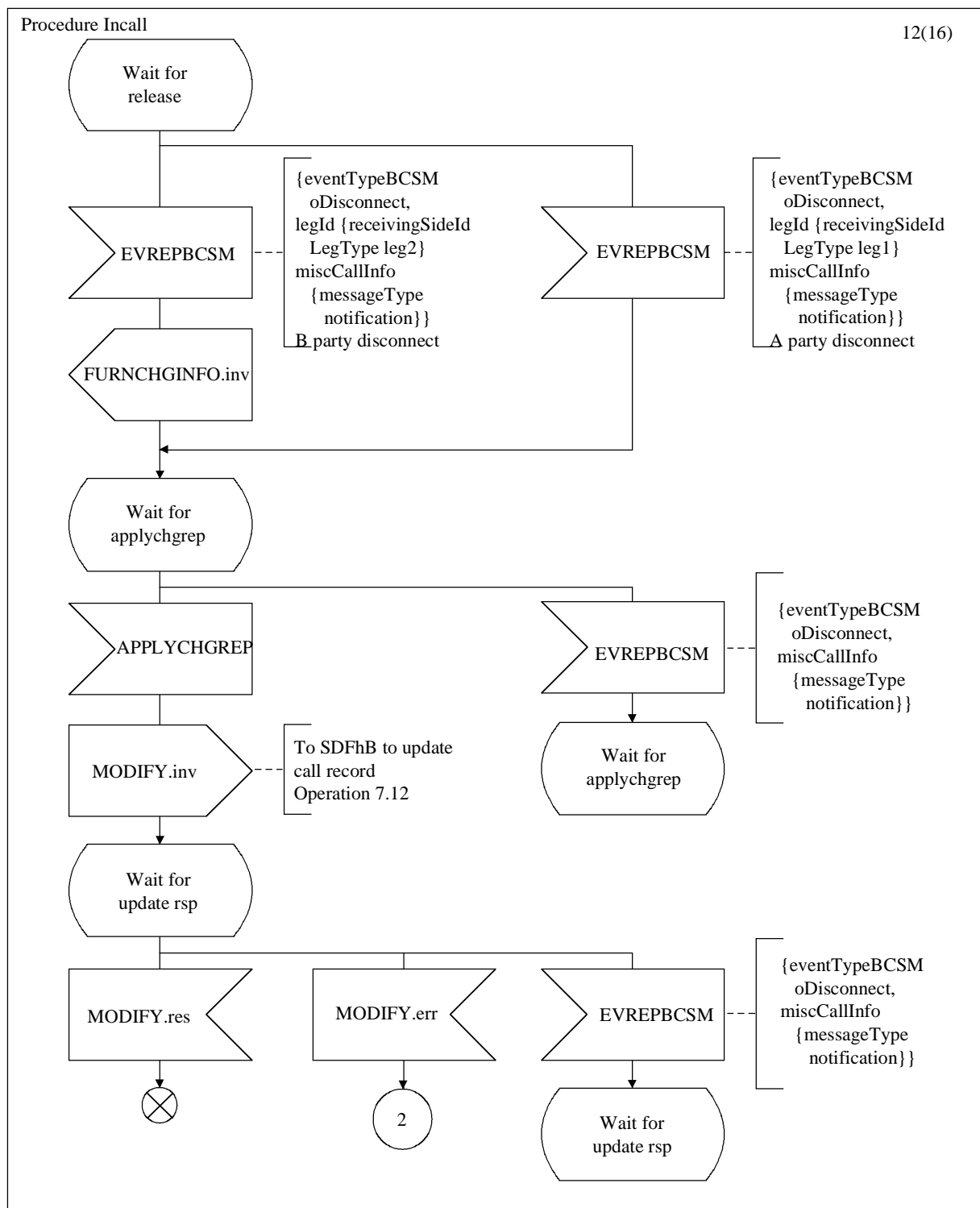
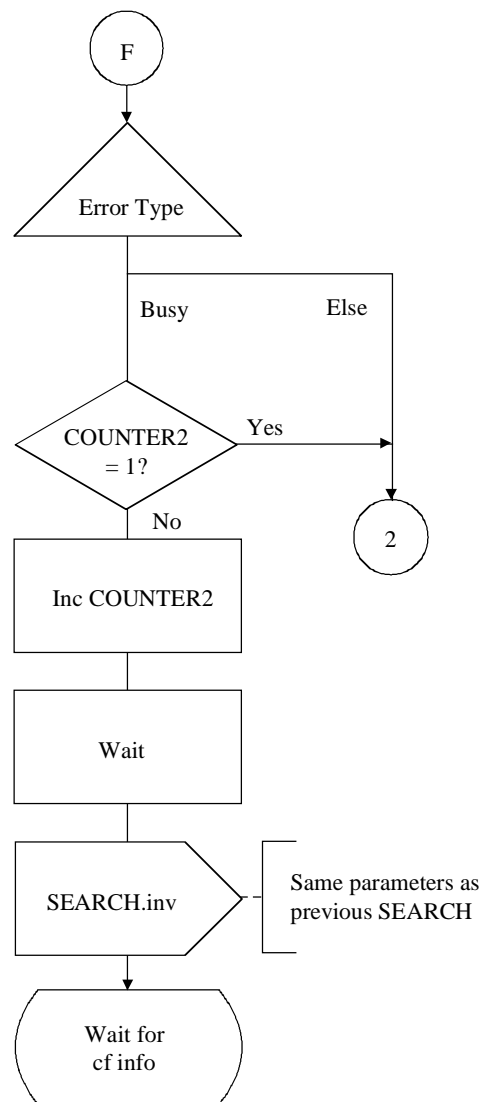
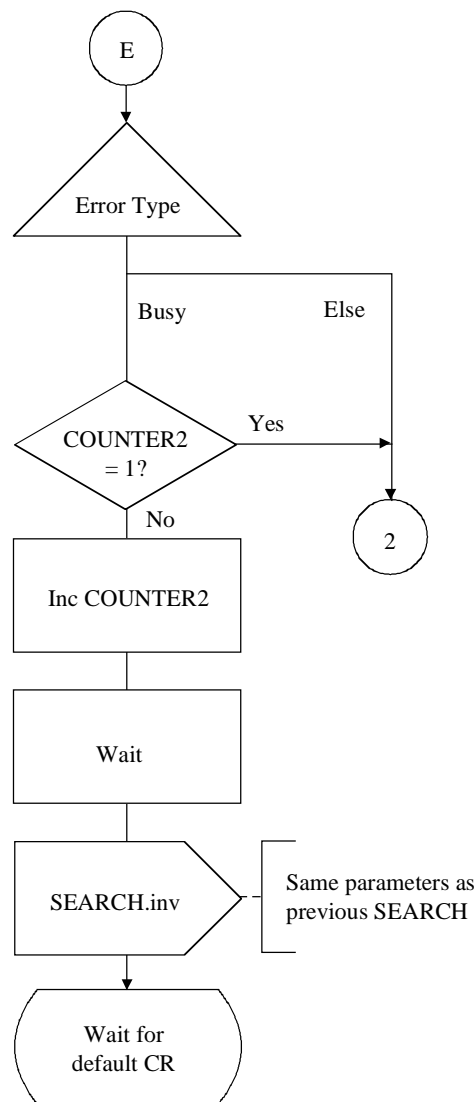


