



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

Q.1218

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(10/95)

RÉSEAU INTELLIGENT

**RECOMMANDATION RELATIVE À
L'INTERFACE POUR L'ENSEMBLE DE
CAPACITÉS 1 DU RÉSEAU INTELLIGENT**

Recommandation UIT-T Q.1218

(Antérieurement «Recommandation du CCITT»)

AVANT-PROPOS

L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'Union internationale des télécommunications (UIT). Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'études à traiter par les Commissions d'études de l'UIT-T lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la CMNT (Helsinki, 1^{er}-12 mars 1993).

La Recommandation révisée UIT-T Q.1218, que l'on doit à la Commission d'études 11 (1993-1996) de l'UIT-T, a été approuvée le 17 octobre 1995 selon la procédure définie dans la Résolution n° 1 de la CMNT.

NOTE

Dans la présente Recommandation, l'expression «Administration» est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue de télécommunications.

© UIT 1996

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

TABLE DES MATIÈRES

	<i>Page</i>	
0	Introduction	1
0.1	Références normatives	1
0.2	Méthodologie de définition	2
0.3	Exemples de scénarios physiques.....	3
0.4	Architecture du protocole d'application du réseau intelligent ou protocole INAP	9
0.5	Adressage dans le protocole INAP.....	11
0.6	Relations entre les Recommandations Q.1214 et la présente Recommandation	11
0.7	Mécanismes de compatibilité utilisés pour le protocole INAP	13
1	Règles applicables aux entités SACF/MACF	14
1.1	Réflexion du contexte d'application du sous-système TCAP	14
1.2	Exécution séquentielle/parallèle des opérations	14
2	Syntaxe abstraite du protocole d'application dans l'ensemble de capacités 1 du réseau intelligent	15
2.1	SSF-SCF, SCF-SRF Interface	15
2.1.1	IN CS-1 Operation Types.....	15
2.1.2	IN CS-1 Error Types	31
2.1.3	IN CS-1 Data Types	33
2.1.4	IN CS-1 application protocol (operation and error codes)	57
2.1.5	Application Contexts.....	65
2.2	Interface SCF-SDF.....	67
2.2.1	Introduction au sous-ensemble DAP dans la Recommandation X.500 du réseau intelligent.....	67
2.2.1.1	Adaptation des concepts contenus dans la Recommandation X.500 en fonction du réseau intelligent	67
2.2.1.2	Utilisation d'un sous-ensemble limité de Recommandations X.500	68
2.2.1.3	Hypothèses de travail	68
2.2.2	Le sous-ensemble protocole d'accès à l'annuaire X.500 du réseau intelligent	69
2.2.2.1	Examen de X.511 devant être utilisé dans le réseau intelligent.....	69
2.2.2.2	Sous-ensemble de protocole d'accès à l'annuaire.....	76
2.2.2.3	Profile X.501	85
2.2.2.4	Profil ASN.1 pour le service abstrait d'annuaire pour le CS-I du RI.....	85
3	Sémantique	92
3.1	Définition des procédures et des entités	92
3.1.1	Procédures relatives à l'entité application de l'entité SSF.....	92
3.1.1.1	Considérations générales	92
3.1.1.2	Modèle et interfaces	92
3.1.1.3	Relations entre la machine à états finis FSM de l'entité SSF et les fonctions CCF et de maintenance.....	93
3.1.1.4	La machine à états finis (SSME-FSM) de gestion de l'entité SSF (SSF management finite state machine)	95
3.1.1.5	Diagramme de transition d'état de l'entité SSF	97
3.1.1.6	Machine FSM de transfert de tâche à une entité SSF assistante	104
3.1.2	Procédures relatives à l'entité application de l'entité SCF	107
3.1.2.1	Généralités.....	107
3.1.2.2	Modèle et interfaces	107
3.1.2.3	Relation entre la machine SCF-FSM et les fonctions de maintenance/programmes de logique de service	108
3.1.2.4	Diagramme de transition d'état partiel de l'entité de gestion SCF (modèle SCME).....	110
3.1.2.5	Le modèle SCSM	113

	<i>Page</i>	
3.1.3	Procédures relatives à l'entité application SRF.....	128
3.1.3.1	Considérations générales.....	128
3.1.3.2	Modèle et interfaces.....	129
3.1.3.3	Relation entre la machine SRF-FSM et les fonctions de maintenance/gestion de connexion support.....	129
3.1.3.4	Le modèle SRSM.....	131
3.1.3.5	Exemples de procédures de commande SRF.....	134
3.1.4	Procédures relatives à l'entité application de l'entité SDF.....	150
3.1.4.1	Considérations générales.....	150
3.1.4.2	Modèle et interfaces.....	150
3.1.4.3	Relation entre la machine SDF-FSM et la fonction de maintenance.....	150
3.1.4.4	Modèle de transition d'état SDF.....	150
3.2	Procédures de traitement des erreurs.....	153
3.2.1	Procédures d'erreurs liées aux opérations.....	153
3.2.1.1	Erreur d'attribut.....	153
3.2.1.2	Canceled (Annulé).....	154
3.2.1.3	CancelFailed (Echec de l'Annulation).....	155
3.2.1.4	ETCFailed (Echec d'établissement de connexion temporaire).....	156
3.2.1.5	ImproperCallerResponse (Réponse inadaptée du demandeur).....	156
3.2.1.6	MissingCustomerRecord (Absence d'enregistrement du client).....	157
3.2.1.7	MissingParameter (Paramètre manquant).....	159
3.2.1.8	Erreur de nom.....	162
3.2.1.9	ParameterOutOfRange (Paramètre hors gamme).....	163
3.2.1.10	RequestedInfoError (Erreur d'information demandée).....	164
3.2.1.11	Erreur dans la prestation de service.....	164
3.2.1.12	Erreur de sécurité.....	165
3.2.1.13	SystemFailure (Echec du système).....	166
3.2.1.14	TaskRefused (Tâche refusée).....	168
3.2.1.15	UnavailableResource (Ressource indisponible).....	169
3.2.1.16	UnexpectedComponentSequence.....	170
3.2.1.17	UnexpectedDataValue (valeur imprévue d'une donnée).....	171
3.2.1.18	UnexpectedParameter (Paramètre non prévu).....	173
3.2.1.19	UnknownLegID (Identificateur de demi-appel inconnu).....	174
3.2.1.20	UnknownResource (Ressource inconnue).....	174
3.2.1.21	Erreur de mise à jour.....	174
3.2.2	Procédures d'erreur relatives aux entités.....	175
3.2.2.1	Fin de temporisation T _{SSF}	175
3.2.2.2	Fin de la temporisation d'application T _{SRF}	176
3.3	Procédures détaillées d'opération.....	176
3.3.1	ActivateServiceFiltering procedure (Procédure d'activation du service de filtrage).....	176
3.3.1.1	Description générale.....	176
3.3.1.2	Entité d'appel (SCF).....	179
3.3.1.3	Entité de réponse (SSF).....	180
3.3.2	Procédure ActivityTest (essai d'activité).....	181
3.3.2.1	Description générale.....	181
3.3.2.2	Entité d'appel (SCF).....	181
3.3.2.3	Entité de réponse (SSF).....	181
3.3.3	Procédure AddEntry (Ajouter une entrée).....	181
3.3.3.1	Description générale.....	181
3.3.3.2	Entité d'appel (SCF).....	182
3.3.3.3	Entité de réponse (SDF).....	182
3.3.4	AnalysedInformation Procedure (Procédure après l'information analysée).....	183
3.3.4.1	Description générale.....	183
3.3.4.2	Entité d'appel (SSF).....	184
3.3.4.3	Entité de réponse (SCF).....	185
3.3.5	AnalyseInformation procedure.....	186
3.3.5.1	Description générale.....	186
3.3.5.2	Entité d'appel (SCF).....	186
3.3.5.3	Entité de réponse (SSF).....	187

	<i>Page</i>	
3.3.6	ApplyCharging procedure (Appliquer la procédure de taxation)	189
3.3.6.1	Description générale.....	189
3.3.6.2	Entité d'appel (SCF)	189
3.3.6.3	Entité de réponse (SSF).....	190
3.3.7	Procédure ApplyChargingReport	190
3.3.7.1	Description générale.....	190
3.3.7.2	Entité d'appel (SSF).....	190
3.3.7.3	Entité de réponse (SCF)	191
3.3.8	AssistRequestInstructions procedure (Procédure d'instructions de demande d'assistance)	191
3.3.8.1	Description générale.....	191
3.3.8.2	Entité d'appel (SSF/SRF).....	192
3.3.8.3	Entité de réponse (SCF)	192
3.3.9	Bind Procedure (Procédure de rattachement).....	192
3.3.9.1	Description générale.....	192
3.3.9.2	Entité d'appel (SCF)	193
3.3.9.3	Entité de réponse (SDF)	193
3.3.10	CallGap procedure (Procédure d'espacement des appels)	193
3.3.10.1	Description générale.....	193
3.3.10.2	Entité d'appel (SCF)	196
3.3.10.3	Entité de réponse (SSF).....	196
3.3.11	CallInformationReport procedure (Procédure de comptes rendus d'informations d'appel)	197
3.3.11.1	Description générale.....	197
3.3.11.2	Entité d'appel (SSF).....	197
3.3.11.3	Entité de réponse (SCF)	198
3.3.12	CallInformationRequest procedure.....	199
3.3.12.1	Description générale.....	199
3.3.12.2	Entité d'appel (SCF)	199
3.3.12.3	Entité de réponse (SSF).....	200
3.3.13	Procédure d'annulation	200
3.3.13.1	Description générale.....	200
3.3.13.2	Entité d'appel (SCF)	200
3.3.13.3	Entité de réponse (SRF)	201
3.3.13.4	Entité de réponse (SSF).....	201
3.3.14	Procédure CollectedInformation (informations recueillies)	201
3.3.14.1	Description générale.....	201
3.3.14.2	Entité d'appel (SSF).....	203
3.3.14.3	Entité de réponse (SCF)	205
3.3.15	Procédure CollectInformation	206
3.3.15.1	Description générale.....	206
3.3.15.2	Entité d'appel (SCF)	206
3.3.15.3	Entité de réponse (SSF).....	207
3.3.16	Connect procedure (Procédure de connexion).....	207
3.3.16.1	Description générale.....	207
3.3.16.2	Entité d'appel (SCF)	208
3.3.16.3	Entité de réponse (SSF).....	209
3.3.17	Procédure ConnectToResource (Connecter à la ressource).....	210
3.3.17.1	Description générale.....	210
3.3.17.2	Entité d'appel (SCF)	210
3.3.17.3	Entité de réponse (SSF).....	211
3.3.18	Continue procedure (Procédure de maintien de l'appel).....	211
3.3.18.1	Description générale.....	211
3.3.18.2	Entité d'appel (SCF)	211
3.3.18.3	Entité de réponse (SSF).....	211
3.3.19	Procédure DisconnectForwardConnection (Déconnecter la connexion vers l'avant)	212
3.3.19.1	Description générale.....	212
3.3.19.2	Entité d'appel (SCF)	212
3.3.19.3	Entité de réponse (SSF).....	213

	<i>Page</i>
3.3.20 Procédure "EstablishTemporaryConnection" (Etablissement d'une connexion temporaire).....	213
3.3.20.1 Description générale.....	213
3.3.20.2 Entité d'appel (SCF).....	214
3.3.20.3 Entité de réponse (SSF).....	214
3.3.21 Procédure EventNotificationCharging (chargement de notification d'événement).....	215
3.3.21.1 Description générale.....	215
3.3.21.2 Entité d'appel (SSF).....	215
3.3.21.3 Entité de réponse (SCF).....	216
3.3.22 Procédure EventReportBCSM.....	216
3.3.22.1 Description générale.....	216
3.3.22.2 Entité d'appel (SSF).....	217
3.3.22.3 Entité de réponse (SCF).....	218
3.3.23 Procédure "FurnishChargingInformation" (Fournir les informations de taxation).....	218
3.3.23.1 Description générale.....	218
3.3.23.2 Entité d'appel (SCF).....	219
3.3.23.3 Entité de réponse (SSF).....	219
3.3.24 Procédure HoldCallInNetwork (Mise en garde du réseau).....	220
3.3.24.1 Description générale.....	220
3.3.24.2 Entité d'appel (SCF).....	220
3.3.24.3 Entité de réponse (SSF).....	221
3.3.25 Procédure "InitialDP" (Point de détection initial).....	221
3.3.25.1 Description générale.....	221
3.3.25.2 Entité d'appel (SSF).....	223
3.3.25.3 Entité de réponse (SCF).....	223
3.3.26 Procédure "InitiateCallAttempt".....	224
3.3.26.1 Description générale.....	224
3.3.26.2 Entité d'appel (SCF).....	224
3.3.26.3 Entité de réponse (SSF).....	225
3.3.27 Procédure "ModifyEntry" (Modifier l'entrée).....	225
3.3.27.1 Description générale.....	225
3.3.27.2 Entité d'appel (SCF).....	226
3.3.27.3 Entité de réponse (SDF).....	226
3.3.28 Procédure O_Answer (Réponse au départ).....	227
3.3.28.1 Description générale.....	227
3.3.28.2 Entité d'appel (SSF).....	227
3.3.28.3 Entité de réponse (SCF).....	228
3.3.29 Procédure O_Disconnect.....	229
3.3.29.1 Description générale.....	229
3.3.29.2 Entité d'appel (SSF).....	230
3.3.29.3 Entité de réponse (SCF).....	231
3.3.30 Procédure O_No_Answer.....	231
3.3.30.1 Description générale.....	231
3.3.30.2 Entité d'appel (SSF).....	232
3.3.30.3 Entité de réponse (SCF).....	233
3.3.31 Procédure OriginationAttemptAuthorized (Procédure d'autorisation des tentatives d'appel au départ).....	233
3.3.31.1 Description générale.....	233
3.3.31.2 Entité d'appel (SSF).....	235
3.3.31.3 Entité de réponse (SCF).....	235
3.3.32 Procédure PlayAnnouncement.....	236
3.3.32.1 Description générale.....	236
3.3.32.2 Entité d'appel (SCF).....	237
3.3.32.3 Entité de réponse (SRF).....	237
3.3.33 Procédure PromptAndCollectUserInfo.....	238
3.3.33.1 Description générale.....	238
3.3.33.2 Entité de réponse (SCF).....	241
3.3.33.3 Entité de réponse (SRF).....	241

	<i>Page</i>
3.3.34 Procédure ReleaseCall (libération d'appel)	242
3.3.34.1 Description générale.....	242
3.3.34.2 Entité d'appel (SCF)	242
3.3.34.3 Entité de réponse (SSF).....	243
3.3.35 Procédure RemoveEntry (Retirer une entrée).....	243
3.3.35.1 Description générale.....	243
3.3.35.2 Entité d'appel (SCF)	243
3.3.35.3 Entité de réponse (SDF)	244
3.3.36 Procédure RequestCurrentStatusReport (Demande de compte rendu d'état).....	244
3.3.36.1 Description générale.....	244
3.3.36.2 Entité d'appel (SCF)	244
3.3.36.3 Entité de réponse (SSF).....	245
3.3.37 Procédure RequestFirstStatusMatchReport (Demande de premier compte rendu de correspondance)	245
3.3.37.1 Description générale.....	245
3.3.37.2 Entité d'appel (SCF)	246
3.3.37.3 Entité de réponse (SSF).....	246
3.3.38 Procédure RequestNotificationChargingEvent (Evénement de demande de notification de taxation).....	247
3.3.38.1 Description générale.....	247
3.3.38.2 Entité d'appel (SCF)	247
3.3.38.3 Entité de réponse (SSF).....	247
3.3.39 Procédure RequestReportBCSMEEvent.....	248
3.3.39.1 Description générale.....	248
3.3.39.2 Entité d'appel (SCF)	249
3.3.39.3 Entité de réponse (SSF).....	250
3.3.40 Procédure ResetTimer (Remise à zéro du temporisateur)	250
3.3.40.1 Description générale.....	250
3.3.40.2 Entité d'appel (SCF)	250
3.3.40.3 Entité de réponse (SSF).....	250
3.3.41 Procédure RouteSelectFailure	251
3.3.41.1 Description générale.....	251
3.3.41.2 Entité d'appel (SSF).....	253
3.3.41.3 Entité de réponse (SCF)	253
3.3.42 Procédure Search (Recherche)	254
3.3.42.1 Description générale.....	254
3.3.42.2 Entité d'appel (SCF)	254
3.3.42.3 Entité de réponse (SDF)	254
3.3.43 Procédure SelectFacility.....	255
3.3.43.1 Description générale.....	255
3.3.43.2 Entité d'appel (SCF)	255
3.3.43.3 Entité de réponse (SSF).....	256
3.3.44 Procédure SelectRoute	256
3.3.44.1 Description générale.....	256
3.3.44.2 Entité d'appel (SCF)	257
3.3.44.3 Entité de réponse (SSF).....	257
3.3.45 Procédure SendChargingInformation	259
3.3.45.1 Description générale.....	259
3.3.45.2 Entité d'appel (SCF)	259
3.3.45.3 Entité de réponse (SSF).....	259
3.3.46 Procédure ServiceFilteringResponse.....	260
3.3.46.1 Description générale.....	260
3.3.46.2 Entité d'appel (SSF).....	261
3.3.46.3 Entité de réponse (SCF)	261
3.3.47 Procédure SpecializedResourceReport.....	262
3.3.47.1 Description générale.....	262
3.3.47.2 Entité d'appel (SRF)	262
3.3.47.3 Entité de réponse (SCF)	262

	<i>Page</i>	
3.3.48	Procédure StatusReport (Compte rendu d'état).....	263
3.3.48.1	Description générale.....	263
3.3.48.2	Entité d'appel (SSF).....	263
3.3.48.3	Entité de réponse (SCF)	263
3.3.49	Procédure TAnswer	264
3.3.49.1	Description générale.....	264
3.3.49.2	Entité d'appel (SSF).....	264
3.3.49.3	Entité de réponse (SCF)	265
3.3.50	Procédure TBusy	265
3.3.50.1	Description générale.....	265
3.3.50.2	Entité d'appel (SSF).....	267
3.3.50.3	Entité de réponse (SCF)	267
3.3.51	Procédure TDisconnect	268
3.3.51.1	Description générale.....	268
3.3.51.2	Entité d'appel (SSF).....	269
3.3.51.3	Entité de réponse (SCF)	270
3.3.52	Procédure TermAttemptAuthorized	270
3.3.52.1	Description générale.....	270
3.3.52.2	Entité d'appel.....	271
3.3.52.3	Entité de réponse (SCF)	271
3.3.53	Procédure TNoAnswer	272
3.3.53.1	Description générale.....	272
3.3.53.2	Entité d'appel (SSF).....	272
3.3.53.3	Entité de réponse (SCF)	273
3.3.54	Procédure Unbind (Détachement)	273
3.3.54.1	Description générale.....	273
3.3.54.2	Entité d'appel (SCF)	274
3.3.54.3	Entité de réponse (SDF)	274
3.3.55	RequestEveryStatusChangeReport procedure	274
3.3.55.1	Description générale.....	274
3.3.55.2	Entité d'appel (SCF)	274
3.3.55.3	Entité de réponse (SFF).....	275
3.4	Services assurés par le protocole d'application TCAP	275
3.4.1	Procédures normales.....	276
3.4.1.1	Messages SSF-vers-SCF.....	276
3.4.1.2	Messages SCF-vers-SSF.....	278
3.4.1.3	Messages de l'entité SCF-vers/à partir de-SRF.....	280
3.4.2	Procédures anormales.....	280
3.4.2.1	Messages de l'entité SCF-vers-l'association SSF/SRF.....	281
3.4.2.2	Messages de l'association SSF/SRF vers l'entité SSF.....	282
3.4.3	Etablissement du dialogue.....	282
3.4.3.1	Envoi d'une primitive de demande TC-BEGIN.....	282
3.4.3.2	Réception d'une indication TC-BEGIN.....	283
3.4.3.3	Réception de la première primitive d'indication TC-CONTINUE	283
3.4.3.4	Réception d'une primitive d'indication TC-END.....	283
3.4.3.5	Réception d'une primitive d'indication TC-U-ABORT	283
3.4.3.6	Réception d'une primitive d'indication TC-P-ABORT	283
3.4.4	Continuité du dialogue	283
3.4.4.1	Envoi de l'entité.....	283
3.4.4.2	Réception de l'entité	284
3.4.5	Fin de dialogue.....	284
3.4.5.1	Envoi d'une demande TC-END.....	284
3.4.5.2	Réception d'une primitive d'indication TC-END.....	284
3.4.6	Interruption de l'utilisateur	284
3.4.6.1	Envoi d'une primitive de demande TC-U-ABORT	284
3.4.6.2	Réception d'une primitive d'indication TC-U-ABORT	284

	<i>Page</i>	
3.4.7	Interruption du service fournisseur.....	284
3.4.7.1	Réception d'une indication TC-P-ABORT	284
3.4.8	Procédures pour les opérations INAP.....	285
3.4.8.1	Appel d'opération	285
3.4.8.2	Réception de l'appel d'opération.....	285
3.4.8.3	Réponse d'opération	285
3.4.8.4	Réception d'une réponse	285
3.4.8.5	Autres événements.....	286
3.4.9	Correspondances vers les services TC.....	287
3.4.9.1	Commande de dialogue	287
3.4.9.2	Procédures d'opération	288
Annexe A – Diagrammes SDL INAP		289
A.1	Introduction.....	289
A.2	Diagrammes SDL.....	289
Annexe B – Description des automates SCSM (Etats associés à l'entité SDF) et SDSM		342
B.1	Description du processus SCSM	342
B.1.1	Etat 1 – "Repos"	342
B.1.1.1	Procédures normales.....	342
B.1.1.2	Procédures exceptionnelles	342
B.1.2	Etat 2 – "Wait for subsequent requests"	342
B.1.2.1	Procédures normales.....	342
B.1.2.2	Procédures exceptionnelles	343
B.1.3	Etat 3 – "Wait_for_Bind_result"	344
B.1.3.1	Procédures normales.....	344
B.1.3.2	Procédures exceptionnelles	344
B.1.4	Etat 4 – "Rattachement SDF"	344
B.1.4.1	Procédures normales.....	344
B.1.4.2	Procédures exceptionnelles	345
B.2	Description du processus SDSM.....	345
B.2.1	Etat 1 – "Repos"	345
B.2.1.1	Procédures normales.....	345
B.2.1.2	Procédures exceptionnelles	345
B.2.2	Etat 2 – "Rattachement en cours"	345
B.2.2.1	Procédures normales.....	345
B.2.2.2	Procédures exceptionnelles	348
B.2.3	Etat 3 – "Rattachement SCF"	349
B.2.3.1	Procédures normales.....	349
B.2.3.2	Procédures exceptionnelles	349
Appendice I – Modélisation des données de service.....		352
I.1	Etude du besoin de modélisation des données de service	352
I.1.1	Généralités sur la modélisation.....	353
I.1.2	Modélisation du service – Définition de l'information nécessaire	353
I.2	Conseils pour l'élaboration des modèles de données de service.....	354
I.2.1	Sélection des ATTRIBUTS.....	354
I.2.1.1	ATTRIBUT Définition de types.....	355
I.2.1.2	Définition des valeurs autorisées	355
I.2.1.3	Définition des valeurs par défaut.....	356
I.2.1.4	Définition des valeurs MAX des attributs	356
I.2.1.5	Définition des durées de vie des attributs	356
I.2.1.6	Définition des règles de CORRESPONDANCE	356
I.2.1.7	Attribution des identifiants d'objets attributs	356
I.2.2	Sélection des CLASSES D'OBJET	357
I.2.2.1	Définition de la CLASSE D'OBJET.....	357
I.2.2.2	Sélection du nom RDN (NAME-FORM).....	357
I.2.2.3	Attribution des identifiants d'objets aux CLASSES D'OBJET	357

	<i>Page</i>
I.2.3	Définition de l'arbre DIT 357
I.2.3.1	Définition de la hiérarchie des objets (RÈGLES DE STRUCTURE) 357
I.2.3.2	Définition de chemins alternatifs de nommage (Alias)..... 359
I.2.3.3	Définition du nombre maximal d'entrées subordonnées 359
I.3	Exemple de service (similaire à UPT)..... 359
I.3.1	Description d'exemple de service 359
I.3.1.1	Élément de service – Appel entrant 359
I.3.1.2	Élément du service – Enregistrement d'appel entrant 360
I.3.1.3	Élément de service – Filtrage d'appel entrant 361
I.3.1.4	Élément de service – Appel sortant 361
I.3.1.5	Élément de service – Filtrage d'appel sortant par localisation..... 361
I.3.1.6	Informations générales relatives au service 362
I.3.2	Exemple de modelage des données de service 362
I.3.2.1	Sélection des attributs..... 362
I.3.2.2	Sélection des CLASSES D'OBJETS 367
I.3.2.3	Conception de l'arbre DIT du service..... 368
I.3.2.4	Attribution du contrôle d'accès 370
I.4	Définition des contextes 371
I.4.1	Contexte de valeur d'attribut d'index numérique 372
Appendice II – Aspects relatifs à l'interface du réseau intelligent devant faire l'objet d'un complément d'étude en ce qui concerne l'ensemble de capacités CS-1 372	
II.1	Généralités 372
II.1.1	Considérations générales 372
II.1.2	Relation avec les autres appendices des Recommandations de la série Q.1200 372
II.1.3	Format du document..... 372
II.2	Opérations..... 372
II.2.1	Considérations applicables à toutes les opérations de cet appendice 372
II.2.2	Opération Add Party (Ajouter un participant)..... 373
II.2.2.1	Considérations générales 373
II.2.2.2	Description 373
II.2.3	Opération Attach (Attacher) 373
II.2.3.1	Considérations générales 373
II.2.3.2	Description 373
II.2.4	Opération Change parties (changement de participants) 373
II.2.4.1	Description 373
II.2.5	Opérations "Detach" (Détacher)..... 373
II.2.5.1	Considérations générales 373
II.2.5.2	Description 373
II.2.6	Opération de mise en garde d'une connexion de correspondant 373
II.2.6.1	Description 373
II.2.7	Opération de tentative d'appel au départ (dans le cas de plus d'un correspondant) 374
II.2.7.1	Considérations générales 374
II.2.7.2	Description 374
II.2.8	Opération de reconnexion 374
II.2.8.1	Description 374
II.2.9	Opération de libération de la connexion d'un correspondant..... 374
II.2.9.1	Considérations générales 374
II.2.9.2	Description 374
II.3	Paramètres..... 374
II.3.1	Considérations applicables à tous les paramètres de cet appendice..... 374
II.3.2	Paramètre identification de demi-appel créé (issu de l'opération d'analyse des informations)..... 374
II.3.2.1	Description 374
II.3.3	Paramètre Identification de demi-appel créé (issu de l'opération de connexion)..... 374
II.3.3.1	Description 374

	<i>Page</i>
II.3.4 Paramètre Identification de demi-appel créé (issu de l'opération de tentative d'appel au départ)	375
II.3.4.1 Description	375
II.3.5 Paramètre Identification de demi-appel créé (issu de l'opération de sélection d'équipement)	375
II.3.5.1 Description	375
II.3.6 Paramètre Identification de demi-appel créé (issu de l'opération de sélection d'acheminement).....	375
II.3.6.1 Description	375
II.3.7 Paramètre demi-appel 1 (issu de l'opération du point de détection initial).....	375
II.3.7.1 Description	375
II.3.8 Paramètre demi-appel 2 (issu de l'opération du point de détection initial).....	375
II.3.8.1 Description	375
II.3.9 Paramètre identification d'appel	375
II.3.9.1 Description	375
II.4 Modules en notation ASN.1 des opérations et des paramètres.....	375
II.4.1 Syntaxe abstraite du protocole d'application de l'ensemble CS-1 du réseau intelligent – Appendice	375
II.5 Procédures.....	380
Appendice III – ASN.1 étendu.....	380

RÉSUMÉ

La présente Recommandation définit le protocole d'application du réseau intelligent offrant les capacités requises par les services cibles de l'ensemble CS-1 aux interfaces de cet ensemble (SSF-SCF, SCF-SDF et SCF-SRF) définies dans la Recommandation Q.1211. Elle définit certains scénarios possibles de piles de protocoles, les opérations entre les entités et les procédures à suivre au niveau de chaque entité fonctionnelle.

Les révisions concernant la présente Recommandation donnent des détails relatifs aux procédures d'exploitation, de même qu'aux services assurés à partir des couches sous-jacentes de protocole, et aux compléments, ceci afin d'assurer l'applicabilité à plusieurs réseaux de gestion, une nouvelle interface SCF-SDF pour la gestion et l'acquisition des données basées sur les Recommandations de la série X.500, ainsi que des procédures détaillées de relevés de défauts. Ces révisions ont pour objet de faciliter des réalisations cohérentes ayant des origines diverses dans différents réseaux.

Les travaux de normalisation associés figurent dans chacune des Recommandations de la série Q.1200 (Réseau intelligent).

RECOMMANDATION RELATIVE À L'INTERFACE POUR L'ENSEMBLE DE CAPACITÉS 1 DU RÉSEAU INTELLIGENT

(Helsinki, 1993; modifiée en 1995)

0 Introduction

La présente Recommandation spécifie le protocole d'application de réseau intelligent (INAP) (*intelligent network application protocol*) nécessaire pour prendre en charge l'ensemble de capacités CS-1. Ce protocole prend en charge les interactions entre les quatre entités fonctionnelles (FE) (*functional entity*) suivantes, définies dans le modèle fonctionnel du réseau intelligent:

- l'entité fonction commutation de service (SSF) (*service switching function*);
- l'entité fonction commande de service (SCF) (*service control function*);
- l'entité fonction ressource spécialisée (SRF) (*specialized resource function*);
- l'entité fonction données de service (SDF) (*service data function*).

Le domaine d'application couvert par la présente Recommandation est un nouveau perfectionnement du protocole INAP à la fois pour le réseau numérique à intégration des services (RNIS) et pour le réseau téléphonique public commuté (RTPC).

Il est conçu comme guide pour les fabricants et pour les exploitants de réseau afin d'assurer l'interfonctionnement entre les matériels de différents industriels pour toutes les interfaces définies dans l'ensemble de capacités CS-1 de RI (SCF-SSF, SCF-SRF et SCF-SDF) et entre les exploitants de réseau pour les interfaces entre les réseaux (SCF-SDF).

Etant donné que la présente Recommandation est prévue pour les premières étapes de la mise en place du RI dans le RNIS/RTPC existant, on suppose que seules des solutions simples peuvent résoudre les problèmes d'interaction de fonctionnement entre le RI/ et le RNIS et/ou le RTPC.

NOTE – Des solutions plus perfectionnées pour les interactions de fonctionnement entre le RI et le RNIS/RTPC devraient être étudiées dans le domaine d'application des versions futures du protocole INAP et des normes de signalisation du RNIS et/ou du RTPC.

0.1 Références normatives

Les Recommandations et autres références suivantes contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente Recommandation. Au moment de la publication, les éditions indiquées étaient en vigueur. Toute recommandation ou autre référence est sujette à révision; tous les utilisateurs de la présente Recommandation sont donc invités à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et autres références indiquées ci-après. Une liste des Recommandations UIT-T en vigueur est publiée régulièrement.

- Recommandation UIT-T X.500 (1993) | ISO/IEC 9594-1:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: vue d'ensemble des concepts, des modèles et des services.*
- Recommandation UIT-T X.501 (1993) | ISO/IEC 9594-2:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: les modèles.*
- Recommandation UIT-T X.509 (1993) | ISO/IEC 9594-8:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: le cadre d'authentification.*
- Recommandation UIT-T X.511 (1993) | ISO/IEC 9594-3:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: définition du service abstrait.*
- Recommandation UIT-T X.518 (1993) | ISO/IEC 9594-4:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: procédures réparties.*
- Recommandation UIT-T X.519 (1993) | ISO/IEC 9594-5:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: spécifications des protocoles.*
- Recommandation UIT-T X.680 (1994) | ISO/IEC 8824-1:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation abstraite de syntaxe N° 1 (ASN.1): spécification de la notation de base.*

- Recommandation UIT-T X.681 (1994) | ISO/IEC 8824-2:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation abstraite de syntaxe N° 1 (ASN.1): spécification de l'objet d'information.*
- Recommandation UIT-T X.682 (1994) | ISO/IEC 8824-3:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation abstraite de syntaxe N° 1 (ASN.1): spécifications relatives aux contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/IEC 8824-4:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation abstraite de syntaxe N° 1 (ASN.1): paramétrisation des spécifications ASN.1.*
- Recommandation UIT-T X.690 (1994) | ISO/IEC 8825-1:1995, *Technologie de l'information – Interconnexion des systèmes ouverts – Spécifications des règles de codage de la notation ASN.1: règles de codage pour les versions de base, canonique et distinguée.*
- Recommandation UIT-T X.880 (1994) | ISO/IEC 13712-1:1995, *Technologie de l'information – Opérations à distance: Concepts, modèle et notation.*
- Recommandation Q.29 du CCITT (1988), *Sources de bruit et méthodes de réduction de bruit dans l'autocommutateur téléphonique.*
- Recommandation UIT-T Q.711 (1993), *Signalisation sémaphore N° 7 – Organe de commande de la connexion de signalisation.*
- Recommandation UIT-T Q.762 (1993), *Signalisation sémaphore N° 7 – Fonctions générales des messages et des signaux de la partie usager RNIS (ISUP) du Système de signalisation N° 7.*
- Recommandation UIT-T Q.763 (1993), *Signalisation sémaphore N° 7 – Formats et codes de la partie usager RNIS (ISUP) du Système de signalisation N° 7.*
- Recommandation UIT-T Q.771 (1993), *Signalisation sémaphore N° 7 – Description fonctionnelle des capacités de transaction.*
- Recommandation UIT-T Q.772 (1993), *Signalisation sémaphore N° 7 – Capacités de transaction: définitions des éléments d'information.*
- Recommandation UIT-T Q.773 (1993), *Signalisation sémaphore N° 7 – Capacités de transaction: format et codage des messages.*
- Recommandation UIT-T Q.774 (1993), *Signalisation sémaphore N° 7 – Capacités de transaction: procédures.*
- Recommandation UIT-T Q.775 (1993), *Signalisation sémaphore N° 7 – Directives pour l'utilisation des capacités de transaction.*
- Recommandation UIT-T Q.931 (1993), *Système de signalisation numérique N° 1 (DSS 1) – Spécification de la couche 3 de l'interface réseau-usager RNIS pour la commande des appels simples.*
- Recommandation UIT-T Q.932 (1993), *Système de signalisation numérique N° 1 (DSS 1) – Procédures génériques pour la commande des services supplémentaires du RNIS.*
- Recommandation UIT-T Q.1290 (1995), *Réseau intelligent: glossaire des termes employés dans la définition des réseaux intelligents.*

0.2 Méthodologie de définition

La définition de ce protocole peut être subdivisée en trois parties:

- la définition des règles applicables aux entités SACF/MACF (*single association control function/multiple association control function*) (fonction commande d'association unique/fonction commande d'association multiple) pour le protocole (voir l'article 1);
- la définition des opérations transférées entre entités (voir l'article 2);
- la définition des actions entreprises au niveau de chaque entité (voir l'article 3).

Les règles SACF/MACF sont définies en langage naturel. Les opérations sont définies en notation ASN.1 (voir les Recommandations X.208 et X.680) et les actions en termes de diagrammes de transition d'état. D'autres indications sur les actions à exécuter à la réception d'une opération sont données en 2, les flux d'information correspondants sont traités à l'article 6/Q.1214 (voir 0.6 pour la relation entre les flux d'information et les opérations).

Le protocole INAP est un protocole d'utilisateur d'élément de service d'opération distante (ROSE) (*remote operation service element*) (voir les Recommandations X.219 et X.229) contenu dans la sous-couche composantes du sous-système application de gestion des transactions (TCAP) (*transaction capability application part*) (voir les Recommen

datations Q.771 à Q.775) et le système de signalisation numérique n° 1 (DSS 1) (*digital subscriber signalling 1*) (voir la Recommandation Q.932). A présent, les unités de données de protocole d'application (APDU) (*application protocol data unit*) de l'élément ROSE sont acheminées dans les messages de la sous-couche transaction du système de signalisation n° 7 (système SS n° 7) et dans les messages ENREGISTREMENT, FACILITÉ et commande d'appel définis dans la Recommandation Q.931, du système DSS 1. D'autres protocoles pourront être ajoutés ultérieurement.