Figure 14/Q.1551 (sheet 12 of 16) – Incoming UPT Call procedures



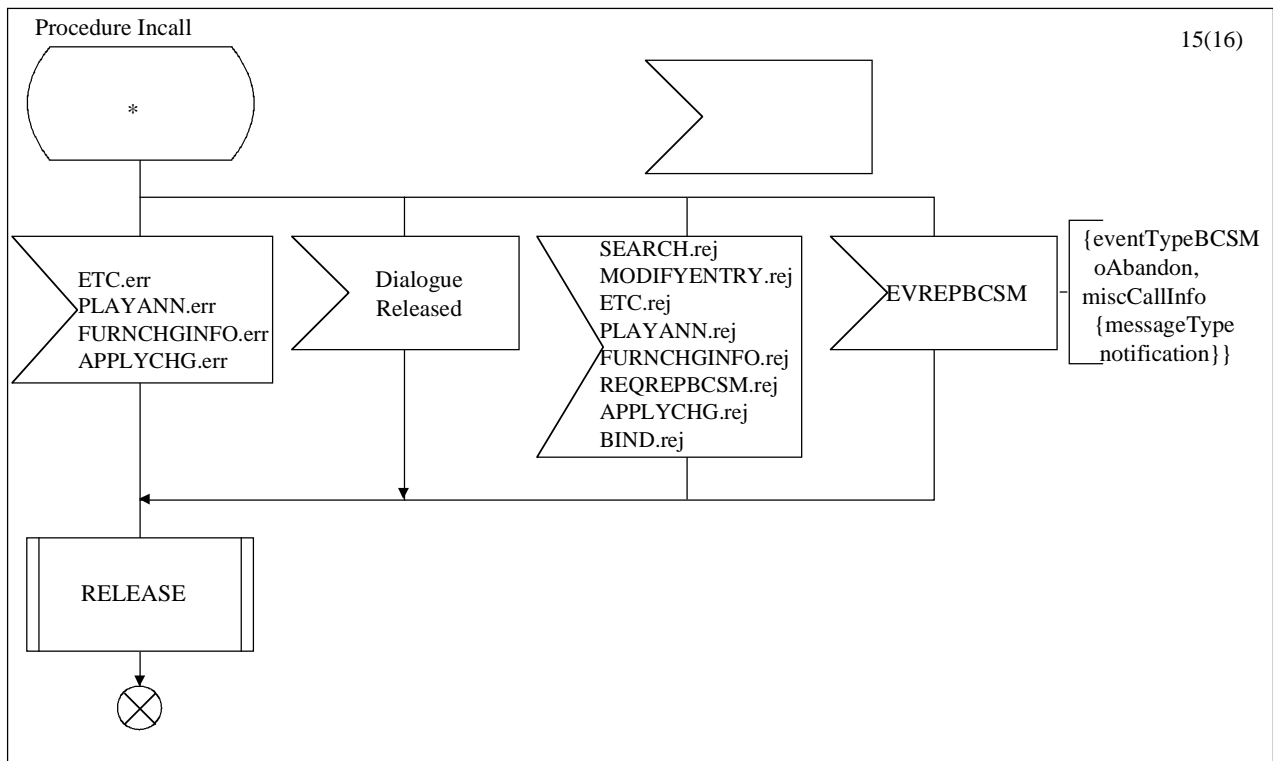
T1183410-96

Figure 14/Q.1551 (sheet 13 of 16) – Incoming UPT Call procedures



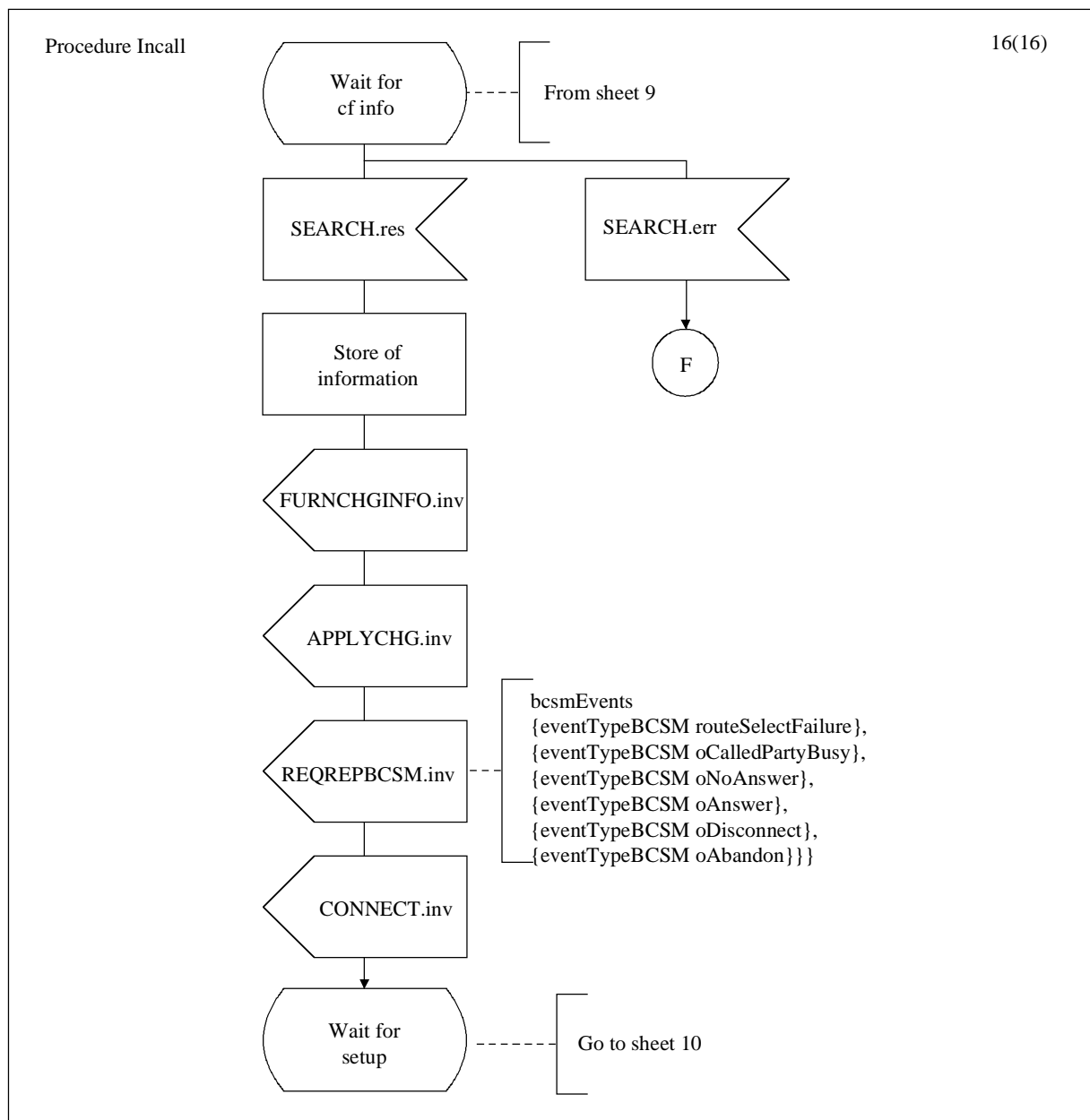
T1183420-96

Figure 14/Q.1551 (sheet 14 of 16) – Incoming UPT Call procedures



T1183430-96

Figure 14/Q.1551 (sheet 15 of 16) – Incoming UPT Call procedures



T1183440-96

Figure 14/Q.1551 (sheet 16 of 16) – Incoming UPT Call procedures

6.5 Service profile management procedure

There are two procedures in service profile management: the Service Profile Interrogation (SPI) procedure, and the Service Profile Modification (SPM) procedure.

The IA procedure must have been successfully completed before the start of these procedures.

6.5.1 Service profile interrogation procedure

6.5.1.1 General

The Service Profile Interrogation procedure is used when the UPT user wants to read the following information in his own service profile:

- default registration address;

- registration address;
- registration expiration time;
- routing address for supplementary services;
- activation status of the supplementary services;
- no reply condition timer.

6.5.1.2 Detailed procedure

Figure 15 shows the Service_Profile_Interrogation (SPI) procedure.

The SPI procedure is invoked by the UPT_SLP process when the UPT user answers with the Service Profile Interrogation Code to the FRI procedure.

The SCF formulates and sends a P&C request indication. On receipt of the P&C, the SRF requests from the user what information the user wants to view. It should be noted that the user can only request a piece of information at a time.

The following events move the SCF out of "Wait for Service Profile Code ACK." state:

- the user has abandoned the procedure: the SCF receives an EVREPBSCM indicating the user release. The procedure is terminated with Exit = NOK. This is included in the "State *" of SDL diagrams;
- an error has occurred for the P&C operation (P&C.err):
 - if the error is "ImproperCallerResponse", the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries (COUNTER1) is incremented and, if not exceeded, the prompt is repeated to the user that is allowed to make another attempt. If the number of retries (COUNTER1) is exceeded the processing continues as described in subclause "Maximum number of retries exceeded";
 - else for any other error the procedure is terminated with Exit = NOK.
- the P&C operation is rejected (P&C.rej): the procedure is terminated with Exit = NOK;
- if the information is invalid, the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries (COUNTER1) is incremented and, if not exceeded, the SCF sends a P&C allowing the user to have a view of another information in his profile. If the number of retries is exceeded the processing as described in subclause "Maximum number of retries exceeded".

Retrieve service profile

Once collected, the code provided by the user is used to generate a SEARCH operation to the SDFh. This operation retrieves the requested piece of information from the user's service profile (Wait for Search Result). The four possible outcomes are:

- dialogue released by IN node (Dialogue_released): the reasons for releasing the dialogue prematurely are described in 6.1. Regardless of the reason for release, the call handling procedure is terminated and the SSF is instructed to release the call;
- user abandon: the SCF is notified of the release by the EVREPBSCM operation from the SSF. The SCF will terminate the SPI procedure;
- operation error returned (SEARCH.err): the possible error causes returned in the SEARCH operation (.err) are described in Recommendation Q.1218 [1]:
 - in the case of Busy in 'Service Error' type the SCF can retry the same SDF after a time delay and for a limited number of times (COUNTER2). If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed;

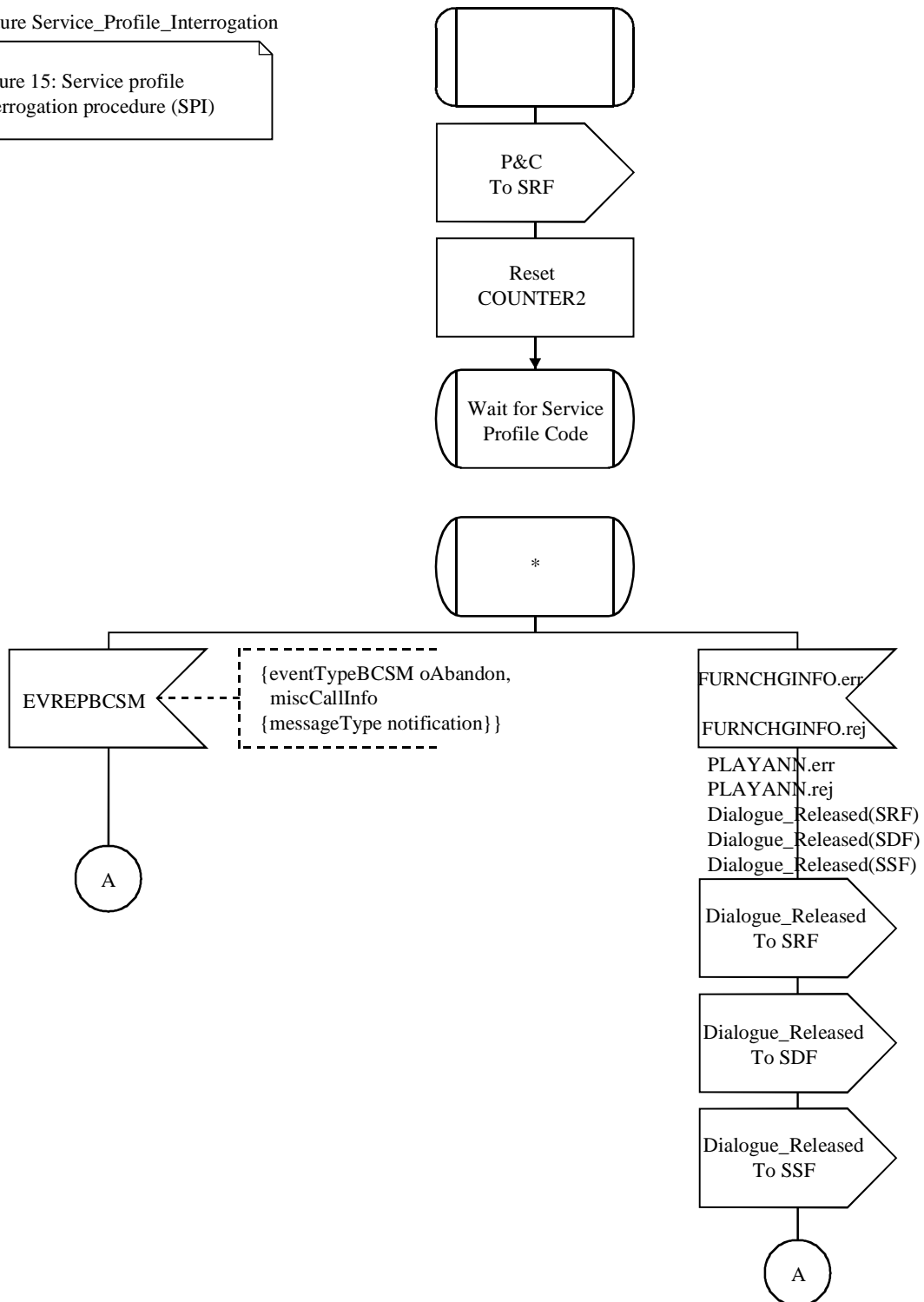
- for all other error types returned by the SDF, the SCF sends a P&C request indication in order to prompt the UPT user to ask him to terminate or request another procedure (this is done through the FRI procedure).
- successful result (SEARCH.rsp): the interrogation of the service profile is accepted. The SCF sends a PLAYANN in order to give the requested information to the UPT user. Afterwards the SCF sends FURNCHGINFO to the SSF, then the SCF sends a P&C request indication in order to prompt the UPT user to ask him to terminate or request another procedure (this is done through the FRI procedure).