Le protocole INAP (en tant qu'utilisateur d'élément ROSE) et le protocole des éléments ROSE sont spécifiés en notation ASN.1. Actuellement, les seules règles de codage normalisé possibles des unités PDU résultantes sont les règles de codage de base BER (voir la Recommandation X.209).

0.3 Exemples de scénarios physiques

Ce protocole prendra en charge toute mise en correspondance d'entités fonctionnelles avec des entités physiques (PE) (*physical entity*). Les opérateurs de réseau et les fabricants d'équipements étant libres d'installer les entités fonctionnelles en des endroits différents, il leur appartient de décider des emplacements communs les plus rationnels. Le protocole est donc défini en supposant une distribution maximale (c'est-à-dire une entité physique par entité fonctionnelle).

Les scénarios étudiés dans le présent paragraphe illustrent la prise en charge du protocole INAP dans un environnement de réseau utilisant le système SS n° 7, mais cela ne signifie pas que seul le système SS n° 7 puisse servir de protocole de réseau pour prendre en charge le protocole INAP.

Les procédures types de commande d'entité SRF à appliquer pour chaque scénario physique sont traitées en 3.1.3.5.

L'interface entre l'entité SCF distante et l'entité SDF est constituée par le protocole INAP utilisant le sous-système TCAP qui utilise à son tour les services des sous-systèmes SCCP en mode sans connexion et MTP (voir la Figure 1). L'entité SDF est responsable de l'interfonctionnement avec d'autres protocoles pour accéder à d'autres types de réseaux.

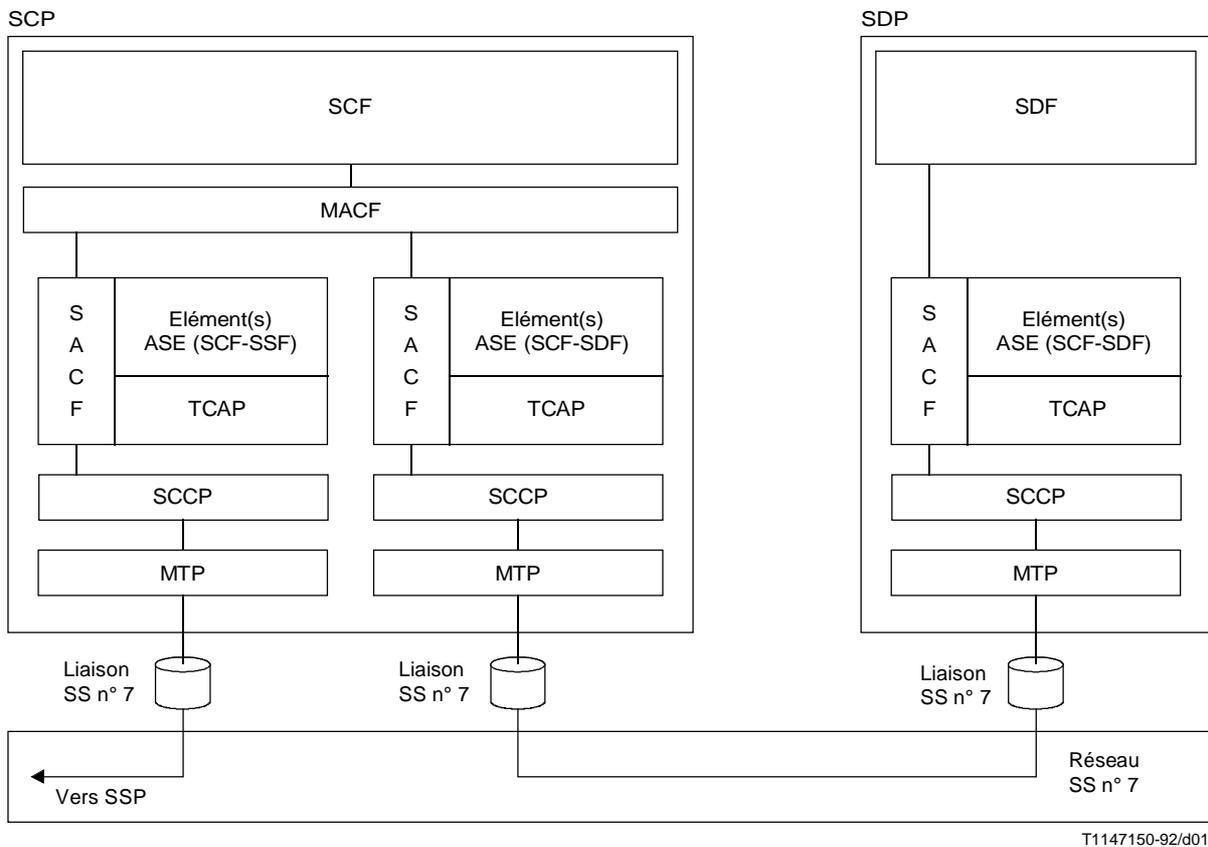


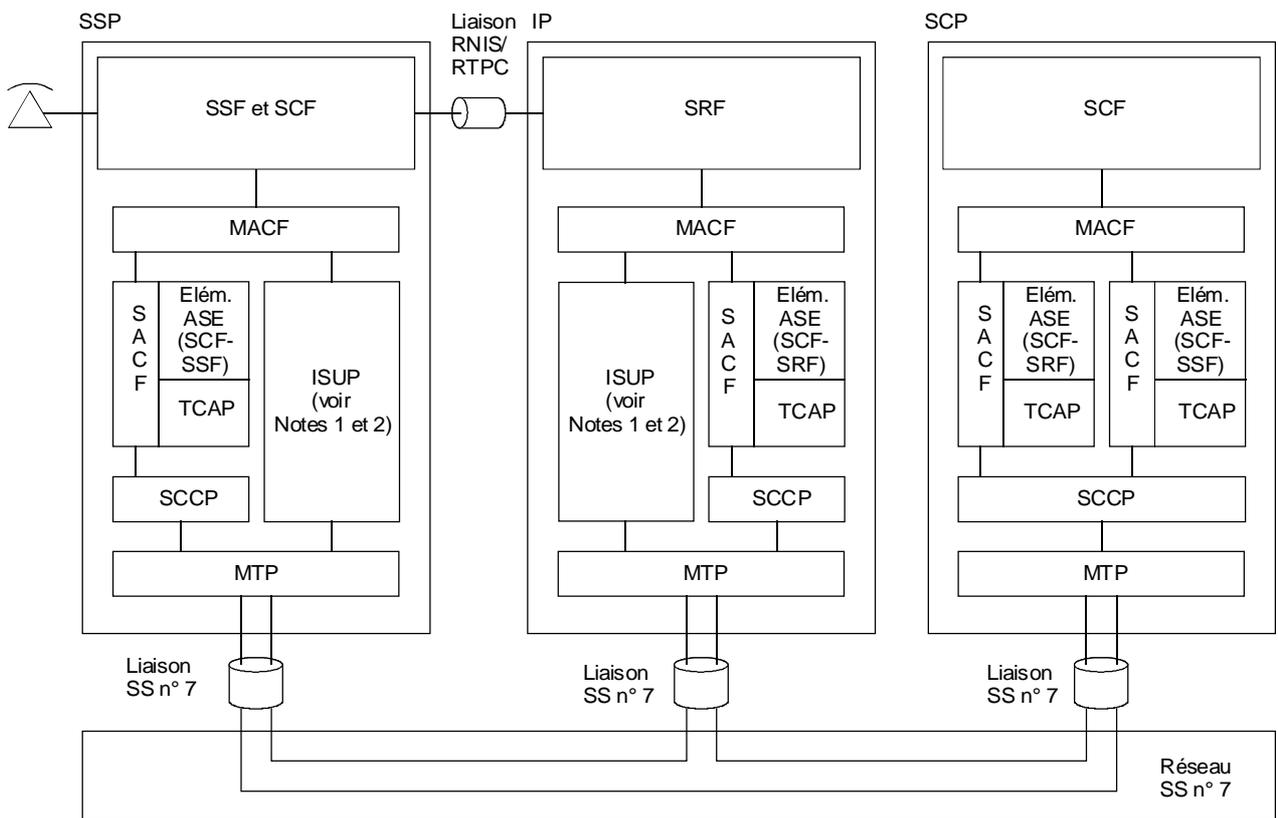
FIGURE 1/Q.1218
Interface physique entre points SCP et SDP

Quand le sous-système TCAP apparaîtra dans une des figures suivantes, il sera censé représenter les fonctionnalités du sous-système TCAP associées à un dialogue et à une transaction uniques (par opposition à l'entité du sous-ensemble TCAP).

S'il est nécessaire de segmenter et de réassembler les messages INAP à l'interface entre l'entité SCF et l'entité SDF (et à d'autres interfaces si nécessaire) en raison de la longueur des messages, il convient d'utiliser la procédure de segmentation et de réassemblage pour les messages SCCP en mode sans connexion comme cela est spécifié dans Q.714.

Un certain nombre d'exemples de scénarios de prise en charge des entités fonctionnelles SCF, SSF et SRF sont identifiés sous la forme d'entités physiques. Ces scénarios sont illustrés aux Figures 2 à 6. Chaque exemple se caractérise par:

- i) la méthode de prise en charge de la relation SCF-SRF;
- ii) le type de système de signalisation entre les entités SSF et SRF.



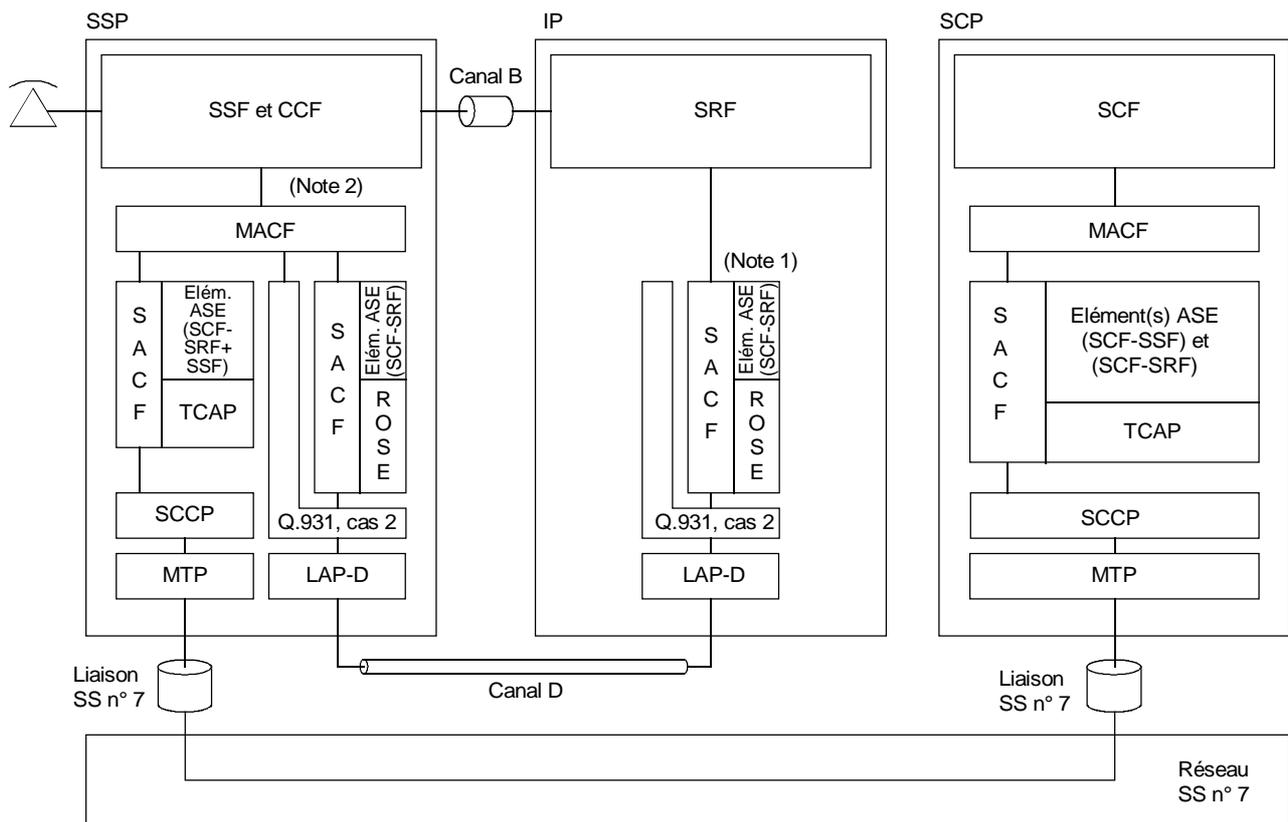
T1147160-92/d02

NOTES

- 1 Le transfert de l'information de corrélation doit être prise en charge. Ce transfert peut être réalisé par le sous-système ISUP sans ajouter un nouveau paramètre ISUP.
- 2 On peut utiliser d'autres systèmes de signalisation.

FIGURE 2/Q.1218

Exemple d'architecture de prise en charge de la fonction SRF, cas 1
(fonction SRF dans un périphérique intelligent (IP) connecté au point SSP
et avec accès par le point SCP par l'intermédiaire d'une connexion directe SS n° 7)



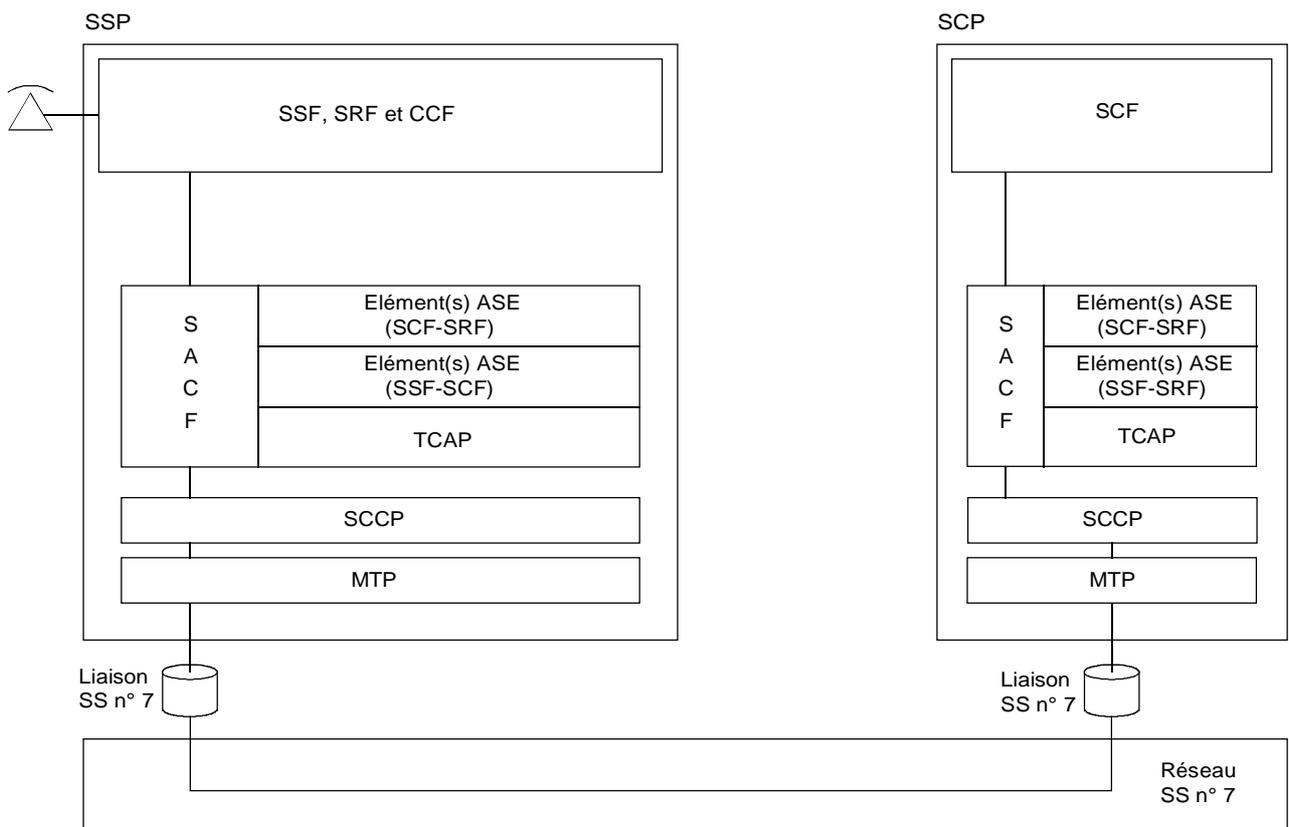
T1146670-92/d03

NOTES

- 1 Les flux d'information entre les fonctions SCF et SRF sont pris en charge par cette entité ROSE.
- 2 La fonction de relais est assurée par la fonction MACF ou par le processus d'application au point SSP.

FIGURE 3/Q.1218

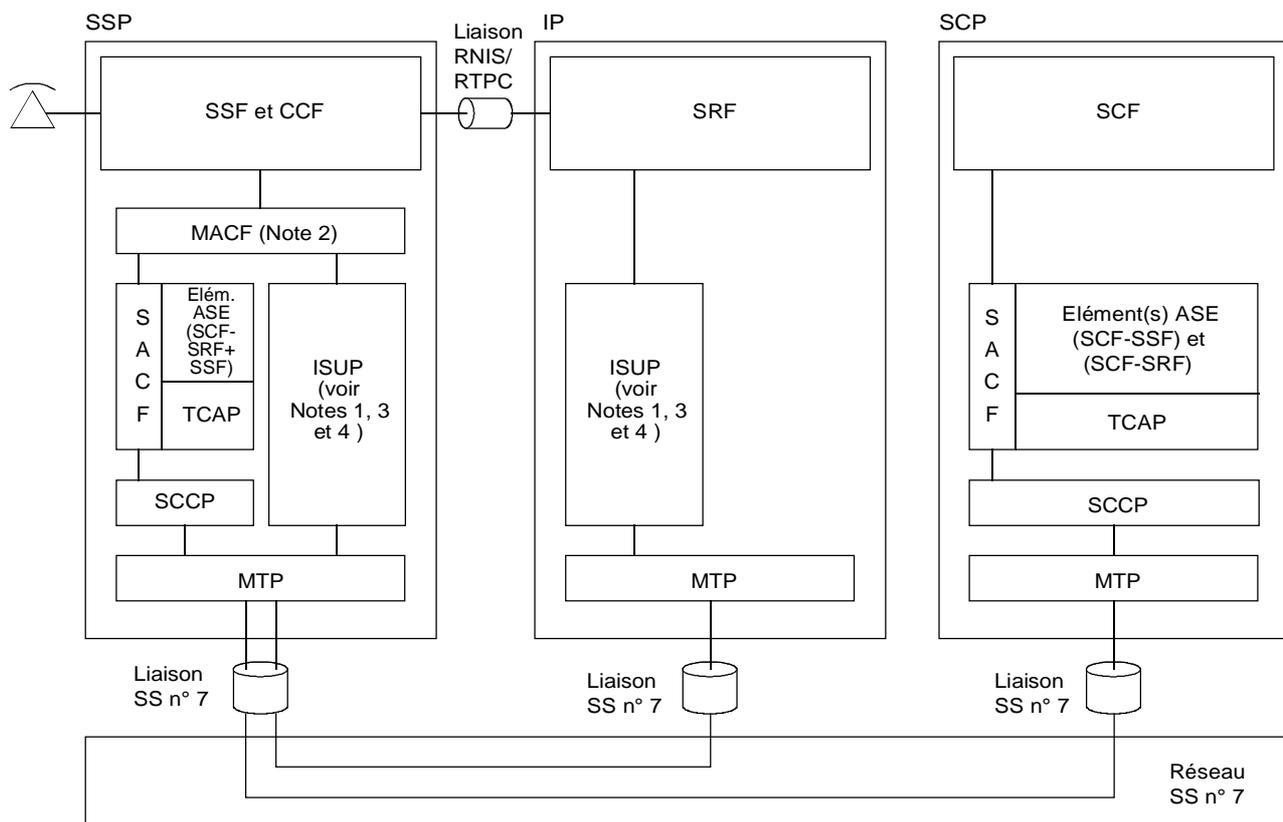
**Exemple d'architecture de prise en charge de la fonction SRF, cas 2
(fonction SRF dans un périphérique intelligent connecté au point SSP et avec accès
par le point SCP par l'intermédiaire d'un canal D via le point SSP)**



T1146680-92/d04

FIGURE 4/Q.1218

**Exemple d'architecture de prise en charge de la fonction SRF, cas 3
(fonction SRF dans le point SSP et avec accès par le protocole d'application du point SSP)**



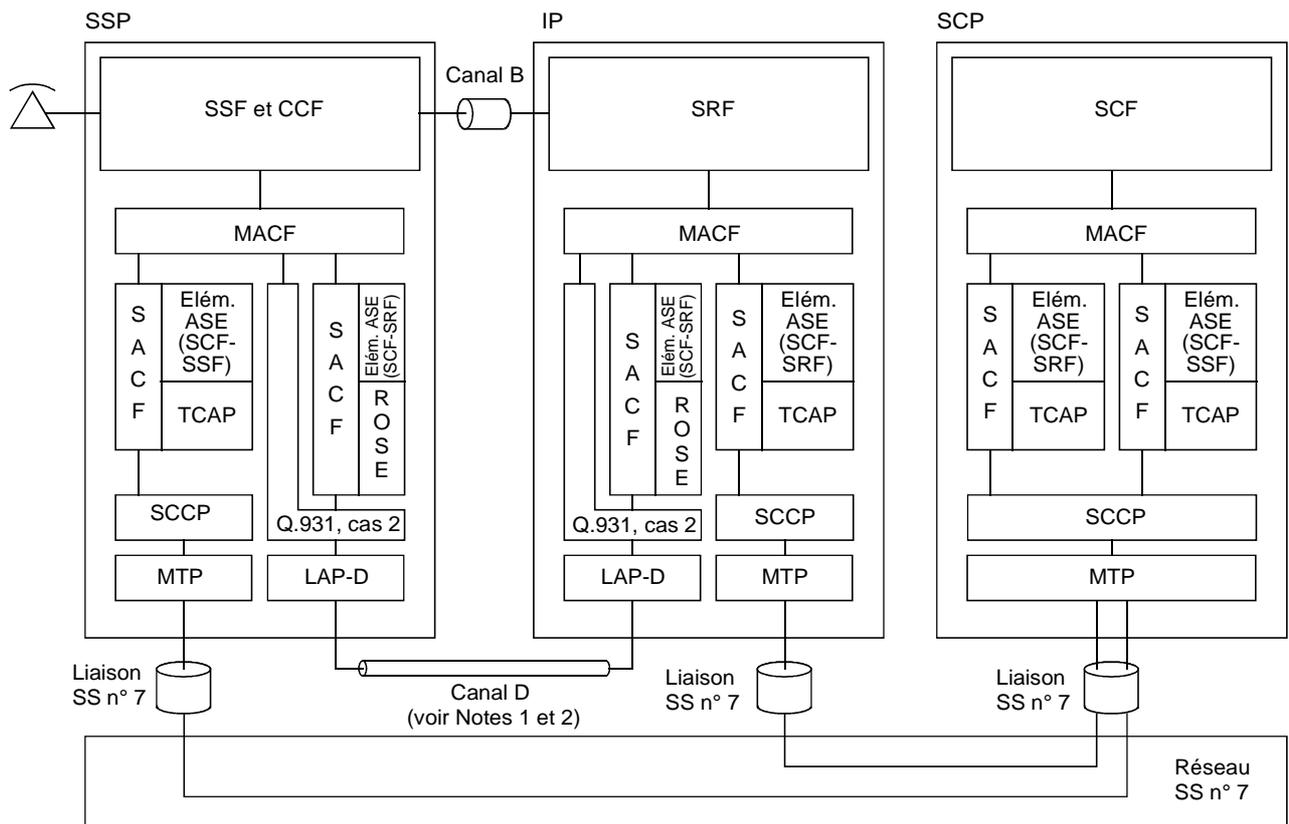
T1146690-92/d05

NOTES

- 1 Les flux d'information entre les fonctions SCF et SRF ainsi que les opérations de commande de connexion sont directement pris en charge par le sous-système ISUP.
- 2 La fonction relais est assurée par la fonction MACF ou par le processus d'application du point SSP.
- 3 On suppose que le sous-système ISUP permet l'acheminement de l'information ROSE.
- 4 On peut utiliser d'autres systèmes de signalisation.

FIGURE 5/Q.1218

**Exemple d'architecture de prise en charge de la fonction SRF, cas 4
(fonction SRF dans un périphérique intelligent connecté au point SSP et avec accès
par le point SCP par l'intermédiaire du sous-système ISUP via le point SSP)**



T1146700-92/d06

NOTES

- 1 Le transfert des informations de corrélation doit être pris en charge.
- 2 On peut utiliser d'autres systèmes de signalisation.

FIGURE 6/Q.1218

**Exemple d'architecture de prise en charge de la fonction SRF, cas 5
(fonction SRF dans un périphérique intelligent connecté aux points SCP et SSP
avec accès par l'intermédiaire respectivement d'une liaison SS n° 7 et d'un canal D)**

Le Tableau 1 donne un récapitulatif de la sélection des caractéristiques pour chaque figure.

TABLEAU 1/Q.1218

Type de système de signalisation entre entités SSF et SRF	Méthode de prise en charge de la relation SCF-SRF	
	Liaison directe TCAP	Relais via le SSP
ISUP	Figure 2 ^{a)}	Figure 5 ^{d)}
DSS 1	Figure 6 ^{e)}	Figure 3 ^{b)}
Dépend de la mise en œuvre	Selon Figures 2 ou 6 mais avec interface SCP-IP dépendant de la mise en œuvre	Figure 4 ^{c)}
<p>Informations complémentaires se rapportant à toutes les figures:</p> <p>a) Figure 2: toutes les associations sont prises en charge par le système de signalisation n° 7, par les sous-systèmes TCAP ou ISUP. Dans ce cas, le périphérique intelligent IP est l'un des nœuds du réseau.</p> <p>b) Figure 3: l'accès au périphérique intelligent (IP) ne peut s'effectuer que par le système de signalisation DSS 1. Ce périphérique peut être une entité située hors du réseau.</p> <p>c) Figure 4: le point de commutation de service ou point SSP prend en charge à la fois les entités CCF/SSF et SRF. Le traitement de l'entité SRF par l'entité SCF peut être le même que celui de la Figure 3.</p> <p>d) Figure 5: on ne peut accéder au périphérique intelligent (IP) que par le sous-système ISUP. Le traitement de l'entité SRF par l'entité SCF pourrait être le même que celui de la Figure 3.</p> <p>e) Figure 6: le traitement de l'entité SRF par l'entité SCF pourrait être le même que celui de la Figure 2. On pourrait utiliser d'autres types de systèmes de signalisation.</p>		

0.4 Architecture du protocole d'application du réseau intelligent ou protocole INAP

4 Architecture du protocole d'application du réseau intelligent ou protocole INAP

De nombreux termes utilisés dans ce paragraphe sont tirés de l'ISO IS-9545 relative à la structure de la couche application de l'OSI.

L'architecture du protocole INAP peut être celle de la Figure 7.

Une entité physique a des interactions uniques (cas a) ou des interactions multiples coordonnées (cas b) avec les autres entités physiques.

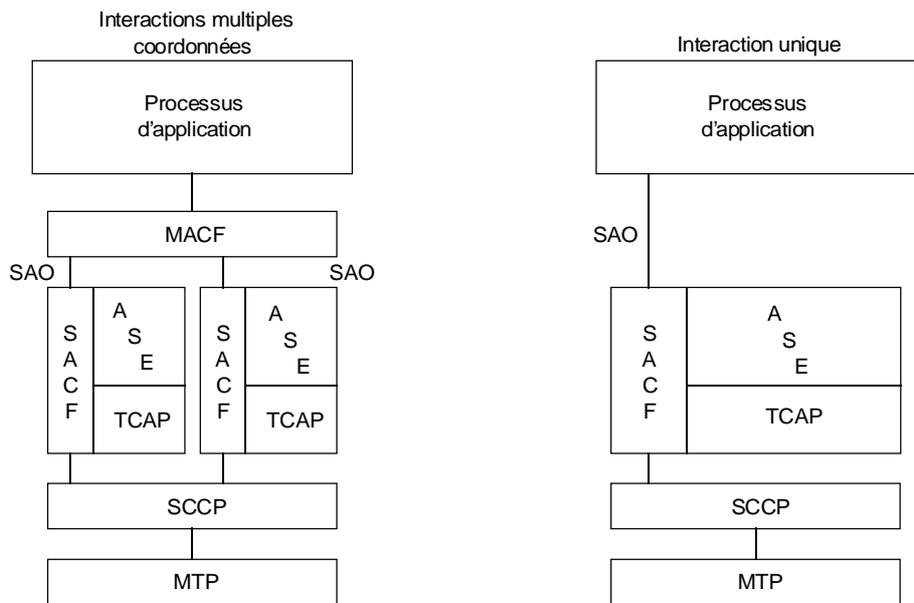
Dans le cas a, l'entité SACF assure, en utilisant les éléments ASE, une fonction de coordination, qui comprend l'ordonnancement des opérations prises en charge par les éléments ASE (sur la base de l'ordre des primitives reçues). L'objet SAO représente l'entité SACF plus un ensemble d'éléments ASE à utiliser lors d'une interaction unique entre deux entités physiques.

Dans le cas b, la fonction MACF assure la fonction de coordination entre plusieurs objets SAO dont chacun interagit avec un objet SAO dans une entité physique distante.

Chaque élément ASE prend en charge une ou plusieurs opérations. La description de chaque opération est liée à l'action de modélisation des entités fonctionnelles correspondantes (voir la Recommandation Q.1214 et l'article 3 de la présente Recommandation). Chaque opération est spécifiée au moyen d'une macro OPERATION décrite à la Figure 8, excepté pour l'interface SCF vers SDF, qui est décrite au moyen de la notation de classe CLASS.

L'utilisation du mécanisme de négociation du contexte d'application [défini dans les Recommandations de la série Q.770 (*sous-système application pour la gestion des transactions*)] permet à deux entités en communication de déterminer exactement leurs capacités réelles et les capacités nécessaires à l'interface. Ce mécanisme devrait faciliter l'évolution grâce aux ensembles de capacités.

Lorsque deux entités en communication n'indiquent pas de contexte d'application particulier, un mécanisme de préarrangement du contexte doit être pris en charge.

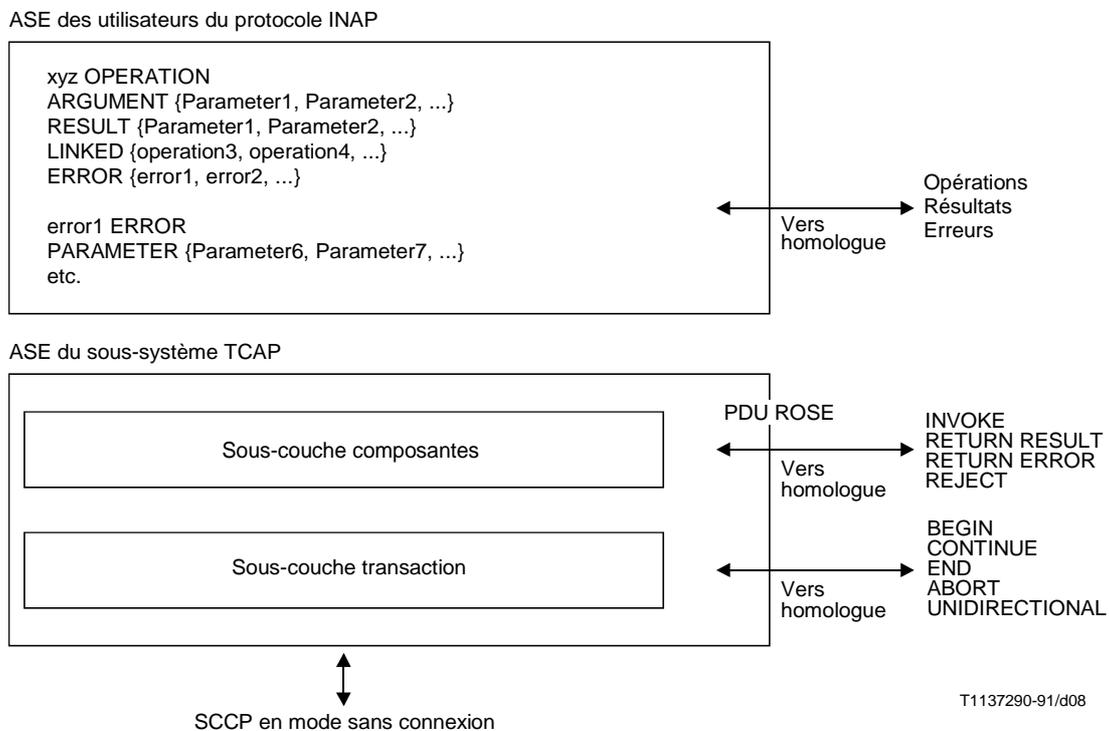


T1146710-92/d07

- SACF Fonction de commande d'association unique (*single association control function*)
- MACF Fonction de commande d'association multiple (*multiple association control function*)
- SAO Objet d'association unique ou objet SAO (*single association object*)
- ASE Elément du service d'application (*application service element*)
- INAP Protocole d'application du réseau intelligent (*intelligent network application protocol*)

NOTE – Le protocole INAP regroupe les spécifications de tous les éléments du service application.

FIGURE 7/Q.1218
Architecture du protocole INAP



T1137290-91/d08

FIGURE 8/Q.1218
Description du fonctionnement

0.4.1 Gestion des encombrements de la signalisation INAP pour le Système de signalisation n° 7

Le type de procédure qui a été défini pour la gestion des encombrements du côté utilisateur du RNIS s'appliquera également à la signalisation INAP. Les procédures applicables à la gestion des encombrements de la signalisation INAP seront dans la mesure du possible alignées sur les procédures de gestion des encombrements du côté utilisateur du RNIS telles qu'elles ont été définies (voir D.211/Q.767), c'est-à-dire qu'à la réception de la primitive N-PCSTATE contenant l'information «encombrement du point de signalisation» envoyée par le SCCP, la procédure INAP réduira en plusieurs étapes la charge de trafic (par exemple points de détection initiaux, informations analysées ou tentatives d'appel) dans la direction concernée.

La procédure ci-dessus peut s'appliquer uniquement au trafic utilisant l'adressage Point Code MTP dans la direction concernée.

0.5 Adressage dans le protocole INAP

L'appellation globale du sous-système SCCP et l'adressage des codes sémaphores dans le sous-système MTP [voir les Recommandations des séries Q.700 (*sous-système transport de messages*) et Q.710 (*sous-système commande des connexions sémaphores*)] permettent aux unités PDU d'atteindre leur destination physique (c'est-à-dire le code point correct) indépendamment du réseau dans lequel elles se trouvent.

Dans un nœud, le choix du ou des numéro(s) SSN attribué(s) au protocole INAP dépend de l'opérateur/réalisateur du réseau.

Indépendamment de ce qui précède, tout schéma d'adressage pris en charge par le sous-système SCCP peut être utilisé.

0.6 Relations entre les Recommandations Q.1214 et la présente Recommandation

La liste ci-après énumère tous les flux d'information et les fait correspondre un à un avec une opération, sauf exception indiquée.