Maximum number of retries exceeded

If the maximum number of retries (COUNTER1) is exceeded a PLAYANN is played to the user. If the user hangs up, the RELEASE procedure is executed. Otherwise the procedure is terminated with Exit = NOK.

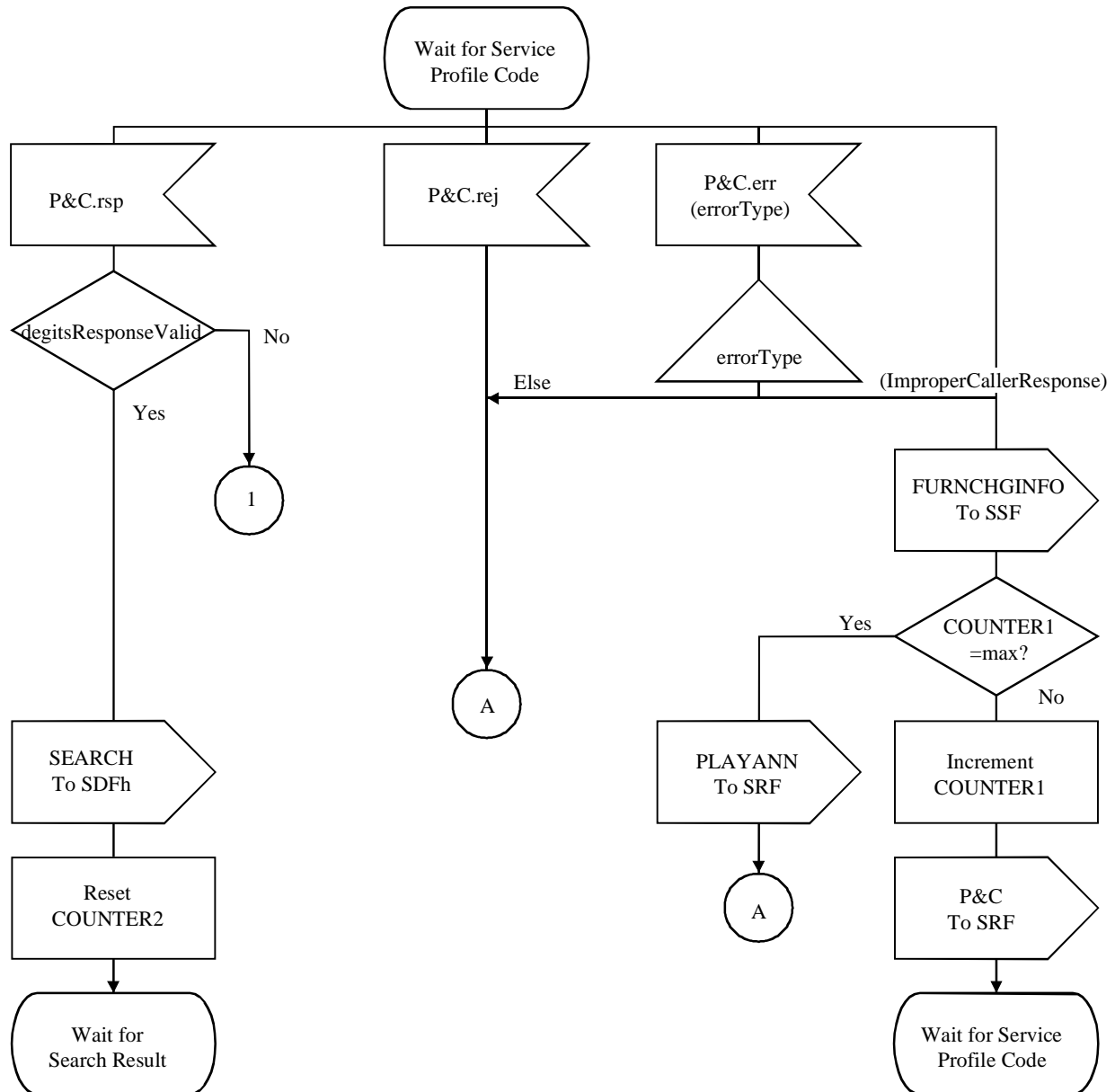
Figure 15: Service profile interrogation procedure (SPI)

1(5)



T1180710-96

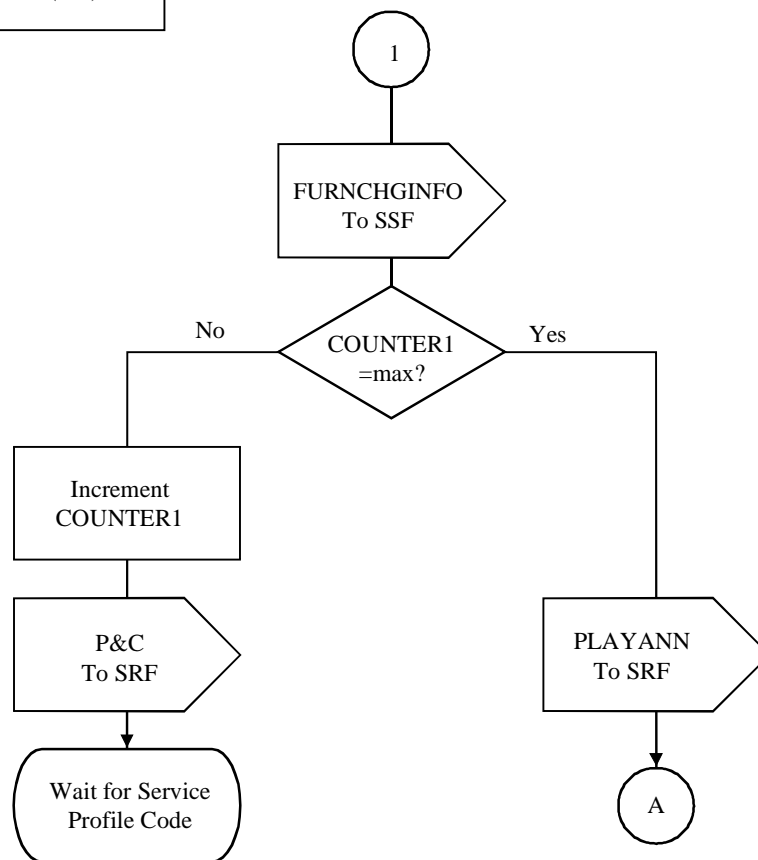
Figure 15/Q.1551 (sheet 1 of 5) – Service Profile Interrogation (SPI) procedure

Figure 15: Service profile
interrogation procedure (SPI)