<i>Référence Rec. Q.1214</i>	<i>Flux d'information</i>	<i>Nom de l'opération correspondante</i>
6.4.2.1	Activation du filtrage des services	ActivateServiceFiltering
6.4.2.2	Essai d'activité	ActivityTest
6.4.2.3	Réponse d'essai d'activité	Return Result from ActivityTest
6.4.2.4	Informations analysées	AnalysedInformation, InitialDP, EventReportBCSM
6.4.2.5	Analyse des informations	AnalyseInformation, Connect
6.4.2.6	Application de la taxation	ApplyCharging
6.4.2.7	Compte rendu d'application de la taxation	ApplyChargingReport
6.4.2.8	Instructions de demande d'assistance	AssistRequestInstructions
6.4.2.9	Espacement des appels	CallGap
6.4.2.10	Compte rendu d'informations d'appel	CallInformationReport
6.4.2.11	Demande d'informations d'appel	CallInformationRequest
6.4.2.13	Annulation de demande de compte rendu d'état	CancelStatusReportRequest
6.4.2.14	Informations recueillies	CollectedInformation, InitialDP, EventReportBCSM
6.4.2.15	Recueil des informations	CollectInformation
6.4.2.16	Connexion	Connect
6.4.2.17	Connexion à la ressource	ConnectToResource

<i>Référence Rec. Q.1214</i>	<i>Flux d'information</i>	<i>Nom de l'opération correspondante</i>
6.4.2.18	Continuation	Continue
6.4.2.19	Déconnexion vers l'avant	DisconnectForwardConnection
6.4.2.20	Etablissement d'une connexion temporaire	EstablishTemporaryConnection
6.4.2.21	Notification d'événement de type taxation	EventNotificationCharging
6.4.2.22	Compte rendu sur événement de type BCSM	EventReportBCSM
6.4.2.23	Fourniture des informations de taxation	FurnishChargingInformation
6.4.2.24	Maintien de l'appel dans le réseau	HoldCallInNetwork, ResetTimer and FurnishChargingInformation
6.4.2.25	Point de détection initial	InitialDP
6.4.2.26	Emission d'une tentative d'appel	InitiateCallAttempt
6.4.2.27	OAnswer (réponse au départ)	OAnswer, InitialDP, EventReport BCSM
6.4.2.28	OCalledPartyBusy (occupation du demandé au départ)	OCalledPartyBusy, InitialDP, EventReport BCSM
6.4.2.29	ODisconnect (déconnexion au départ)	ODisconnect, InitialDP, EventReport BCSM
6.4.2.30	OMidCall (semi-communication au départ)	OMidCall, InitialDP, EventReport BCSM
6.4.2.31	ONo_Answer (non-réponse au départ)	ONo_Answer, InitialDP, EventReport BCSM
6.4.2.32	Tentative d'appel au départ autorisée	OriginationAttemptAuthorized
6.4.2.33	Libération de l'appel	ReleaseCall
6.4.2.34	Demande de notification sur événement de taxation	RequestNotificationChargingEvent
6.4.2.35	Demande de compte rendu sur événement de type BCSM	RequestReportBCSMEvent
6.4.2.36	Demande de compte rendu d'état	RequestCurrentStatusReport RequestFirstStatusMatchReport RequestEveryStatusChangeReport
6.4.2.37	Remise à zéro de la temporisation	ResetTimer
6.4.2.38	Echec de sélection d'acheminement	Route Select Failure, InitialDP, EventReport BCSM
6.4.2.39	Sélection d'équipement	SelectFacility
6.4.2.40	Sélection d'acheminement	Select Route, Connect
6.4.2.41	Envoi des informations de taxation	SendChargingInformation
6.4.2.42	Réponse de filtrage de service	ServiceFilteringResponse
6.4.2.43	Compte rendu d'état	StatusReport
6.4.2.44	TAnswer (réponse à l'arrivée)	TAnswer, InitialDP, EventReport BCSM
6.4.2.45	TCalled Party Busy (occupation du demandé à l'arrivée)	TCalledParty Busy, InitialDP, EventReport BCSM
6.4.2.46	TDisconnect (déconnexion à l'arrivée)	TDisconnect, InitialDP EventReport BCSM
6.4.2.47	Term Attempt Authorized (Autorisation d'appel à l'arrivée)	TermAttemptAuthorized, InitialDP

<i>Référence Rec. Q.1214</i>	<i>Flux d'information</i>	<i>Nom de l'opération correspondante</i>
6.4.2.48	TMidCall (semi-communication à l'arrivée)	TMidCall, InitialDP, EventReport BCSM
6.4.2.49	TNoAnswer (pas de réponse à l'arrivée)	TNoAnswer, InitialDP, EventReport BCSM
6.5.2.1	Instructions de demande d'assistance par SRF	AssistRequestInstructions
6.5.2.2	Annulation d'annonce	Cancel
6.5.2.3	Informations d'utilisateur recueillies	Return Result from Prompt and collect user information
6.5.2.4	Passage d'annonce	PlayAnnouncement
6.5.2.5	Recueil d'informations d'utilisateur sur invitation	PromptAndCollectUserInformation
6.5.2.6	Compte rendu de ressource spécialisée	SpecializedResourceReport
6.6.2.1	Rechercher	Search
6.6.2.2	Rechercher le résultat	Return Result from Search
6.6.2.3	ModifyEntry (modifier une entrée)	ModifyEntry
6.6.2.4	Modify Entry Result	Return Result from ModifyEntry
6.6.2.5	Authentifier	Bind
6.6.2.6	Authentifier le résultat	Return Result from Bind
6.6.2.7	AddEntry (ajouter une entrée)	AddEntry
6.6.2.8	Add Entry Result (ajouter le résultat d'une entrée)	Return Result from AddEntry
6.6.2.9	RemoveEntry	RemoveEntry
6.6.2.10	Remove Entry Result	Return Result from RemoveEntry

0.7 Mécanismes de compatibilité utilisés pour le protocole INAP

0.7.1 Introduction

Ce paragraphe définit les mécanismes de compatibilité qui doivent être utilisés pour assurer la compatibilité des versions futures du protocole INAP.

Il y a trois catégories de compatibilité:

- *Changements mineurs au protocole INAP dans de futures versions normalisées:*
 changement mineur peut être défini comme un changement de fonctionnalité qui n'est pas essentiel pour le service de réseau intelligent demandé. En cas de modification d'une fonction existante, il peut être admis que la fonction en question soit exécutée soit sous forme d'une ancienne version, soit sous forme de variante ayant subi certaines modifications. Si le changement est purement accessoire, il peut être admis que la fonction en question ne soit pas du tout exécutée et que les entités d'application homologues n'aient pas besoin de connaître les effets des changements. Pour des changements mineurs, un nouveau contexte d'application n'est pas nécessaire.
- *Changements importants au protocole INAP dans de futures versions normalisées:*
 un changement majeur peut être défini comme un changement de fonctionnalité qui est essentiel pour un service de réseau intelligent demandé. En cas de modification d'une fonction existante, les deux entités d'application auront des connaissances partagées concernant une variante fonctionnelle demandée. Si le changement est purement accessoire, le service de réseau intelligent demandé ne sera pas réalisé si une des entités d'application n'apporte pas de fonctionnalité additionnelle. Pour des changements importants, un nouveau contexte d'application est nécessaire.

- *Changements au protocole INAP spécifiques au réseau:*

ces ajouts peuvent avoir pour le service un caractère majeur ou mineur. Aucun nouveau contexte d'application n'est supposé être défini pour ce type de changement. Au moment de la définition, de tels ajouts ne seraient pas inclus sous une forme identique dans des versions subséquentes des Recommandations.

0.7.2 Définition des mécanismes de compatibilité du protocole INAP

0.7.2.1 Procédures relatives à des changements majeurs apportés à INAP

Afin de favoriser l'introduction de changements fonctionnels majeurs, le protocole permet une synchronisation entre les deux applications en ce qui concerne la fonctionnalité devant être réalisée. Cette synchronisation a lieu avant qu'on fasse appel à la nouvelle fonction pour chacune des entités d'application, afin d'éviter des procédures compliquées de repli. La solution choisie pour atteindre une telle synchronisation est d'utiliser les procédures de négociation du contexte d'application indiquées dans la Recommandation Q.773.

0.7.2.2 Procédures pour les ajouts mineurs apportés à INAP

Le marqueur de mécanisme d'extension devra être utilisé pour de futurs ajouts mineurs normalisés qui seront apportés au protocole INAP. Ce mécanisme permet différentes mises en œuvre des extensions en incluant «un marqueur d'extensions» dans la définition du type. Ces extensions sont exprimées par des champs optionnels qui sont placés derrière le marqueur. Quand une entité reçoit des paramètres inconnus après le marqueur, ils ne sont pas pris en compte (voir les Recommandations de la série X.680).

0.7.2.3 Procédures en vue de l'inclusion d'ajouts spécifiques au réseau dans le protocole INAP

Ce mécanisme est basé sur la capacité de déclarer explicitement des champs de tous types par l'option Macro d'après la syntaxe abstraite n° 1 (ASN.1) en dehors d'une opération type. Cela fonctionne en définissant un «Champ d'extension» qui est placé à la fin de la définition type. Ce champ d'extension est défini comme un ensemble d'extensions, une extension pouvant contenir n'importe quel type d'extension. Chaque extension est associée à une valeur qui définit si le nœud d'arrivée devrait ne tenir aucun compte du champ qui n'a pas été identifié ou refuser le message, ce qui est similaire au mécanisme nécessaire à la compréhension décrit à l'article précédent. Il convient de se référer à la Recommandation Q.1400 pour la définition de ce mécanisme.

En ce qui concerne l'interface SCF-vers-SDF, les mécanismes d'extension sont définis en 2.2.

1 Règles applicables aux entités SACF/MACF

1.1 Réflexion du contexte d'application du sous-système TCAP

Les règles de négociation du contexte d'application du sous-système TCAP imposent de refléter le contexte d'application, s'il est acceptable, dans le premier message vers l'arrière.

Si le contexte d'application n'est pas acceptable et que l'utilisateur des capacités de transaction ne souhaite pas poursuivre le dialogue, il peut fournir au demandeur un autre contexte d'application pouvant servir pour lancer un nouveau dialogue.

La négociation du contexte d'application ne s'applique qu'aux interfaces SCF.

Les Recommandations de la série Q.770 (*sous-système application pour la gestion des transactions*) donnent une description plus détaillée du mécanisme de négociation du contexte d'application du sous-système TCAP.

1.2 Exécution séquentielle/parallèle des opérations

Dans certains cas, il peut être nécessaire de savoir si les opérations doivent être exécutées séquentiellement ou en parallèle (synchronisées). Les opérations que l'on peut synchroniser sont:

- les opérations de taxation, qui peuvent être synchronisées avec d'autres opérations.

Pour indiquer que les opérations doivent être synchronisées, on les inclut dans le même message. Lorsqu'une des opérations précitées ne doit pas être exécutée avant la fin ou un certain stade d'avancement d'une autre, l'entité physique émettrice (généralement le sous-système SCP) peut commander cette exécution en envoyant ces opérations dans deux messages distincts.

Cette méthode n'impose pas l'exécution simultanée de toutes les opérations contenues dans le même message, mais indique simplement que ces opérations doivent être synchronisées là où la synchronisation a un sens (dans la situation identifiée ci-dessus).

En cas d'incompatibilité entre les règles génériques mentionnées ci-dessus et les règles spécifiques applicables aux entités fonctionnelles spécifiées à l'article 3, les règles spécifiques relatives aux entités fonctionnelles ont la priorité sur les règles génériques.

2 Syntaxe abstraite du protocole d'application dans l'ensemble de capacités 1 du réseau intelligent

Le présent article spécifie la syntaxe abstraite du protocole d'application dans l'ensemble CS-1 du réseau intelligent d'après la syntaxe abstraite n° 1 (ASN.1) définie dans les Recommandations X.208 et X.680.

Les règles de codage qui s'appliquent à cette syntaxe sont les règles de codage de base de la syntaxe ASN.1, définies dans la Recommandation X.209, y compris les restrictions précisées en 4.1.1/Q.773. D'autres codages sont cités pour des paramètres utilisés dans les Recommandations existantes traitant du sous-système ISUP (Q.763) et du système DSS 1 (Q.931).

En ce qui concerne les paramètres ISUP et les paramètres DSS 1 utilisés dans le protocole INAP, seule la valeur du paramètre sera codée comme cela est défini dans le sous-ensemble ISUP ou le système DSS 1. Les identificateurs de paramètres DSS 1/ISUP définis sont éliminés et remplacés par les identificateurs de paramètres définis par le protocole INAP.

La mise en correspondance des macros OPERATION et ERROR avec les composantes du sous-système TCAP est définie dans la Recommandation Q.773. La classe d'une opération n'est pas fixée explicitement mais est spécifiée dans la MACRO OPERATION en ASN.1 comme suit:

- Classe 1 – RESULT et ERRORS apparaissent toutes les deux dans la définition ASN.1 de la MACRO OPERATION.
- Classe 2 – Seulement ERRORS apparaît dans la définition ASN.1 de la MACRO OPERATION.
- Classe 3 – Seulement RESULT apparaît dans la définition ASN.1 de la MACRO OPERATION.
- Classe 4 – Ni ERRORS ni RESULT n'apparaissent dans la définition ASN.1 de la MACRO OPERATION.

Ces classes correspondent respectivement aux classes 2 à 5 spécifiées dans les Recommandations X.219 et Q.932.

La syntaxe abstraite du protocole INAP comprend plusieurs modules en notation ASN.1 décrivant des opérations, des erreurs et les types de données associés dont les valeurs (codes d'opération et codes d'erreur) sont définies dans un module distinct.

Le module contenant toutes les définitions des types des opérations INAP est **IN-CS-1-Operations** et est décrit en 2.1.1 et 2.2.2.4.1.

Le module contenant toutes les définitions des types des erreurs INAP est **IN-CS-1-Errors** et est décrit en 2.1.2. et en 2.2.2.4.1.

Le module contenant toutes les définitions des types de données INAP est **IN-CS-1-DataTypes** et est décrit en 2.1.3 et en 2.2.2.4.1.

Le module contenant les codes d'opérations et d'erreurs pour le protocole est **IN-CS-1-Codes** et est décrit en 2.1.4 et en 2.2.2.4.1.

2.1 SSF-SCF, SCF-SRF Interface

2.1.1 IN CS-1 Operation Types

IN-CS-1-Operations { ccitt recommendation q 1218 modules(0) cs-1-operations(0) version1(0) }

- This module contains the type definitions for the IN CS-1 operations.
- There may be functional redundancies in the operation set related to call processing.
- This may make product interworking more difficult. Administrations wishing to deploy
- IN and equipment manufacturers implementing IN should take this into account.

DEFINITIONS ::=

BEGIN

IMPORTS

OPERATION

FROM TCAPMessages { ccitt recommendation q 773 modules(0) messages(1) version2(2) }

-- *error types*

Cancelled,
CancelFailed,
ETCFailed,
ImproperCallerResponse,
MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
RequestedInfoError,
SystemFailure,
TaskRefused,
UnavailableResource,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter,
UnknownLegID,
UnknownResource

FROM IN-CS-1-Errors { ccitt recommendation q 1218 modules(0) cs-1-errors(1) version1(0) }

-- *argument types*

ActivateServiceFilteringArg,
AnalysedInformationArg,
AnalyseInformationArg,
ApplyChargingArg,
ApplyChargingReportArg,
AssistRequestInstructionsArg,
CallGapArg,
CallInformationReportArg,
CallInformationRequestArg,
CancelArg,
CancelStatusReportRequestArg,
CollectedInformationArg,
CollectInformationArg,
ConnectArg,
ConnectToResourceArg,
EstablishTemporaryConnectionArg,
EventNotificationChargingArg,
EventReportBCSMArg,
FurnishChargingInformationArg,
HoldCallInNetworkArg,
InitialDPArg,
InitiateCallAttemptArg,
MidCallArg,
OAnswerArg,
OCalledPartyBusyArg,
ODisconnectArg,
ONoAnswerArg,
OriginationAttemptAuthorizedArg,
PlayAnnouncementArg,
PromptAndCollectUserInformationArg,
ReceivedInformationArg,
ReleaseCallArg,
RequestCurrentStatusReportArg,
RequestCurrentStatusReportResultArg,
RequestEveryStatusChangeReportArg,
RequestFirstStatusMatchReportArg,
RequestNotificationChargingEventArg,
RequestReportBCSMEEventArg,
ResefTimerArg,
RouteSelectFailureArg,
SelectFacilityArg,
SelectRouteArg,
SendChargingInformationArg,
ServiceFilteringResponseArg,
SpecializedResourceReportArg,

StatusReportArg,
TAnswerArg,
TBusyArg,
TDisconnectArg,
TermAttemptAuthorizedArg,
TNoAnswerArg,

FROM IN-CS-1-DataTypes { ccitt recommendation q 1218 modules(0) cs-1-datatypes(2) version1(0) };

-- TYPE DEFINITIONS FOR IN CS-1 OPERATIONS FOLLOWS

-- SCF-SSF operations

ActivateServiceFiltering ::= OPERATION

ARGUMENT

ActivateServiceFilteringArg

RESULT

ERRORS {

MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedParameter
}

-- Direction: SCF -> SSF, Timer: T_{asf}

-- When receiving this operation, the SSF handles calls to destination in a specified manner
-- without sending queries for every detected call. It is used for example for providing
-- televoiting or mass calling services. Simple registration functionality (counters) and
-- announcement control may be located at the SSF. The operation initializes the specified
-- counters in the SSF.

ActivityTest ::= OPERATION

RESULT

-- Direction: SCF -> SSF, Timer: T_{at}

-- This operation is used to check for the continued existence of a relationship between the SCF
-- and SSF. If the relationship is still in existence, then the SSF will respond. If no reply is
-- received, then the SCF will assume that the SSF has failed in some way and will take the
-- appropriate action.

AnalysedInformation ::= OPERATION

ARGUMENT

AnalysedInformationArg

ERRORS {

MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SSF -> SCF, Timer: T_{adi}

-- This operation is used to indicate availability of routing address and call type. (DP 3 –
-- *Analysed_Info*).
-- For additional information on this operation and its use with open numbering plans, refer to
-- 4.2.2.2a)3)/Q.1214.