T1180720-96

Figure 15/Q.1551 (sheet 2 of 5) – Service Profile Interrogation (SPI) procedure

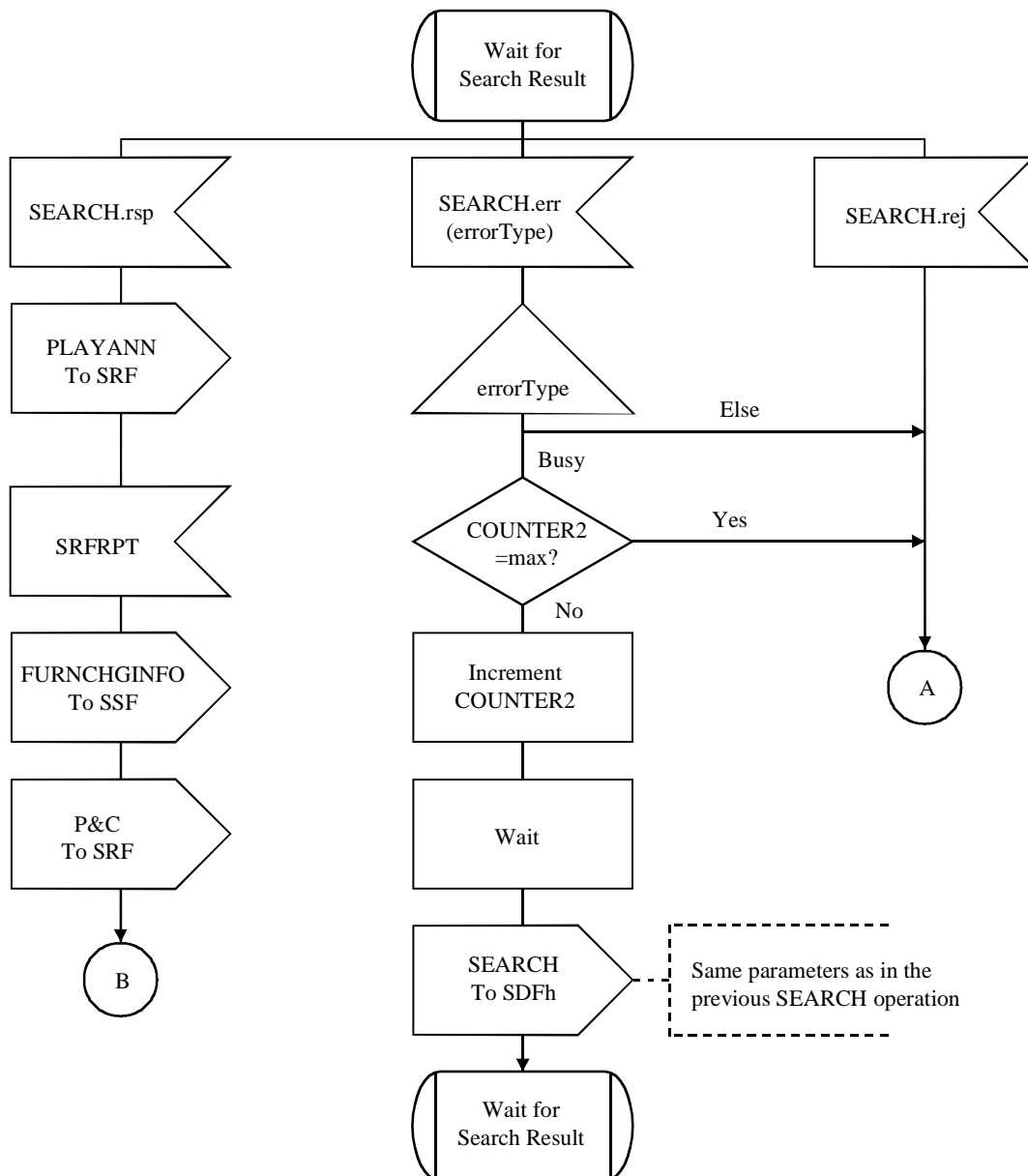
Figure 15: Service profile
interrogation procedure (SPI)



T1180730-96

Figure 15/Q.1551 (sheet 3 of 5) – Service Profile Interrogation (SPI) procedure

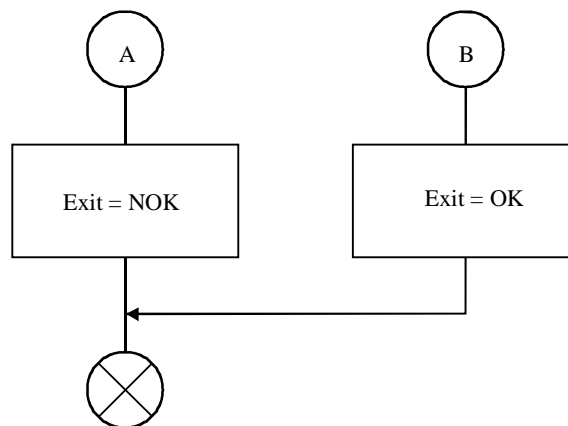
Figure 15: Service profile
interrogation procedure (SPI)



T1180740-96

Figure 15/Q.1551 (sheet 4 of 5) – Service Profile Interrogation (SPI) procedure

Figure 15: Service profile interrogation procedure (SPI)



T1180750-96

Figure 15/Q.1551 (sheet 5 of 5) – Service Profile Interrogation (SPI) procedure

6.5.2 Service profile modification procedure

6.5.2.1 General

The service profile modification procedure is used when the UPT user wants to change his own service profile. The information the user is able to modify is the same as in SPI procedure.

6.5.2.2 Detailed procedure

Figure 16 shows the Service_Profile_Modification (SPM) procedure.

The SPM procedure is invoked by the SCF_Behavior process when the UPT user answers with the Service Profile Modification Code and during the FRI procedure.

The SCF formulates and sends a P&C request indication. On receipt of the P&C, the SRF gives to the user the value of the requested information and asks him if he wants to modify it. It should be noted that the user can only change one piece of information at a time.

The following events move the SCF out of "Wait for user information to modify" state:

- the user has abandoned the procedure: the SCF receives an EVREPBCSM indicating the user release. The procedure is terminated with Exit = NOK. This is included in the "State *" of SDL diagrams;
- an error has occurred for the P&C operation (P&C.err):
 - if the error is "ImproperCallerResponse", the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries (COUNTER1) is incremented and, if not exceeded, the prompt is repeated to the user that is allowed to request the service profile code and the contents of the profile the user wants to modify. If the number of retries (COUNTER1) is exceeded the processing continues as described in subclause "Maximum number of retries exceeded";
 - else for any other error the procedure is terminated with Exit = NOK.
- the P&C operation is rejected (P&C.rej): the procedure is terminated with Exit = NOK;
- if the user's answer is valid, the processing continues as described in subclause "User acknowledgment";
- if the information that the user answers to the prompt is invalid, the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request, the number of retries (COUNTER1) is incremented and, if not exceeded, the SCF sends a P&C allowing the user to request another type of information in his profile to be modified. If the number of retries (COUNTER1) is exceeded, the processing as described in subclause "Maximum number of retries exceeded".

User acknowledgment

With the information provided by the user, the SCF sends a P&C in order to get the user acknowledgment (Wait for user acknowledgment).

- the user has abandoned or the P&C operation error/reject has occurred, the action of the SCF is the same as in the case of "Wait for user information to modify" state;
- when the user answers "yes", the SCF sends the PLAYANN in order to notify to the user that the SPM procedure is being processed, and the processing continues as described in subclause "Change service profile";
- when the user answers "no" to the prompt, the SCF sends a FURNCHGINFO to request the SSF to create a record for the user request. The SCF sends a P&C allowing the user to return to the Feature Request Identification (FRI) (follow-on);
- when the user requests another feature to the prompt, the SCF sends a P&C request indication in order to prompt the UPT user that he can terminate or request another procedure (this is done through the FRI procedure).

Change service profile

If the user accepts the service profile modification, the SCF sends a MODIFY operation to SDFh in order to modify to the service profile. The SCF waits for the SDFh answer (state "Wait for Update Confirmation").

The following events move the SCF out of this state:

- dialogue released by IN node (Dialogue_released): the reasons for releasing the dialogue prematurely are described in 6.1. Regardless of the reason for release, the call handling procedure is terminated and the SSF is instructed to release the call;
- user abandon: the SCF is notified of the release by the EVREPBSCSM operation from the SSF. The SCF will terminate the SPM procedure;
- the MODIFY operation is successful: the modification to the service profile is accepted. Afterwards the SCF sends a P&C request indication in order to prompt the UPT user that the modification to his service profile has been successfully completed. He is further asked to either terminate or request another procedure (this is done through the FRI procedure);
- operation errors are returned (MODIFY.err): the possible error causes returned in the MODIFY operation (.err) are described in Recommendation Q.1218 [1]:
 - if the error is "Constraint Violation" the modification to the service profile is denied. The SCF sends FURNCHGINFO to the SSF to record the user request, the number of retries is incremented (COUNTER1). If the maximum number of retries is not exceeded, the SCF sends a P&C in order to inform the user that the request cannot be handled and that he is given the option of terminating or requesting another procedure (this is done through the FRI procedure). If the maximum number is exceeded, the processing continues as described in subclause "Maximum number of retries exceeded";
 - in the case of 'Service Error' type Busy the SCF can retry the same SDF after a time delay and for a limited number of times (COUNTER2). If the number of retries (COUNTER2) is exceeded, the RELEASE procedure is performed;
 - for all other error types returned by the SDF, the procedure is terminated with Exit = NOK.
- the MODIFY operation is rejected (MODIFY.rej) the procedure is terminated with Exit = NOK.

Maximum number of retries exceeded

If the maximum number of retries (COUNTER1) is exceeded a PLAYANN is played to the user. If the user hangs up, the RELEASE procedure is executed. Otherwise the procedure is terminated with Exit = NOK.

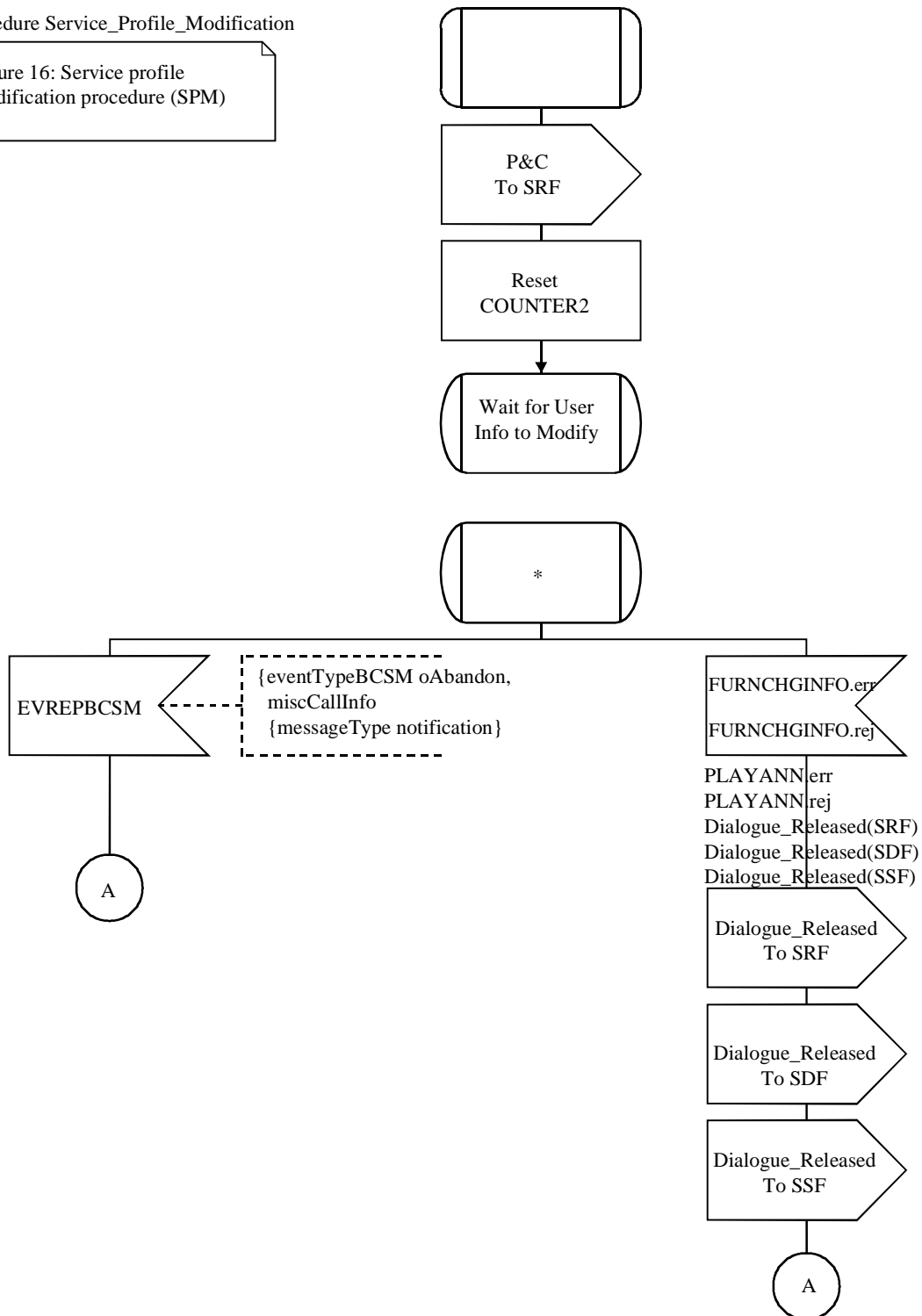
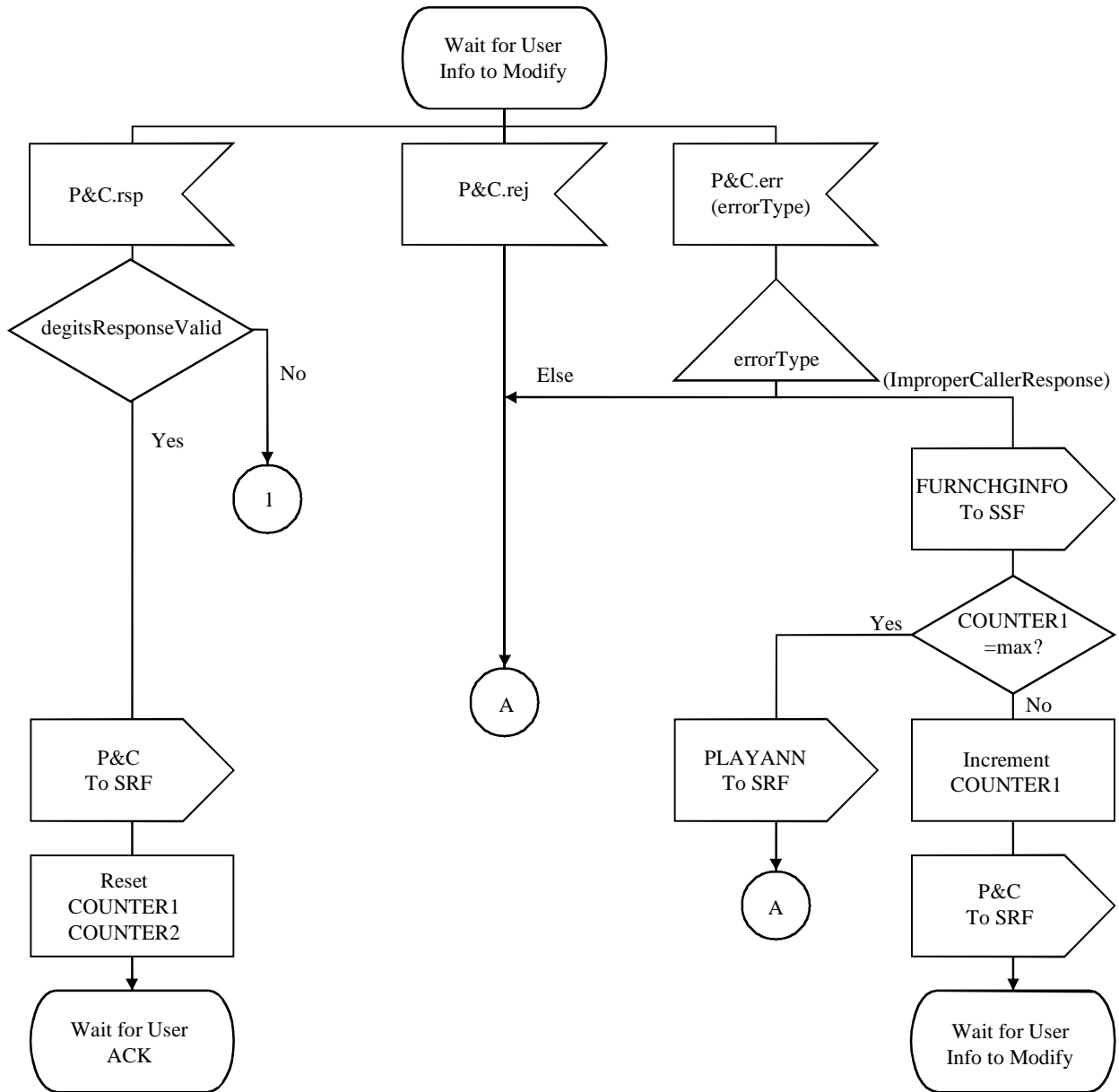


Figure 16/Q.1551 (sheet 1 of 6) – Service Profile Modification (SPM) procedure

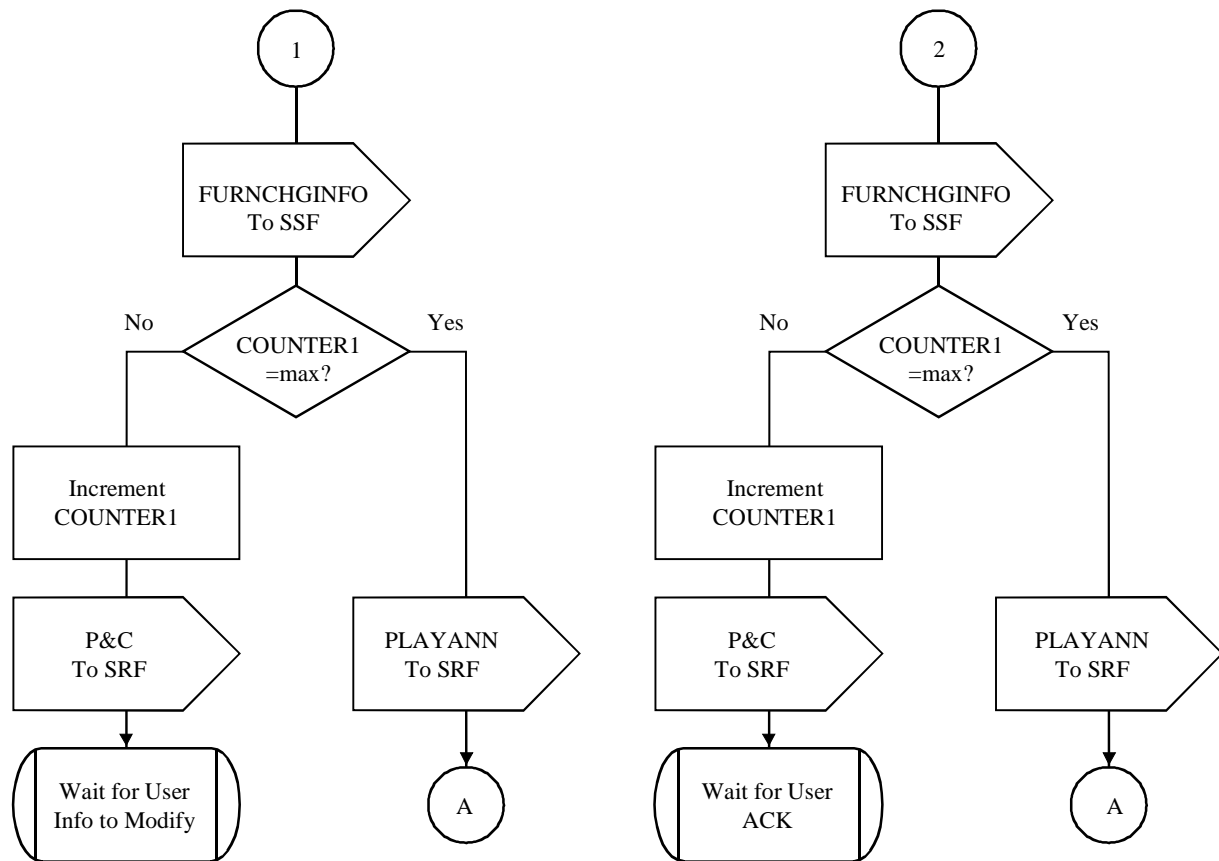
Figure 16: Service profile modification procedure (SPM)