AnalyseInformation ::= OPERATION

ARGUMENT

AnalyseInformationArg

ERRORS {

MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
}

**UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

- *Direction: SCF → SSF, Timer: T_{ai}*
- *This operation is used to request the SSF to perform the originating basic call processing actions*
- *to analyse destination information that is either collected from a calling party or provided by the SCF*
- *(e.g. for number translation). This includes actions to validate the information according to an office*
- *or customized dialling plan, and if valid, to determine call termination information, to include the called*
- *party address, the type of call (e.g. intra-network or inter-network), and carrier (if inter-network).*
- *If the called party is not served by the SSF, the SSF also determines a route index based on the called*
- *party address and class of service, where the route index points to a list of outgoing trunk groups.*

ApplyCharging ::= OPERATION

ARGUMENT

ApplyChargingArg

ERRORS {

**MissingParameter,
UnexpectedComponentSequence,
UnexpectedParameter,
UnexpectedDataValue,
ParameterOutOfRange,
SystemFailure,
TaskRefused
}**

- *Direction: SCF → SSF, Timer: T_{ac}*
- *This operation is used for interacting from the SCF with the SSF charging mechanisms. The*
- *ApplyChargingReport operation provides the feedback from the SSF to the SCF.*

ApplyChargingReport ::= OPERATION

ARGUMENT

ApplyChargingReportArg

ERRORS {

**MissingParameter,
UnexpectedComponentSequence,
UnexpectedParameter,
UnexpectedDataValue,
ParameterOutOfRange,
SystemFailure,
TaskRefused
}**

- *Direction: SSF → SCF, Timer: T_{acr}*
- *This operation is used by the SSF to report to the SCF the occurrence of a specific charging event as*
- *requested by the SCF using the ApplyCharging operation.*

AssistRequestInstructions ::= OPERATION

ARGUMENT

AssistRequestInstructionsArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

- *Direction: SSF → SCF or SRF → SCF, Timer: T_{ari}*
- *This operation is used when there is an assist or a hand-off procedure and may be sent by the SSF*
- *or SRF to the SCF. This operation is sent by the assisting SSF or assisting SRF to the SCF, when the*
- *initiating SSF has set up a connection to the SRF or to the assisting SSF as a result of receiving*
- *an EstablishTemporaryConnection or Connect operation (in the case of hand-off) from the SCF.*
- *Refer to clause 3 for a description of the procedures associated with this operation.*

```

CallGap ::= OPERATION
  ARGUMENT
    CallGapArg
  -- Direction: SCF -> SSF, Timer: Tcg
  -- This operation is used to request the SSF to reduce the rate at which specific service requests are sent to
  -- the SCF. Use of this operation by the SCF to gap queries and updates at the SDF is for further study.

CallInformationReport ::= OPERATION
  ARGUMENT
    CallInformationReportArg
  -- Direction: SSF -> SCF, Timer: Tcirp
  -- This operation is used to send specific call information for a single call to the SCF as requested by the
  -- SCF in a previous CallInformationRequest.

CallInformationRequest ::= OPERATION
  ARGUMENT
    CallInformationRequestArg
  ERRORS {
    MissingParameter,
    ParameterOutOfRange,
    RequestedInfoError,
    SystemFailure,
    TaskRefused,
    UnexpectedComponentSequence,
    UnexpectedDataValue,
    UnexpectedParameter
  }
  -- Direction: SCF -> SSF, Timer: Tcirq
  -- This operation is used to request the SSF to record specific information about a single call and report it to
  -- the SCF (with a CallInformationReport operation).

Cancel ::= OPERATION
  ARGUMENT
    CancelArg
  ERRORS {
    CancelFailed,
    MissingParameter,
    TaskRefused
  }
  -- Direction: SCF -> SSF, or SCF -> SRF, Timer: Tcan
  -- This generic operation cancels the correlated previous operation or all previous requests. The following
  -- operations can be canceled: PlayAnnouncement, PromptAndCollectUserInformation.

CancelStatusReportRequest ::= OPERATION
  ARGUMENT
    CancelStatusReportRequestArg
  ERRORS {
    CancelFailed,
    MissingParameter,
    TaskRefused
  }
  -- Direction: SCF -> SSF, Timer: Tcsr
  -- This operation cancels the following processes: RequestFirstStatusMatchReport and
  -- RequestEveryStatusChangeReport.

CollectedInformation ::= OPERATION
  ARGUMENT
    CollectedInformationArg
  ERRORS {
    MissingCustomerRecord,
    MissingParameter,
    ParameterOutOfRange,
    SystemFailure,
    TaskRefused,
    UnexpectedComponentSequence,
    UnexpectedDataValue,
    UnexpectedParameter
  }

```

-- Direction: SSF → SCF, Timer: T_{cdi}
 -- This operation is used to indicate availability of complete initial information package/dialling string from
 -- originating party. (This event may have already occurred in the case of en bloc signalling, in which case
 -- the waiting duration in this PIC is zero.) (DP 2 – Collected_Info). For additional information on this
 -- operation and its use with open numbering plans, refer to 4.2.2.2.1-2/Q.1214.

CollectInformation ::= OPERATION

ARGUMENT

CollectInformationArg

ERRORS {

MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SCF → SSF, Timer: T_{ci}
 -- This operation is used to request the SSF to perform the originating basic call processing actions to
 -- prompt a calling party for destination information, then collect destination information according to a
 -- specified numbering plan (e.g. for virtual private networks).

Connect ::= OPERATION

ARGUMENT

ConnectArg

ERRORS {

MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SCF → SSF, Timer: T_{con}
 -- This operation is used to request the SSF to perform the call processing actions to route or forward a call
 -- to a specified destination. To do so, the SSF may or may not use destination information from the calling
 -- party (e.g. dialled digits) and existing call set-up information (e.g. route index to a list of trunk groups),
 -- depending on the information provided by the SCF.
 -- When address information is only included in the Connect operation, call processing resumes at PIC3 in
 -- the O-BCSM.
 -- When address information and routing information is included, call processing resumes at PIC4.

ConnectToResource ::= OPERATION

ARGUMENT

ConnectToResourceArg

ERRORS {

MissingParameter,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SCF → SSF, Timer: T_{ctr}
 -- This operation is used to connect a call from the SSP to the physical entity containing the SRF.
 -- Refer to clause 3 for a description of the procedures associated with this operation.

Continue ::= OPERATION

-- Direction: SCF → SSF, Timer: T_{cue}
 -- This operation is used to request the SSF to proceed with call processing at the DP at which it
 -- previously suspended call processing to await SCF instructions (i.e. proceed to the next point
 -- in call in the BCSM). The SSF continues call processing without substituting new data from SCF.

DisconnectForwardConnection ::= OPERATION

ERRORS {
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence
}

- *Direction: SCF → SSF, Timer: T_{dfc}*
- *This operation is used to disconnect a forward temporary connection or a connection to a resource.*
- *Refer to clause 3 for a description of the procedures associated with this operation.*

EstablishTemporaryConnection ::= OPERATION

ARGUMENT
 EstablishTemporaryConnectionArg
ERRORS {
 ETCFailed,
 MissingParameter,
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
}

- *Direction: SCF → SSF, Timer: T_{etc}*
- *This operation is used to create a connection to a resource for a limited period of time*
- *(e.g. to play an announcement, to collect user information); it implies the use of the assist*
- *procedure. Refer to clause 3 for a description of the procedures associated with this operation.*

EventNotificationCharging ::= OPERATION

ARGUMENT
 EventNotificationChargingArg

- *Direction: SSF → SCF, Timer: T_{enc}*
- *This operation is used by the SSF to report to the SCF the occurrence of a specific charging event*
- *type as previously requested by the SCF in a RequestNotificationChargingEvent operation.*

EventReportBCSM ::= OPERATION

ARGUMENT
 EventReportBCSMArg

- *Direction: SSF → SCF, Timer: T_{erb}*
- *This operation is used to notify the SCF of a call-related event (e.g. BCSM events such as busy or*
- *no answer) previously requested by the SCF in a RequestReportBCSMEvent operation.*

FurnishChargingInformation ::= OPERATION

ARGUMENT
 FurnishChargingInformationArg
ERRORS {
 MissingParameter,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
}

- *Direction: SCF → SSF, Timer: T_{fci}*
- *This operation is used to request the SSF to generate, register a call record or to include some*
- *information in the default call record. The registered call record is intended for off line charging of the*
- *call.*

HoldCallInNetwork ::= OPERATION

ARGUMENT
 HoldCallInNetworkArg
ERRORS {
 MissingParameter,
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
}

- Direction: SCF → SSF, Timer: T_{hcn}
- This operation is used to provide the capability of queueing a call during the set-up phase (e.g. to provide a call completion to busy, the call would be queued until the destination becomes free).

InitialDP ::= OPERATION

ARGUMENT

InitialDPArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**

}

- Direction: SSF → SCF, Timer: T_{idp}

- This operation is used after a TDP to indicate request for service.

InitiateCallAttempt ::= OPERATION

ARGUMENT

InitiateCallAttemptArg

ERRORS {

**MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**

}

- Direction: SCF → SSF, Timer: T_{ica}

- This operation is used to request the SSF to create a new call to one call party using address information provided by the SCF.

OAnswer ::= OPERATION

ARGUMENT

OAnswerArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**

}

- Direction: SSF → SCF, Timer: T_{oa}

- This operation is used for indication from the terminating half BCSM that the call is accepted and answered by terminating party (e.g. terminating party goes offhook, Q.931 Connect message received, ISDN-UP Answer message received) (DP 7 – O_Answer). For additional information on this operation, refer to 4.2.2.2.1-5/Q.1214.

OCalledPartyBusy ::= OPERATION

ARGUMENT

OCalledPartyBusyArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**

}

-- Direction: SSF -> SCF, Timer: T_{ob}
-- This operation is used for Indication from the terminating half BCSM that the terminating party is busy
-- (DP 5 – O_Called_Party_Busy). For additional information on this operation, refer to 4.2.2.2.1-4/Q.1214.

ODisconnect ::= OPERATION

ARGUMENT

ODisconnectArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**
}

-- Direction: SSF -> SCF, Timer: T_{od}
-- This operation is used for a disconnect indication (e.g. onhook, Q.931 Disconnect message, SS7 Release message) is received from the originating party, or received from the terminating party via the terminating half BCSM. (DP 9 – O_Disconnect). For additional information on this operation, refer to 4.2.2.2.1-5/Q.1214.

OMidCall ::= OPERATION

ARGUMENT

MidCallArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**
}

-- Direction: SSF -> SCF, Timer: T_{omc}
-- This operation is used to indicate a feature request is received from the originating party
-- (e.g. hook flash, ISDN feature activation, Q.931 HOLD or RETrieve message). (DP 8 – O_Mid_Call).
-- For additional information on this operation, refer to 4.2.2.2.1-5/Q.1214.

ONoAnswer ::= OPERATION

ARGUMENT

ONoAnswerArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter**
}

-- Direction: SSF -> SCF, Timer: T_{ona}
-- This operation is used for indication from the terminating half BCSM that the terminating party does not answer within a specified time period (DP 6 – O_No_Answer). For additional information on this operation, refer to 4.2.2.2.1-4/Q.1214.

OriginationAttemptAuthorized ::= OPERATION

ARGUMENT

OriginationAttemptAuthorizedArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,**

**TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

- Direction: SSF → SCF, Timer: T_{oaa}
- This operation is used to indicate the desire to place outgoing call (e.g. offhook, Q.931 Set-up message, ISDN-UP IAM message) and authority/ability to place outgoing call verified
- (DP 1 – Origination_Attempt_Authorized). For additional information on this operation, refer to
- 4.2.2.2.1-1/Q.1214.

**ReleaseCall ::= OPERATION
ARGUMENT**

ReleaseCallArg

- Direction: SCF → SSF, Timer: T_{rc}
- This operation is used to tear down an existing call at any phase of the call for all parties involved in the call.

**RequestCurrentStatusReport ::= OPERATION
ARGUMENT**

RequestCurrentStatusReportArg

RESULT

RequestCurrentStatusReportResultArg

ERRORS {

**MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedParameter,
UnknownResource
}**

- Direction: SCF → SSF, Timer: T_{rcs}
- This operation is used to request the SSF to report immediately the busy/idle status of a physical termination resource.

**RequestEveryStatusChangeReport ::= OPERATION
ARGUMENT**

RequestEveryStatusChangeReportArg

RESULT

ERRORS {

**MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedParameter,
UnknownResource
}**

- Direction: SCF → SSF, Timer: T_{res}
- This operation is used to request the SSF to report every change of busy/idle status of a physical termination resource.

**RequestFirstStatusMatchReport ::= OPERATION
ARGUMENT**

RequestFirstStatusMatchReportArg

RESULT

ERRORS {

**MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedParameter,
UnknownResource
}**

-- Direction: SCF → SSF, Timer: T_{rfs}
 -- This operation is used to request the SSF to report the first change busy/idle to the specified status of
 -- a physical termination resource.

RequestNotificationChargingEvent ::= OPERATION

ARGUMENT

RequestNotificationChargingEventArg

ERRORS {

MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SCF → SSF, Timer: T_{rnc}
 -- This operation is used by the SCF to instruct the SSF on how to manage the charging events
 -- which are received from other FE's and not under control of the service logic instance.

RequestReportBCSMEvent ::= OPERATION

ARGUMENT

RequestReportBCSMEventArg

ERRORS {

MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SCF → SSF, Timer: T_{rrb}
 -- This operation is used to request the SSF to monitor for a call-related event (e.g. BCSM events such as
 -- busy or no answer), then send a notification back to the SCF when the event is detected.
 -- It is proposed that Event Detection Point (EDP) processing be always initiated by RequestReportBCSMEvent
 -- and the EDP may be acknowledged with either an EventReportBCSM or by a DP-specific operations:
 -- NOTE – The application context should identify whether Request Report BCSM Event ASE and DP
 -- Generic BCSM EventReport ASE are being used, or whether Request Report BCSM EventASE,
 -- Basic BCP DP ASE, and Advanced BCP DP ASE are being used.
 -- – For a particular IN, only one of the two alternatives identified by the respective ASEs should be
 -- selected (i.e. only one approach should be selected for a given application context).
 -- – Further study is required to identify the small set of parameters required to be conveyed for EDPs
 -- when the Basic BCP DP ASE and Advanced BCP DP ASE are used.
 -- – For CS2 further study should be given for the feasibility of progressing of one of both
 -- alternatives for both TDPs and EDPs.
 -- – Every EDP must be explicitly armed by the SCF via a RequestReportBCSMEvent operation. No
 -- implicit arming of EDPs at the SSF after reception of any operation (different from
 -- RequestReportBCSMEvent) from the SCF is allowed.

ResetTimer ::= OPERATION

ARGUMENT

ResetTimerArg

ERRORS {

MissingParameter,
ParameterOutOfRange,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- Direction: SCF → SSF, Timer: T_{rt}
 -- This operation is used to request the SSF to refresh an application timer in the SSF.

RouteSelectFailure ::= OPERATION

ARGUMENT

RouteSelectFailureArg

ERRORS {

**MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- Direction: SSF -> SCF, Timer: T_{rsf}

*-- This operation is used to indicate that the SSP is unable to select a route (e.g. unable to determine a correct route, no more routes on route list) or indication from the terminating half BCSM that a call cannot be presented to the terminating party (e.g. network congestion) (DP 4 – Route_Select_Failure).
-- For additional information on this operation, refer to 4.2.2.2.1-4/Q.1214.*

SelectFacility ::= OPERATION

ARGUMENT

SelectFacilityArg

ERRORS {

**MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- Direction: SCF -> SSF, Timer: T_{sf}

-- This operation is used to request the SSF to perform the terminating basic call processing actions to select the terminating line if it is idle, or selects an idle line from a multi-line hunt group, or selects an idle trunk from a trunk group, as appropriate. If no idle line or trunk is available, the SSF determines that the terminating facility is busy.

SelectRoute ::= OPERATION

ARGUMENT

SelectRouteArg

ERRORS {

**MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- Direction: SCF -> SSF, Timer: T_{sr}

*-- This operation is used to request the SSF to perform the originating basic call processing actions to determine routing information and select a route for a call, based either on call information available to the SSF, or on call information provided by the SCF (e.g. for alternate routing), to include the called party address, type of call, carrier, route index, and one or more alternate route indices.
-- Based on the routing information, the SSF attempts to select a primary route for the call, and if the route is busy, attempts to select an alternate route. The SSF may fail to select a route for the call if all routes are busy.*

SendChargingInformation ::= OPERATION

ARGUMENT

SendChargingInformationArg

ERRORS {

**MissingParameter,
UnexpectedComponentSequence,
UnexpectedParameter,
}**

```

    ParameterOutOfRange,
    SystemFailure,
    TaskRefused,
    UnknownLegID
}
-- Direction: SCF -> SSF, Timer: Tsci
-- This operation is used to instruct the SSF on the charging information to send by the SSF. The charging
-- information can either be sent back by means of signalling or internal if the SSF is located in the local
-- exchange. In the local exchange this information may be used to update the charge meter or to create a
-- standard call record.

ServiceFilteringResponse ::= OPERATION
    ARGUMENT
        ServiceFilteringResponseArg
-- Direction: SSF -> SCF, Timer: Tsfr
-- This operation is used to send back to the SCF the values of counters specified in a previous
-- ActivateServiceFiltering operation.

StatusReport ::= OPERATION
    ARGUMENT
        StatusReportArg
-- Direction: SSF -> SCF, Timer: Tsrp
-- This operation is used as a response to RequestFirstStatusMatchReport or
-- RequestEveryStatusChangeReport operations.

TAnswer ::= OPERATION
    ARGUMENT
        TAnswerArg
    ERRORS {
        MissingCustomerRecord,
        MissingParameter,
        ParameterOutOfRange,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
    }
-- Direction: SSF -> SFC, Timer: Tta
-- This operation is used to indicate that the call is accepted and answered by terminating party
-- (e.g. terminating party goes offhook, Q.931 Connect message received, ISDN-UP Answer message
-- received) (DP 15 - TAnswer). For additional information on this operation, refer to 4.2.2.2-10/Q.1214.

TBusy ::= OPERATION
    ARGUMENT
        TBusyArg
    ERRORS {
        MissingCustomerRecord,
        MissingParameter,
        ParameterOutOfRange,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
    }
-- Direction: SSF -> SCF, Timer: Ttb
-- This operation is used to indicate all resources in group busy (DP 13 - TBusy).
-- For additional information on this operation, refer to 4.2.2.2-8/Q.1214.

TDisconnect ::= OPERATION
    ARGUMENT
        TDisconnectArg
    ERRORS {
        MissingCustomerRecord,
        MissingParameter,
        ParameterOutOfRange,

```

SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

- Direction: SSF → SCF, Timer: T_{td}
- This operation is used for a disconnect indication (e.g. onhook, Q.931 Disconnect message,
- SS7 Release message) is received from the terminating party, or received from the originating party
- via the originating half BCSM. (DP 17 – $T_{Disconnect}$.) For additional information on this operation,
- refer to 4.2.2.2.2-10/Q.1214.