T1180770-96

Figure 16/Q.1551 (sheet 2 of 6) – Service Profile Modification (SPM) procedure

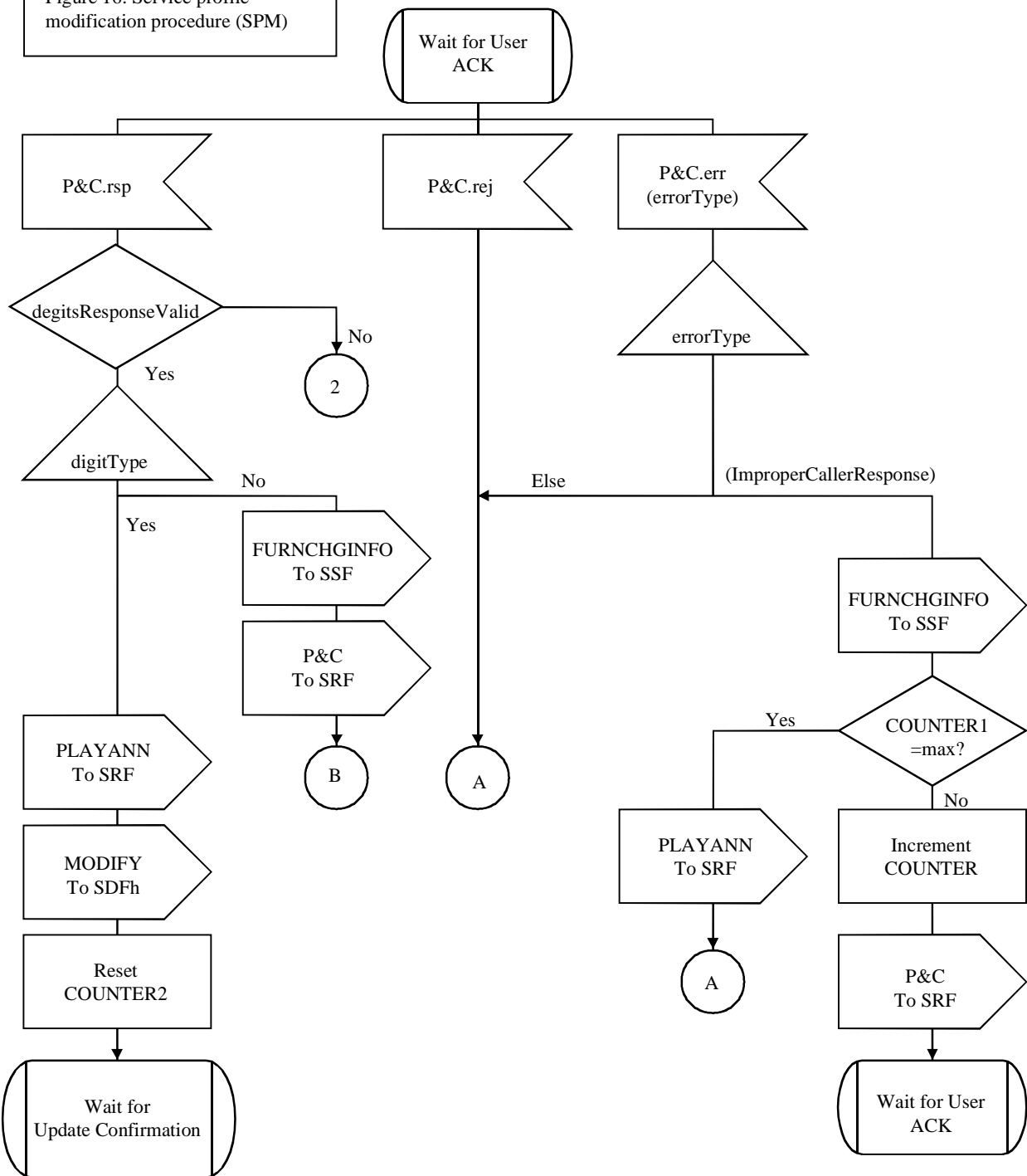
Figure 16: Service profile modification procedure (SPM)



T1180780-96

Figure 16/Q.1551 (sheet 3 of 6) – Service Profile Modification (SPM) procedure

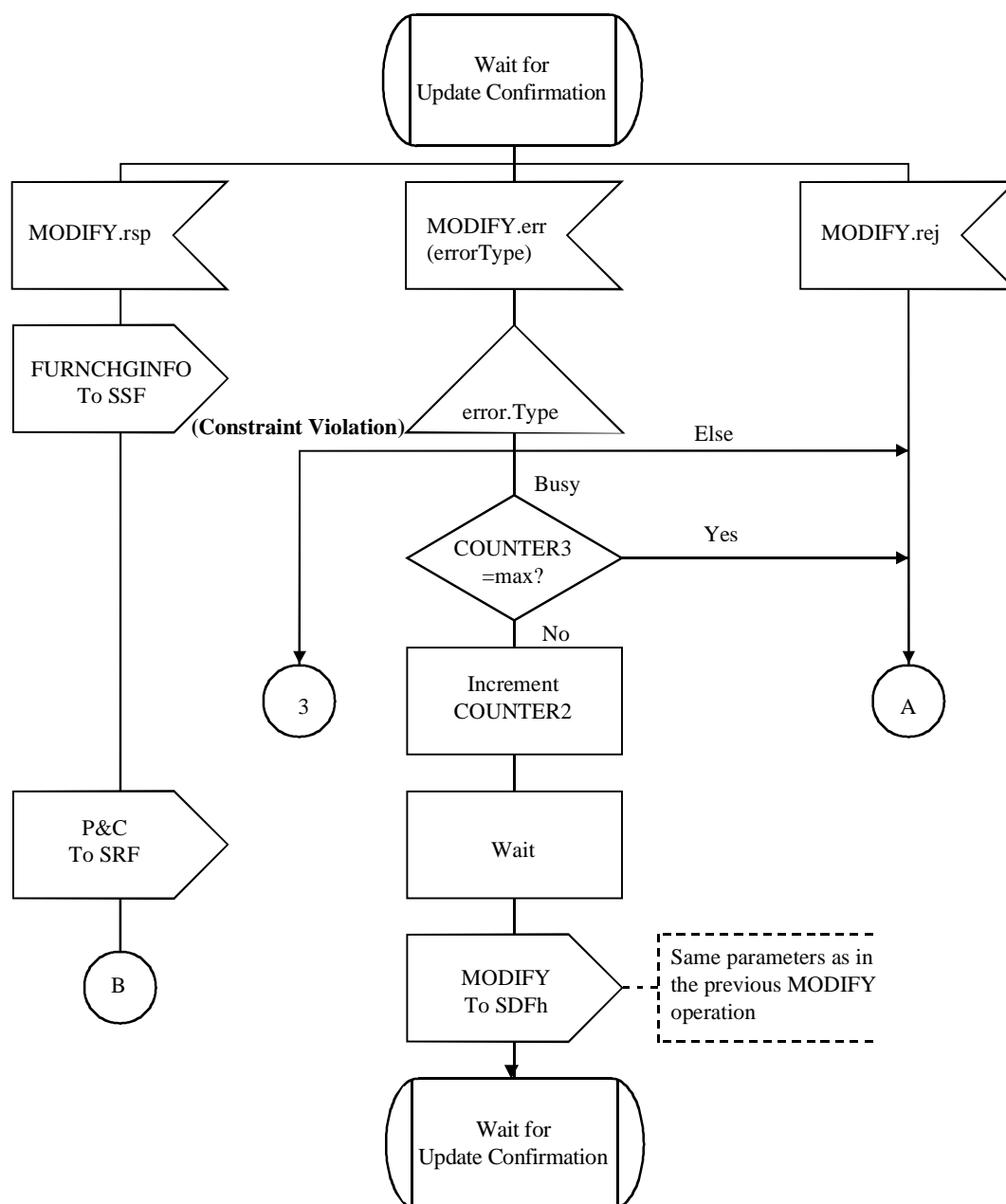
Figure 16: Service profile modification procedure (SPM)



T1180790-96

Figure 16/Q.1551 (sheet 4 of 6) – Service Profile Modification (SPM) procedure

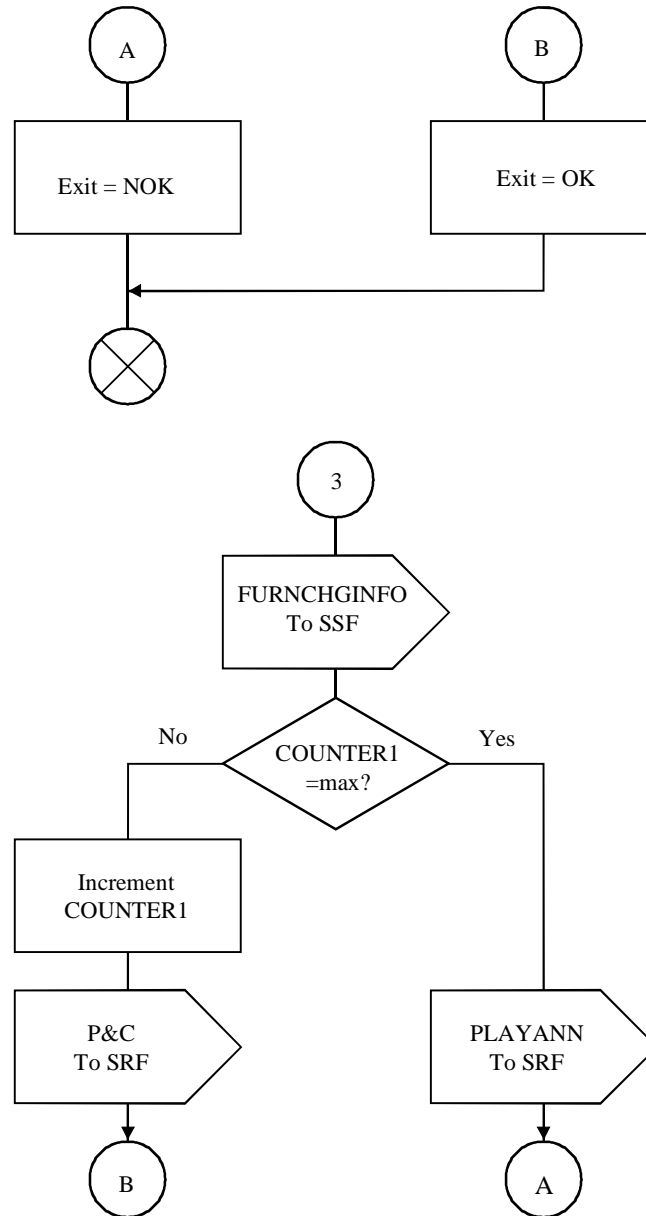
Figure 16: Service profile modification procedure (SPM)



T1180800-96

Figure 16/Q.1551 (sheet 5 of 6) – Service Profile Modification (SPM) procedure

Figure 16: Service profile modification procedure (SPM)



T1180810-96

Figure 16/Q.1551 (sheet 6 of 6) – Service Profile Modification (SPM) procedure

ANNEX A

This Annex contains an ASN.1 module that describes the UPT information model.