TermAttemptAuthorized ::= OPERATION
ARGUMENT

TermAttemptAuthorizedArg

ERRORS {

MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

- Direction: SSF → SCF, Timer: T_{taa}
- This operation is used for indication of incoming call received from originating half BCSM and authority
- to route call to a specified terminating resource (or group) verified. (DP 12 – $Termination_Authorized$.)
- For additional information on this operation, refer to 4.2.2.2.2-7/Q.1214.

TMidCall ::= OPERATION

ARGUMENT

MidCallArg

ERRORS {

MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

- Direction: SSF → SCF, Timer: T_{tmc}
- This operation is used to indicate that a feature request is received from the terminating party (e.g. hook
- flash, ISDN feature activation Q.931 HOLD or RETRIEVE message). (DP 16 – T_{Mid_Call} .)
- For additional information on this operation, refer to 4.2.2.2.2-10/Q.1214.

TNoAnswer ::= OPERATION

ARGUMENT

TNoAnswerArg

ERRORS {

MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

- Direction: SSF → SCF, Timer: T_{tna}
- This operation is used to indicate that the terminating party does not answer within a specified duration.
- (DP 14 – T_{No_Answer} .) For additional information on this operation, refer to 4.2.2.2.2-9/Q.1214.

-- *SCF-SRF operations*
 -- *AssistRequestInstructions*
 -- *SRF → SCF*
 -- *Refer to previous description of this operation in the SCF-SSF operations clause.*
 -- *Cancel*
 -- *SCF → SRF*
 -- *Refer to previous description of this operation in the SCF-SSF operations clause.*

PlayAnnouncement ::= OPERATION

ARGUMENT

PlayAnnouncementArg

ERRORS {

Cancelled,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter,
UnavailableResource

}

LINKED {

SpecializedResourceReport

}

-- *Direction: SCF → SRF, Timer: T_{pa}*
 -- *This operation is to be used after Establish Temporary Connection (assist procedure with a second SSP)*
 -- *or a Connect to Resource (no assist) operation. It may be used for inband interaction with an analogue*
 -- *user, or for interaction with an ISDN user. In the former case, the SRF is usually collocated with the SSF*
 -- *for standard tones (congestion tone...) or standard announcements. In the latter case, the SRF is always*
 -- *collocated with the SSF in the switch. Any error is returned to the SCF. The timer associated with this*
 -- *operation must be of a sufficient duration to allow its linked operation to be correctly correlated.*

PromptAndCollectUserInformation ::= OPERATION

ARGUMENT

PromptAndCollectUserInformationArg

RESULT

ReceivedInformationArg

ERRORS {

Cancelled,
ImproperCallerResponse,
MissingParameter,
ParameterOutOfRange,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnavailableResource,
UnexpectedDataValue,
UnexpectedParameter

}

-- *Direction: SCF → SRF, Timer: T_{pc}*
 -- *This operation is used to interact with a user to collect information.*

SpecializedResourceReport ::= OPERATION

ARGUMENT

SpecializedResourceReportArg

-- *Direction: SRF → SCF, Timer: T_{srr}*
 -- *This operation is used as the response to a PlayAnnouncement operation when the announcement*
 -- *completed report indication is set.*

END

Les intervalles de valeurs suivantes s'appliquent aux temporisations propres aux opérations du protocole INAP:

Court: 1-10 secondes

Moyen: 1-60 secondes

Long: 1 seconde-30 minutes

Le Tableau 2 énumère tous les temporisateurs de fonctionnement et les intervalles de temporisation correspondants. La valeur définitive pour chaque temporisateur peut être propre au réseau et doit être dévinie par l'exploitant du réseau.

TABLEAU 2/Q.1218

Nom de l'opération	Temporisateur	Intervalle de temporisation
ActivateServiceFiltering	T _{asf}	moyen
ActivityTest	T _{at}	court
AnalyzeInformation	T _{ai}	à étudier
AnalyzedInformation	T _{adi}	à étudier
ApplyCharging	T _{ac}	court
ApplyChargingReport	T _{acr}	court
AssistRequestInstructions	T _{ari}	court
CallGap	T _{cg}	court
CallInformationReport	T _{cirp}	court
CallInformationRequest	T _{cirq}	court
Cancel	T _{can}	court
CancelStatusReportRequest	T _{csr}	à étudier
CollectedInformation	T _{cdi}	à étudier
CollectInformation	T _{ci}	moyen
Connect	T _{con}	court
ConnectToResource	T _{ctr}	court
Continue	T _{cue}	court
DisconnectForwardConnection	T _{dfc}	court
EstablishTemporaryConnection	T _{etc}	moyen
EventNotificationCharging	T _{enc}	court
EventReportBCSM	T _{erb}	court
FurnishChargingInformation	T _{fci}	court
HoldCallInNetwork	T _{hcn}	à étudier
InitialDP	T _{idp}	court
InitiateCallAttempt	T _{ica}	court
OAnswer	T _{oa}	à étudier
OCalledPartyBusy	T _{ob}	à étudier
ODisconnect	T _{od}	à étudier
OMidCall	T _{omc}	à étudier
ONoAnswer	T _{ona}	à étudier
OriginationAttemptAuthorized	T _{oaa}	à étudier
ReleaseCall	T _{rc}	court

TABLEAU 2/Q.1218 (fin)

Nom de l'opération	Temporisateur	Intervalle de temporisation
RequestCurrentStatusReport	T _{rcs}	à étudier
RequestEveryStatusChangeReport	T _{res}	court
RequestFirstStatusMatchReport	T _{rfs}	court
RequestNotificationChargingEvent	T _{rnc}	court
RequestReportBCSMEEvent	T _{rrb}	court
ResetTimer	T _{rt}	court
RouteSelectFailure	T _{rsf}	à étudier
SelectFacility	T _{sf}	à étudier
SelectRoute	T _{sr}	à étudier
SendChargingInformation	T _{sci}	court
ServiceFilteringResponse	T _{sfr}	court
StatusReport	T _{srp}	à étudier
TAnswer	T _{ta}	à étudier
TBusy	T _{tb}	à étudier
TDisconnect	T _{td}	à étudier
TermAttemptAuthorized	T _{taa}	à étudier
TMidCall	T _{tmc}	à étudier
TNoAnswer	T _{tna}	à étudier
PlayAnnouncement	T _{pa}	long
PromptAndCollectUserInfo	T _{pc}	long
SpecializedResourceReport	T _{srr}	court

2.1.2 IN CS-1 Error Types

IN-CS-1-Errors { ccitt recommendation q 1218 modules(0) cs-1-errors(1) version1(0) }

-- This module contains the type definitions for the IN CS-1 errors.

-- Where a parameter of type CHOICE is tagged with a specific tag value, the tag is automatically

-- replaced with an EXPLICIT tag of the same value.

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

ERROR

FROM TCAPMessages { ccitt recommendation q 773 modules(0) messages(1) version2(2) }

InvokeID,

UnavailableNetworkResource

FROM IN-CS-1-DataTypes { ccitt recommendation q 1218 modules(0) cs-1-datatypes(2) version1(0) };

-- TYPE DEFINITION FOR IN CS-1 ERRORS FOLLOWS

Cancelled ::= ERROR

-- The operation has been cancelled.

```

CancelFailed ::= ERROR
    PARAMETER SEQUENCE {
        problem      [0] ENUMERATED {
            unknownOperation(0),
            tooLate(1),
            operationNotCancellable(2)
        },
        operation    [1] InvokeID
    }
-- The operation failed to be cancelled.

ETCFailed ::= ERROR
-- The establish temporary connection failed.

ImproperCallerResponse ::= ERROR
-- The caller response was not as expected.

MissingCustomerRecord ::= ERROR
-- The Service Logic Program could not be found in the SCF.

MissingParameter ::= ERROR
-- An expected optional parameter was not received.

ParameterOutOfRange ::= ERROR
-- The parameter was not as expected (e.g. missing or out of range).

RequestedInfoError ::= ERROR
    PARAMETER ENUMERATED {
        unknownRequestedInfo(1),
        requestedInfoNotAvailable(2)
        -- other values FFS
    }
-- The requested information cannot be found.

SystemFailure ::= ERROR
    PARAMETER
        unavailableNetworkResource UnavailableNetworkResource
-- The operation could not be completed due to a system failure at the serving physical entity.

TaskRefused ::= ERROR
    PARAMETER ENUMERATED {
        generic(0),
        unobtainable (1),
        congestion(2)
        -- other values FFS
    }
-- An entity normally capable of the task requested cannot or chooses not to perform the task at this
-- time. This includes error situations like congestion and unobtainable address as used in (e.g. the
-- connect operation).

UnavailableResource ::= ERROR
-- A requested resource is not available at the serving entity.

UnexpectedComponentSequence ::= ERROR
-- An incorrect sequence of Components was received (e.g. "DisconnectForwardConnection"
-- followed by "PlayAnnouncement").

UnexpectedDataValue ::= ERROR
-- The data value was not as expected (e.g. routing number expected but billing number received).

UnexpectedParameter ::= ERROR
-- A parameter received was not expected.

UnknownLegID ::= ERROR
-- Leg not known to the SSF.

UnknownResource ::= ERROR
-- Resource whose status is being requested is not known to the serving entity.

END

```

2.1.3 IN CS-1 Data Types

EXTENSION MACRO ::=

BEGIN

```
TYPE NOTATION ::= ExtensionType Criticality
VALUE NOTATION ::= value(INTEGER)
ExtensionType ::= "EXTENSION-SYNTAX" type
Criticality ::= "CRITICALITY" value(CriticalityType)
CriticalityType ::= ENUMERATED {
                        ignore(0),
                        abort(1)
                        }
```

END

```
-- Example of addition of an extension named 'Some Network Specific Indicator' of type
-- BOOLEAN, with criticality 'abort' and to be identified as extension number 1.
-- Example of definition using the above macro:
--
-- SomeNetworkSpecificIndicator ::= EXTENSION
-- EXTENSION-SYNTAX BOOLEAN
-- CRITICALITY abort
--
-- someNetworkSpecificIndicator SomeNetworkSpecificIndicator ::= 1
--
-- Example of transfer syntax, using the ExtensionField datatype as specified in the module
-- below. Assuming the value of the extension is set to TRUE, the extensions parameter
-- becomes a Sequence of type INTEGER ::= 1, criticality ENUMERATED ::= 1 and value [1]
-- EXPLICIT BOOLEAN ::= TRUE.
--
-- Use of Q.1400 defined Extension is ffs.
-- In addition the extension mechanism marker is used to identify the future minor additions to INAP.
```

IN-CS-1-datatypes { ccitt recommendation q 1218 modules(0) cs-1-datatypes(2) version1(0) }

```
-- This module contains the type definitions for the IN CS-1 data types.
-- Where a parameter of type CHOICE is tagged with a specific tag value, the tag is automatically
-- replaced with an EXPLICIT tag of the same value.
-- The following parameters map onto bearer protocol (i.e. Q.931, case 2 and ISUP) parameters:
-- CallingPartySubaddress, CalledPartyNumber,
-- Prefix (derived from dialled digits), DestinationRoutingAddress,
-- DialledDigits, ISDNAccessRelatedInformation, CallingPartysCategory, LocationNumber,
-- TravellingClassMark, AssistingSSPIPRoutingAddress, AlertingPattern (Q.931 only),
-- ReleaseCause (and other Cause parameters), ServiceProfileIdentifier (Q.932 only),
-- BearerCapability, CallingPartyNumber, HighLayerCompatibility, OriginalCalledPartyID,
-- RedirectingPartyID, and RedirectionInformation.
-- The procedures for mapping of parameters onto bearer protocol are ffs.
--
-- The following SSF parameters do not map onto bearer protocol (i.e. Q.931, case 2 and ISUP)
-- parameters and therefore are assumed to be local to the switching system: CallingPartyBusinessGroupID
-- FacilityGroup, FacilityGroupMember, RouteList, LegID, IPSSPCapabilities, IPAvailable, CGEncountered,
-- ForwardingCondition, CorrelationID, ApplicationTimer, TerminalType, MiscCallInfo, TriggerType and
-- ServiceKey.
--
-- Where possible, Administrations should specify the maximum size within their network of
-- parameters specified in this Recommendation that are of an indeterminate length.
```

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

InvokeIdType

FROM TCAPMessages {ccitt recommendation q.773 modules(2) messages(1) version2(2)};

-- TYPE DEFINITIONS FOR IN CS-1 DATA TYPES FOLLOWS

-- Argument data types

-- The ordering of parameters in the argument sequences has been arbitrary. Further study may be required to order arguments in a manner which will facilitate efficient encoding and decoding.

```
ActivateServiceFilteringArg ::= SEQUENCE {
    filteredCallTreatment      [0] FilteredCallTreatment,
    filteringCharacteristics   [1] FilteringCharacteristics,
    filteringTimeOut           [2] FilteringTimeOut,
    filteringCriteria          [3] FilteringCriteria,
    startTime                  [4] DateAndTime                OPTIONAL,
    extensions                  [5] SEQUENCE SIZE(1..numOfExtensions) OF
                                ExtensionField                OPTIONAL
--
    ...
}
```

```
AnalysedInformationArg ::= SEQUENCE {
    dpSpecificCommonParameters [0] DpSpecificCommonParameters,
    dialledDigits               [1] CalledPartyNumber                OPTIONAL,
    callingPartyBusinessGroupID [2] CallingPartyBusinessGroupID  OPTIONAL,
    callingPartySubaddress      [3] CallingPartySubaddress        OPTIONAL,
    callingFacilityGroup        [4] FacilityGroup                  OPTIONAL,
    callingFacilityGroupMember  [5] FacilityGroupMember        OPTIONAL,
    originalCalledPartyID       [6] OriginalCalledPartyID        OPTIONAL,
    prefix                       [7] Digits                        OPTIONAL,
    redirectingPartyID          [8] RedirectingPartyID            OPTIONAL,
    redirectionInformation       [9] RedirectionInformation        OPTIONAL,
    routeList                   [10] RouteList                     OPTIONAL,
    travellingClassMark         [11] TravellingClassMark           OPTIONAL,
    extensions                   [12] SEQUENCE SIZE(1..numOfExtensions) OF
                                ExtensionField                OPTIONAL,
    featureCode                 [13] FeatureCode                   OPTIONAL,
    accessCode                   [14] AccessCode                   OPTIONAL,
    carrier                       [15] Carrier                     OPTIONAL
--
    ...
}
```

-- For the OPTIONAL parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

```
AnalyseInformationArg ::= SEQUENCE {
    destinationRoutingAddress [0] DestinationRoutingAddress,
    alertingPattern           [1] AlertingPattern                OPTIONAL,
    iSDNAccessRelatedInformation [2] ISDNAccessRelatedInformation  OPTIONAL,
    originalCalledPartyID     [3] OriginalCalledPartyID        OPTIONAL,
    extensions                 [4] SEQUENCE SIZE(1..numOfExtensions) OF
                                ExtensionField                OPTIONAL,
    callingPartyNumber        [5] CallingPartyNumber            OPTIONAL,
    callingPartysCategory     [6] CallingPartysCategory         OPTIONAL,
    calledPartyNumber         [7] CalledPartyNumber             OPTIONAL,
    chargeNumber              [8] ChargeNumber                  OPTIONAL,
    travellingClassMark       [9] TravellingClassMark           OPTIONAL,
    carrier                    [10] Carrier                     OPTIONAL
--
    ...
}
```

```
ApplyChargingArg ::= SEQUENCE {
    aChBillingChargingCharacteristics [0] AChBillingChargingCharacteristics,
    partyToCharge                     [2] LegID                    OPTIONAL,
    extensions                         [3] SEQUENCE SIZE(1..numOfExtensions) OF
                                ExtensionField                OPTIONAL
--
    ...
}
```

-- The partyToCharge parameter indicates the party in the call to which the ApplyCharging operation should be applied. If it is not present, then it is applied to the A-party.

travellingClassMark	[10] TravellingClassMark	OPTIONAL,
extensions	[11] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
featureCode	[12] FeatureCode	OPTIONAL,
accessCode	[13] AccessCode	OPTIONAL,
carrier	[14] Carrier	OPTIONAL
--	...	
	}	

-- For the OPTIONAL parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

CollectInformationArg ::= SEQUENCE {

alertingPattern	[0] AlertingPattern	OPTIONAL,
numberingPlan	[1] NumberingPlan	OPTIONAL,
originalCalledPartyID	[2] OriginalCalledPartyID	OPTIONAL,
travellingClassMark	[3] TravellingClassMark	OPTIONAL,
extensions	[4] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
callingPartyNumber	[5] CallingPartyNumber	OPTIONAL,
dialledDigits	[6] CalledPartyNumber	OPTIONAL
--	...	
	}	

ConnectArg ::= SEQUENCE {

destinationRoutingAddress	[0] DestinationRoutingAddress,	
alertingPattern	[1] AlertingPattern	OPTIONAL,
correlationID	[2] CorrelationID	OPTIONAL,
cutAndPaste	[3] CutAndPaste	OPTIONAL,
forwardingCondition	[4] ForwardingCondition	OPTIONAL,
iSDNAccessRelatedInformation	[5] ISDNAccessRelatedInformation	OPTIONAL,
originalCalledPartyID	[6] OriginalCalledPartyID	OPTIONAL,
routeList	[7] RouteList	OPTIONAL,
scfID	[8] ScfID	OPTIONAL,
travellingClassMark	[9] TravellingClassMark	OPTIONAL,
extensions	[10] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
carrier	[11] Carrier	OPTIONAL,
serviceInteractionIndicators	[26] ServiceInteractionIndicators	OPTIONAL,
callingPartyNumber	[27] CallingPartyNumber	OPTIONAL,
callingPartysCategory	[28] CallingPartysCategory	OPTIONAL,
redirectingPartyID	[29] RedirectingPartyID	OPTIONAL,
redirectionInformation	[30] RedirectionInformation	OPTIONAL
--	...	
	}	

-- For alerting pattern, OPTIONAL denotes that this parameter only applies if SSF is the terminating
-- local exchange for the subscriber.