UPT-DataModel

-- this modules contains the ASN.1 Information Object Notation
-- for defining the contents of an SDF for the UPT service.

DEFINITIONS ::=

BEGIN

IMPORTS

informationFramework, upperBounds

FROM UsefulDefinitions {joint-iso-ccitt ds (5) module (1) usefulDefinitions (0)}

OBJECT-CLASS, ATTRIBUTE, NAME-FORM, STRUCTURE-RULE, alias

FROM InformationFramework informationFramework

organization, organizationalUnit, countryNameForm

FROM SelectedObjectClasses {joint-iso-ccitt ds (5) module (1) selectedObjectClasses (6) 2}

description, integerMatch, numericStringMatch, commonName, surname, octetStringMatch,
integerOrderingMatch, name, seeAlso, organizationalUnitName, reversePrefixMatch

FROM SelectedAttributeTypes {joint-iso-ccitt ds (5) module (1) selectedAttributeTypes (5) 2}

userPassword

FROM AuthenticationFramework {joint-iso-ccitt ds (5) module (1) authenticationFramework (7) 2}

ub-international-isdn-number

FROM UpperBounds upperBounds;

uptProvider OBJECT-CLASS ::= {

SUBCLASS OF {organization}

MUST CONTAIN {providerId}

MAY CONTAIN {description}

ID id-oc-uptProvider}

providerId ATTRIBUTE ::= {

WITH SYNTAX AddressString {ub-providerId}

EQUALITY MATCHING RULE numericStringMatch

SUBSTRINGS MATCHING RULE reversePrefixMatch

SINGLE VALUE TRUE

ID id-at-providerId}

ub-providerId INTEGER ::= 3

AddressString {INTEGER: ub-max-value} ::= NumericString (SIZE (1..ub-max-value))

partner OBJECT-CLASS ::= {

MUST CONTAIN {providerId}

MAY CONTAIN {description}

ID id-oc-partner}

agreedService OBJECT-CLASS ::= {

MUST CONTAIN {providedServiceId}

MAY CONTAIN {providedLocations|
description}

ID id-oc-agreement}


```

providedServiceId ATTRIBUTE ::= {
    WITH SYNTAX                               Service
    EQUALITY MATCHING RULE                     integerMatch
    SINGLE VALUE                               TRUE
    ID                                         id-at-providedServiceId}

Service ::= INTEGER {
--basic services 0-9
    isdnTelephony (0),
-- registration service 10-19
    icRegistration (10),
--profile service 20-29
    serviceProfileModification (20),
-- charging service 30-39
    standard (30),
-- routing service 40-49
    callForwardingUnconditional (40),
    callForwardingOnNoReply (41),
    callForwardingOnBusy (42),
    variableRoutingOnTime (43),
    variableRoutingOnCallingLine (44)}

providedLocations ATTRIBUTE ::= {
    WITH SYNTAX                               AddressString{ub-international-isdn-number}
    EQUALITY MATCHING RULE                     numericStringMatch
    SUBSTRINGS MATCHING RULE                   reversePrefixMatch
    ID                                         id-at-providedLocations}

administrativeUnit OBJECT-CLASS ::= {
    SUBCLASS OF                               {organizationalUnit}
    ID                                         id-oc-administrativeUnit}

userProfile OBJECT-CLASS ::= {
    MUST CONTAIN                               {pui}
    MAY CONTAIN                               {description|
                                                commonName|
                                                surname|
                                                seeAlso|
                                                userCredit|
                                                userPassword|      --as defined in Recommendation X.509
                                                specialPassword|
                                                variablePassword|
                                                allowedServiceFeatures|
                                                callInfoRecords|
                                                activeChargingService|
                                                nbOfFailedAuthentications}
    ID                                         id-oc-userProfile}

pui ATTRIBUTE ::= {
    WITH SYNTAX                               AddressString{ub-pui}
    EQUALITY MATCHING RULE                     numericStringMatch
    SINGLE VALUE                               TRUE
    ID                                         id-at-pui}

ub-pui INTEGER ::= 15

specialPassword ATTRIBUTE ::= {
    WITH SYNTAX                               OCTET STRING (SIZE (0..ub-special-password))
    EQUALITY MATCHING RULE                     octetStringMatch
    ID                                         id-at-specialPassword}

```

```
variablePassword ATTRIBUTE ::= {
    WITH SYNTAX
    EQUALITY MATCHING RULE
    ID
    OCTET STRING (SIZE (0..ub-variable-password))
    octetStringMatch
    id-at-variablePassword}
```

nbOffFailedAuthentications	ATTRIBUTE ::= {
WITH SYNTAX	INTEGER (1..ub-max-nbOffFailedAuthentications)
ORDERING MATCHING RULE	integerOrderingMatch
SINGLE VALUE	TRUE
ID	id-at-nbOffFailedAuthentications}

userCredit ATTRIBUTE ::= {	
WITH SYNTAX	INTEGER (1..ub-maxUserCredit)
ORDERING MATCHING RULE	integerOrderingMatch
SINGLE VALUE	TRUE
ID	id-at-userCredit}

```
callInfoRecords ATTRIBUTE ::= {
    WITH SYNTAX      CallInfoRecord
    ID                id-at-callInfoRecords}
```

```
Cost ::= CHOICE {
    pulse  [0] INTEGER (1..ub-pulse),
    cost   [1] CurrencyValue}
```

```

CurrencyValue::=CHOICE {
    usDollar           [0] Currency,
    frenchFranc        [1] Currency,
    germanMark         [2] Currency,
    dutchGuilder       [3] Currency,
    italianLira        [4] Currency,
    englishPound       [5] Currency,
    spanishPeseta      [6] Currency,
    swedishKrone       [7] Currency,
    norwegianKrone     [8] Currency,
    japaneseYen        [9] Currency}

```

Currency ::= REAL
CFServices ::= SET OF Service (40..49)
Cause ::= OCTET STRING (SIZE(minCauseLength..maxCauseLength))
minCauseLength INTEGER ::= 2
maxCauseLength INTEGER ::= 20

activeChargingService ATTRIBUTE ::= {
 WITH SYNTAX Service (30..39)
 EQUALITY MATCHING RULE integerMatch
 SINGLE VALUE TRUE
 ID id-at-activeChargingService}
allowedServiceFeatures ATTRIBUTE ::= {
 WITH SYNTAX Service
 EQUALITY MATCHING RULE integerMatch
 ID id-at-allowedServiceFeatures}
userProfileAlias OBJECT-CLASS ::= {
 SUBCLASS OF {alias}
 MUST CONTAIN {uptNumber}
 MAY CONTAIN {description}
 ID id-oc-userProfileAlias}
uptNumber ATTRIBUTE ::= {
 WITH SYNTAX IsdnAddress
 EQUALITY MATCHING RULE numericStringMatch
 SUBSTRINGS MATCHING RULE reversePrefixMatch
 SINGLE VALUE TRUE
 ID id-at-uptNumber}
calledUptUser OBJECT-CLASS ::= {
 MUST CONTAIN {name}
 MAY CONTAIN {description|
 icRegistrationAddress|
 allowedRegistrationAddress|
 defaultChargingReference}
 ID id-oc-calledUptUser}
defaultChargingReference ATTRIBUTE ::= {
 WITH SYNTAX IsdnAddress
 EQUALITY MATCHING RULE numericStringMatch
 SUBSTRINGS MATCHING RULE reversePrefixMatch
 SINGLE VALUE TRUE
 ID id-at-defaultChargingReference}
IsdnAddress ::= AddressString{ub-international-isdn-number}
icRegistrationAddress ATTRIBUTE ::= {
 WITH SYNTAX IsdnAddress
 EQUALITY MATCHING RULE numericStringMatch
 SUBSTRINGS MATCHING RULE reversePrefixMatch
 ID id-at-icRegistrationAddress}
allowedRegistrationAddress ATTRIBUTE ::= {
 WITH SYNTAX AddressString{ub-international-isdn-number}
 EQUALITY MATCHING RULE numericStringMatch
 SUBSTRINGS MATCHING RULE reversePrefixMatch
 ID id-at-allowedRegistrationAddress}

```

callingUptUser OBJECT-CLASS ::= {
    MUST CONTAIN {name|
                  allowedDestinations
                  -- callRecords??--}
    MAY CONTAIN {description}
    ID          id-oc-callingUptUser}

allowedDestinations ATTRIBUTE ::= {
    WITH SYNTAX      AddressString{ub-international-isdn-number}
    EQUALITY MATCHING RULE      numericStringMatch
    SUBSTRINGS MATCHING RULE    reversePrefixMatch
    ID          id-at-allowedDestinations}

supplementaryService OBJECT-CLASS ::= {
    KIND          abstract
    MUST CONTAIN {supplServId|
                  supplServiceStatus}
    MAY CONTAIN {name|
                  description}
    ID          id-ao-supplementaryService}

supplServId ATTRIBUTE ::= {
    WITH SYNTAX      SS-Code
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE      TRUE
    ID          id-at-supplServId}

supplServiceStatus ATTRIBUTE ::= {
    WITH SYNTAX      SupplServiceStatus
    SINGLE VALUE      TRUE
    ID          id-at-supplServiceStatus}

SupplServiceStatus ::= BIT STRING {
    provisioned (0),
    registered (1),
    activated (2)}

SS-Code ::= INTEGER {
    clip (11),
    clir (12),
    colp (13),
    colr (14),
    mci (15),
    cfu (16),
    cfb (17),
    cfna (18),
    cfnr (25),
    civr (26),
    tvr (27)}

callForwarding OBJECT-CLASS ::= {
    SUBCLASS OF      {supplementaryService}
    MUST CONTAIN {forwardedToNumber|
                  typesOfNotification}
    MAY CONTAIN {noReplyConditionTimer}
    ID          id-oc-callForwarding}

```

```

forwardedToNumber ATTRIBUTE ::= {
    WITH SYNTAX          IsdnAddress
    EQUALITY MATCHING RULE    numericStringMatch
    SUBSTRINGS MATCHING RULE    reversePrefixMatch
    ID                    id-at-forwardedToNumber}

typesOfNotification ATTRIBUTE ::= {
    WITH SYNTAX    TypesOfNotification
    SINGLE VALUE   TRUE
    ID            id-at-typesOfNotification}

TypesOfNotification ::= BIT STRING {
    servedUserForwardedCall (0),
    callingUserWithForwardedToNumber (1),
    callingUserWithoutForwardedToNumber (2),
    servedUserForwardingActivation (3)}

noReplyConditionTimer ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER (1..ub-noReplyConditionTimer)
    EQUALITY MATCHING RULE    integerMatch
    SINGLE VALUE           TRUE
    ID                    id-at-noReplyConditionTimer}

ub-noReplyConditionTimer INTEGER ::= 10000

-- name forms

uptProviderNameForm NAME-FORM ::= {
    NAMES          uptProvider
    WITH ATTRIBUTES    {providerId}
    ID            id-nf-uptProviderNameForm}

partnerNameForm NAME-FORM ::= {
    NAMES          partner
    WITH ATTRIBUTES    {providerId}
    ID            id-nf-partnerNameForm}

adminUnitNameForm NAME-FORM ::= {
    NAMES          administrativeUnit
    WITH ATTRIBUTES    {organizationalUnitName}
    ID            id-nf-adminUnitNameForm}

agreedServiceNameForm NAME-FORM ::= {
    NAMES          agreedService
    WITH ATTRIBUTES    {providedServiceId}
    ID            id-nf-agreedServiceNameForm}

userProfileNameForm NAME-FORM ::= {
    NAMES          userProfile
    WITH ATTRIBUTES    {pui}
    ID            id-nf-userProfileNameForm}

userProfileAliasNameForm NAME-FORM ::= {
    NAMES          userProfileAlias
    WITH ATTRIBUTES    {uptNumber}
    ID            id-nf-userProfileAliasNameForm}

```

```

calledUptUserNameForm NAME-FORM ::= {
    NAMES          calledUptUser
    WITH ATTRIBUTES {name}
    ID              id-nf-calledUptUserNameForm}

```

```

callingUptUserNameForm NAME-FORM ::= {
    NAMES          callingUptUser
    WITH ATTRIBUTES {name}
    ID              id-nf-callingUptUserNameForm}

```

```

callForwardingNameForm NAME-FORM ::= {
    NAMES          callForwarding
    WITH ATTRIBUTES {supplServId}
    ID              id-nf-callForwardingNameForm}

```

-- structure rules

```

sr0 STRUCTURE-RULE ::= {
    NAME FORM      countryNameForm
    ID              0}

```

```

sr1 STRUCTURE-RULE ::= {
    NAME FORM      uptProviderNameForm
    SUPERIOR RULES {sr0}
    ID              1}

```

```

sr2 STRUCTURE-RULE ::= {
    NAME FORM      adminUnitNameForm
    SUPERIOR RULES {sr1}
    ID              2}

```

```

sr3 STRUCTURE-RULE ::= {
    NAME FORM      partnerNameForm
    SUPERIOR RULES {sr1}
    ID              3}

```

```

sr4 STRUCTURE-RULE ::= {
    NAME FORM      userProfileNameForm
    SUPERIOR RULES {sr2}
    ID              4}

```

```

sr5 STRUCTURE-RULE ::= {
    NAME FORM      userProfileAliasNameForm
    SUPERIOR RULES {sr2}
    ID              5}

```

```

sr6 STRUCTURE-RULE ::= {
    NAME FORM      agreedServiceNameForm
    SUPERIOR RULES {sr3}
    ID              6}

```

```

sr7 STRUCTURE-RULE ::= {
    NAME FORM      calledUptUserNameForm
    SUPERIOR RULES {sr4}
    ID              7}

```

```

sr8 STRUCTURE-RULE ::= {
    NAME FORM      callingUptUserNameForm
    SUPERIOR RULES {sr4}
    ID              8}

```

```

sr9 STRUCTURE-RULE::= {
    NAME FORM                callForwardingNameForm
    SUPERIOR RULES          {sr7}
    ID                        9}

-- object-identifiers assignment
id-oc OBJECT IDENTIFIER ::= {ccitt recommendation q 1551 --upt-- module (0) uptUsefulDefinitions (0)
    objectClasses (0) version1 (0)}
id-ao OBJECT IDENTIFIER ::= {ccitt recommendation q 1551 --upt-- module (0) uptUsefulDefinitions
    (0)abstractObjectClasses (1) version1 (0)}
id-at OBJECT IDENTIFIER ::= {ccitt recommendation q 1551 --upt-- module (0) uptUsefulDefinitions (0)
    attributeTypes (2) version1 (0)}
id-nf OBJECT IDENTIFIER ::= {ccitt recommendation q 1551 --upt-- module (0) uptUsefulDefinitions (0)
    nameForms (3) version1 (0)}

id-oc-uptProvider OBJECT IDENTIFIER ::= {id-oc 1}
id-oc-partner OBJECT IDENTIFIER ::= {id-oc 2}
id-oc-agreement OBJECT IDENTIFIER ::= {id-oc 3}
id-oc-administrativeUnit OBJECT IDENTIFIER ::= {id-oc 4}
id-oc-userProfile OBJECT IDENTIFIER ::= {id-oc 5}
id-oc-userProfileAlias OBJECT IDENTIFIER ::= {id-oc 6}
id-oc-calledUptUser OBJECT IDENTIFIER ::= {id-oc 7}
id-oc-callingUptUser OBJECT IDENTIFIER ::= {id-oc 8}
id-oc-callForwarding OBJECT IDENTIFIER ::= {id-oc 9}

id-ao-supplementaryService OBJECT IDENTIFIER ::= {id-ao 1}

id-at-providerId OBJECT IDENTIFIER ::= {id-at 1}
id-at-providedServiceId OBJECT IDENTIFIER ::= {id-at 2}
id-at-providedLocations OBJECT IDENTIFIER ::= {id-at 3}
id-at-pui OBJECT IDENTIFIER ::= {id-at 4}
id-at-specialPassword OBJECT IDENTIFIER ::= {id-at 5}
id-at-variablePassword OBJECT IDENTIFIER ::= {id-at 6}
id-at-nbOfFailedAuthentications OBJECT IDENTIFIER ::= {id-at 7}
id-at-userCredit OBJECT IDENTIFIER ::= {id-at 8}
id-at-callInfoRecords OBJECT IDENTIFIER ::= {id-at 9}
id-at-activeChargingService OBJECT IDENTIFIER ::= {id-at 10}
id-at-allowedServiceFeatures OBJECT IDENTIFIER ::= {id-at 11}
id-at-uptNumber OBJECT IDENTIFIER ::= {id-at 12}
id-at-defaultChargingReference OBJECT IDENTIFIER ::= {id-at 13}
id-at-icRegistrationAddress OBJECT IDENTIFIER ::= {id-at 14}
id-at-allowedRegistrationAddress OBJECT IDENTIFIER ::= {id-at 15}
id-at-supplServId OBJECT IDENTIFIER ::= {id-at 16}
id-at-supplServiceStatus OBJECT IDENTIFIER ::= {id-at 17}
id-at-forwardedToNumber OBJECT IDENTIFIER ::= {id-at 18}
id-at-typesOfNotification OBJECT IDENTIFIER ::= {id-at 19}
id-at-noReplyConditionTimer OBJECT IDENTIFIER ::= {id-at 20}
id-at-allowedDestinations OBJECT IDENTIFIER ::= {id-at 21}

id-nf-uptProviderNameForm OBJECT IDENTIFIER ::= {id-nf 1}
id-nf-partnerNameForm OBJECT IDENTIFIER ::= {id-nf 2}
id-nf-adminUnitNameForm OBJECT IDENTIFIER ::= {id-nf 3}
id-nf-agreedServiceNameForm OBJECT IDENTIFIER ::= {id-nf 4}
id-nf-userProfileNameForm OBJECT IDENTIFIER ::= {id-nf 5}
id-nf-userProfileAliasNameForm OBJECT IDENTIFIER ::= {id-nf 6}
id-nf-calledUptUserNameForm OBJECT IDENTIFIER ::= {id-nf 7}
id-nf-callingUptUserNameForm OBJECT IDENTIFIER ::= {id-nf 8}
id-nf-callForwardingNameForm OBJECT IDENTIFIER ::= {id-nf 9}

END

```


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 Z	Programming languages