ConnectToResourceArg ::= SEQUENCE {

resourceAddress	CHOICE {	
ipRoutingAddress	[0] IPRoutingAddress,	
legID	[1] LegID,	
both	[2] SEQUENCE {	
	ipRoutingAddress [0] IPRoutingAddress,	
	legID [1] LegID	
	},	
none	[3] NULL	
},		
extensions	[4] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
serviceInteractionIndicators	[30] ServiceInteractionIndicators	OPTIONAL
--	...	
	}	

DpSpecificCommonParameters ::= SEQUENCE {

serviceAddressInformation	[0] ServiceAddressInformation,	
bearerCapability	[1] BearerCapability	OPTIONAL,
calledPartyNumber	[2] CalledPartyNumber	OPTIONAL,
callingPartyNumber	[3] CallingPartyNumber	OPTIONAL,
callingPartysCategory	[4] CallingPartysCategory	OPTIONAL,

```

iPSSPCapabilities      [5] IPSSPCapabilities      OPTIONAL,
iPAavailable           [6] IPAavailable           OPTIONAL,
iSDNAAccessRelatedInformation [7] ISDNAAccessRelatedInformation OPTIONAL,
cGEncountered         [8] CGEncountered         OPTIONAL,
locationNumber        [9] LocationNumber         OPTIONAL,
serviceProfileIdentifier [10] ServiceProfileIdentifier OPTIONAL,
terminalType          [11] TerminalType          OPTIONAL,
extensions             [12] SEQUENCE SIZE(1..numOfExtensions) OF
                        ExtensionField      OPTIONAL,
chargeNumber          [13] ChargeNumber          OPTIONAL,
servingAreaID        [14] ServingAreaID          OPTIONAL
--
...
}
-- OPTIONAL for iPSSPCapabilities, iPAavailable, and cGEncountered denotes network operator
-- specific use. OPTIONAL for callingPartyNumber, and callingPartysCategory refer to clause 3 for
-- the trigger detection point processing rules to specify when these parameters are included in the
-- message. bearerCapability should be appropriately coded as speech.

EstablishTemporaryConnectionArg ::= SEQUENCE {
    assistingSSPIPRoutingAddress [0] AssistingSSPIPRoutingAddress,
    correlationID                [1] CorrelationID                OPTIONAL,
    legID                        [2] LegID                        OPTIONAL,
    scfID                        [3] ScfID                        OPTIONAL,
    extensions                    [4] SEQUENCE SIZE(1..numOfExtensions) OF
                                ExtensionField      OPTIONAL,
    carrier                      [5] Carrier                      OPTIONAL,
    serviceInteractionIndicators [30] ServiceInteractionIndicators OPTIONAL
--
...
}

EventNotificationChargingArg ::= SEQUENCE {
    eventTypeCharging [0] EventTypeCharging,
    eventSpecificInformationCharging [1] EventSpecificInformationCharging OPTIONAL,
    legID [2] LegID OPTIONAL,
    extensions [3] SEQUENCE SIZE(1..numOfExtensions) OF
                ExtensionField OPTIONAL,
    monitorMode [30] MonitorMode DEFAULT notifyAndContinue
--
...
}
-- OPTIONAL denotes network operator specific use.

EventReportBCSMArg ::= SEQUENCE {
    eventTypeBCSM [0] EventTypeBCSM,
    bcsmEventCorrelationID [1] CorrelationID OPTIONAL,
    eventSpecificInformationBCSM [2] EventSpecificInformationBCSM OPTIONAL,
    legID [3] LegID OPTIONAL,
    miscCallInfo [4] MiscCallInfo DEFAULT
                {messageType request},
    extensions [5] SEQUENCE SIZE(1..numOfExtensions) OF
                ExtensionField OPTIONAL
--
...
}

FurnishChargingInformationArg ::= FCIBillingChargingCharacteristics

HoldCallInNetworkArg ::= CHOICE {
    holdcause [0] HoldCause,
    empty [1] NULL
}
-- holdcause is optional and denotes network operator specific use.

InitialDPArg ::= SEQUENCE {
    serviceKey [0] ServiceKey OPTIONAL,
    dialledDigits [1] CalledPartyNumber OPTIONAL,
    calledPartyNumber [2] CalledPartyNumber OPTIONAL,
    callingPartyNumber [3] CallingPartyNumber OPTIONAL,
    callingPartyBusinessGroupID [4] CallingPartyBusinessGroupID OPTIONAL,
    callingPartysCategory [5] CallingPartysCategory OPTIONAL,
    callingPartySubaddress [6] CallingPartySubaddress OPTIONAL

```

cGEncountered	[7]	CGEncountered	OPTIONAL,
iPSSPCapabilities	[8]	IPSSPCapabilities	OPTIONAL,
iPAvailable	[9]	IPAvailable	OPTIONAL,
locationNumber	[10]	LocationNumber	OPTIONAL,
miscCallInfo	[11]	MiscCallInfo	OPTIONAL,
originalCalledPartyID	[12]	OriginalCalledPartyID	OPTIONAL,
serviceProfileIdentifier	[13]	ServiceProfileIdentifier	OPTIONAL,
terminalType	[14]	TerminalType	OPTIONAL,
extensions	[15]	SEQUENCE SIZE(1..numOfExtensions)	OF
		ExtensionField	OPTIONAL,
triggerType	[16]	TriggerType	OPTIONAL,
highLayerCompatibility	[23]	HighLayerCompatibility	OPTIONAL,
serviceInteractionIndicators	[24]	ServiceInteractionIndicators	OPTIONAL,
additionalCallingPartyNumber	[25]	AdditionalCallingPartyNumber	OPTIONAL,
forwardCallIndicators	[26]	ForwardCallIndicators	OPTIONAL,
bearerCapability	[27]	BearerCapability	OPTIONAL,
eventTypeBCSM	[28]	EventTypeBCSM	OPTIONAL,
redirectingPartyID	[29]	RedirectingPartyID	OPTIONAL,
redirectionInformation	[30]	RedirectionInformation	OPTIONAL

-- ...
-- }
-- OPTIONAL for iPSSPCapabilities, iPAvailable, cGEncountered, and miscCallInfo denotes network
-- operator specific use.
-- OPTIONAL for dialledDigits, callingPartyNumber, and callingPartysCategory refer to clause 3 for the
-- trigger detection point processing rules to specify when these parameters are included in the message.
-- OPTIONAL for terminalType indicates that this parameter applies only at originating or terminating
-- local exchanges if the SSF has this information.

InitiateCallAttemptArg ::= SEQUENCE {

destinationRoutingAddress	[0]	DestinationRoutingAddress,	
alertingPattern	[1]	AlertingPattern	OPTIONAL,
iSDNAccessRelatedInformation	[2]	ISDNAccessRelatedInformation	OPTIONAL,
travellingClassMark	[3]	TravellingClassMark	OPTIONAL,
extensions	[4]	SEQUENCE SIZE(1..numOfExtensions)	OF
		ExtensionField	OPTIONAL
serviceInteractionIndicators	[29]	ServiceInteractionIndicators	OPTIONAL,
callingPartyNumber	[30]	CallingPartyNumber	OPTIONAL

-- ...
-- }

MidCallArg ::= SEQUENCE {

dpSpecificCommonParameters	[0]	DpSpecificCommonParameters,	
calledPartyBusinessGroupID	[1]	CalledPartyBusinessGroupID	OPTIONAL,
calledPartySubaddress	[2]	CalledPartySubaddress	OPTIONAL,
callingPartyBusinessGroupID	[3]	CallingPartyBusinessGroupID	OPTIONAL,
callingPartySubaddress	[4]	CallingPartySubaddress	OPTIONAL,
featureRequestIndicator	[5]	FeatureRequestIndicator	OPTIONAL,
extensions	[6]	SEQUENCE SIZE(1..numOfExtensions)	OF
		ExtensionField	OPTIONAL,
carrier	[7]	Carrier	OPTIONAL

-- ...
-- }

-- For the OPTIONAL parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

OAnswerArg ::= SEQUENCE {

dpSpecificCommonParameters	[0]	DpSpecificCommonParameters,	
callingPartyBusinessGroupID	[1]	CallingPartyBusinessGroupID	OPTIONAL,
callingPartySubaddress	[2]	CallingPartySubaddress	OPTIONAL,
callingFacilityGroup	[3]	FacilityGroup	OPTIONAL,
callingFacilityGroupMember	[4]	FacilityGroupMember	OPTIONAL,
originalCalledPartyID	[5]	OriginalCalledPartyID	OPTIONAL,
redirectingPartyID	[6]	RedirectingPartyID	OPTIONAL,
redirectionInformation	[7]	RedirectionInformation	OPTIONAL,

routeList	[8] RouteList	OPTIONAL,
travellingClassMark	[9] TravellingClassMark	OPTIONAL,
extensions	[10] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL
--	...	
	}	

-- For the *OPTIONAL* parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

OCalledPartyBusyArg ::= SEQUENCE {

dpSpecificCommonParameters	[0] DpSpecificCommonParameters,	
busyCause	[1] Cause	OPTIONAL,
callingPartyBusinessGroupID	[2] CallingPartyBusinessGroupID	OPTIONAL,
callingPartySubaddress	[3] CallingPartySubaddress	OPTIONAL,
callingFacilityGroup	[4] FacilityGroup	OPTIONAL,
callingFacilityGroupMember	[5] FacilityGroupMember	OPTIONAL,
originalCalledPartyID	[6] OriginalCalledPartyID	OPTIONAL,
prefix	[7] Digits	OPTIONAL,
redirectingPartyID	[8] RedirectingPartyID	OPTIONAL,
redirectionInformation	[9] RedirectionInformation	OPTIONAL,
routeList	[10] RouteList	OPTIONAL,
travellingClassMark	[11] TravellingClassMark	OPTIONAL,
extensions	[12] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
carrier	[13] Carrier	OPTIONAL
--	...	
	}	

-- For the *OPTIONAL* parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

ODisconnectArg ::= SEQUENCE {

dpSpecificCommonParameters	[0] DpSpecificCommonParameters,	
callingPartyBusinessGroupID	[1] CallingPartyBusinessGroupID	OPTIONAL,
callingPartySubaddress	[2] CallingPartySubaddress	OPTIONAL,
callingFacilityGroup	[3] FacilityGroup	OPTIONAL,
callingFacilityGroupMember	[4] FacilityGroupMember	OPTIONAL,
releaseCause	[5] Cause	OPTIONAL,
routeList	[6] RouteList	OPTIONAL,
extensions	[7] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
carrier	[8] Carrier	OPTIONAL,
connectTime	[9] Integer4	OPTIONAL
--	...	
	}	

-- For the *OPTIONAL* parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

ONoAnswerArg ::= SEQUENCE {

dpSpecificCommonParameters	[0] DpSpecificCommonParameters,	
callingPartyBusinessGroupID	[1] CallingPartyBusinessGroupID	OPTIONAL,
callingPartySubaddress	[2] CallingPartySubaddress	OPTIONAL,
callingFacilityGroup	[3] FacilityGroup	OPTIONAL,
callingFacilityGroupMember	[4] FacilityGroupMember	OPTIONAL,
originalCalledPartyID	[5] OriginalCalledPartyID	OPTIONAL,
prefix	[6] Digits	OPTIONAL,
redirectingPartyID	[7] RedirectingPartyID	OPTIONAL,
redirectionInformation	[8] RedirectionInformation	OPTIONAL,
routeList	[9] RouteList	OPTIONAL,
travellingClassMark	[10] TravellingClassMark	OPTIONAL,
extensions	[11] SEQUENCE SIZE(1..numOfExtensions)	OF
	ExtensionField	OPTIONAL,
carrier	[12] Carrier	OPTIONAL
--	...	
	}	

-- For the *OPTIONAL* parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

```

OriginationAttemptAuthorizedArg ::= SEQUENCE {
    dpSpecificCommonParameters [0] DpSpecificCommonParameters,
    dialledDigits [1] CalledPartyNumber OPTIONAL,
    callingPartyBusinessGroupID [2] CallingPartyBusinessGroupID OPTIONAL,
    callingPartySubaddress [3] CallingPartySubaddress OPTIONAL,
    callingFacilityGroup [4] FacilityGroup OPTIONAL,
    callingFacilityGroupMember [5] FacilityGroupMember OPTIONAL,
    travellingClassMark [6] TravellingClassMark OPTIONAL,
    extensions [7] SEQUENCE SIZE(1..numOfExtensions) OF
    ExtensionField OPTIONAL,
    carrier [8] Carrier OPTIONAL
--
    ...
}
-- For the OPTIONAL parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

PlayAnnouncementArg ::= SEQUENCE {
    informationToSend [0] InformationToSend,
    disconnectFromIPForbidden [1] BOOLEAN DEFAULT TRUE,
    requestAnnouncementComplete [2] BOOLEAN DEFAULT TRUE,
    extensions [3] SEQUENCE SIZE(1..numOfExtensions) OF
    ExtensionField OPTIONAL
--
    ...
}

PromptAndCollectUserInfoArg ::= SEQUENCE {
    collectedInfo [0] CollectedInfo,
    disconnectFromIPForbidden [1] BOOLEAN DEFAULT TRUE,
    informationToSend [2] InformationToSend OPTIONAL,
    extensions [3] SEQUENCE SIZE(1..numOfExtensions) OF
    ExtensionField OPTIONAL
--
    ...
}

ReceivedInformationArg ::= CHOICE {
    digitsResponse [0] Digits,
    iA5Response [1] IA5String
}

ReleaseCallArg ::= Cause
-- A default value of decimal 31 (normal unspecified) should be coded appropriately.

RequestCurrentStatusReportArg ::= ResourceID

RequestCurrentStatusReportResultArg ::= SEQUENCE {
    resourceStatus [0] ResourceStatus,
    resourceID [1] ResourceID OPTIONAL,
    extensions [2] SEQUENCE SIZE(1..numOfExtensions) OF
    ExtensionField OPTIONAL
--
    ...
}

RequestEveryStatusChangeReportArg ::= SEQUENCE {
    resourceID [0] ResourceID,
    correlationID [1] CorrelationID OPTIONAL,
    monitorDuration [2] Duration OPTIONAL,
    extensions [3] SEQUENCE SIZE(1..numOfExtensions) OF
    ExtensionField OPTIONAL
--
    ...
}
-- For correlationID OPTIONAL denotes network operator optional.
-- monitorDuration is required if outside the context of a call. It is not expected if we are in the context
-- of a call, because in that case the end of the call implicitly means the end of the monitoring.

RequestFirstStatusMatchReportArg ::= SEQUENCE {
    resourceID [0] ResourceID OPTIONAL,
    resourceStatus [1] ResourceStatus OPTIONAL,
    correlationID [2] CorrelationID OPTIONAL,

```

```

    monitorDuration          [3] Duration                OPTIONAL,
    extensions                [4] SEQUENCE SIZE(1..numOfExtensions) OF
                               ExtensionField            OPTIONAL,
    bearerCapability          [5] BearerCapability        OPTIONAL
--
    ...
}

```

-- For correlationID OPTIONAL denotes network operator optional.
-- monitorDuration is required if outside the context of a call. It is not expected if we are in the context
-- of a call, because in that case the end of the call implicitly means the end of the monitoring.

```

RequestNotificationChargingEventArg ::= SEQUENCE SIZE(1..numOfChargingEvents)
    OF ChargingEvent

```

```

RequestReportBCSMEEventArg ::= SEQUENCE {
    bcsmEvents                [0] SEQUENCE SIZE(1..numOfBCSMEEvents) OF
                               BCSMEEvent,
    bcsmEventCorrelationID    [1] CorrelationID                OPTIONAL,
    extensions                [2] SEQUENCE SIZE(1..numOfExtensions) OF
                               ExtensionField                OPTIONAL
--
    ...
}

```

-- Indicates the BCSM related events for notification.
-- For correlationID OPTIONAL denotes network operator optional.

```

ResetTimerArg ::= SEQUENCE {
    timerID                   [0] TimerID                    DEFAULT tssf,
    timervalue                [1] TimerValue,
    extensions                [2] SEQUENCE SIZE(1..numOfExtensions) OF
                               ExtensionField                OPTIONAL
--
    ...
}

```

```

RouteSelectFailureArg ::= SEQUENCE {
    dpSpecificCommonParameters [0] DpSpecificCommonParameters,
    dialledDigits              [1] CalledPartyNumber          OPTIONAL,
    callingPartyBusinessGroupID [2] CallingPartyBusinessGroupID  OPTIONAL,
    callingPartySubaddress      [3] CallingPartySubaddress    OPTIONAL,
    callingFacilityGroup        [4] FacilityGroup              OPTIONAL,
    callingFacilityGroupMember  [5] FacilityGroupMember       OPTIONAL,
    failureCause                [6] Cause                     OPTIONAL,
    originalCalledPartyID       [7] OriginalCalledPartyID     OPTIONAL,
    prefix                      [8] Digits                    OPTIONAL,
    redirectingPartyID          [9] RedirectingPartyID        OPTIONAL,
    redirectionInformation      [10] RedirectionInformation    OPTIONAL,
    routeList                   [11] RouteList                 OPTIONAL,
    travellingClassMark         [12] TravellingClassMark       OPTIONAL,
    extensions                  [13] SEQUENCE SIZE(1..numOfExtensions) OF
                               ExtensionField                OPTIONAL,
    carrier                     [14] Carrier                  OPTIONAL
--
    ...
}

```

-- For the OPTIONAL parameters, refer to clause 3 for the trigger detection point processing
-- rules to specify when these parameters are included in the message.

```

SelectFacilityArg ::= SEQUENCE {
    alertingPattern            [0] AlertingPattern            OPTIONAL,
    destinationNumberRoutingAddress [1] CalledPartyNumber    OPTIONAL,
    iSDNAAccessRelatedInformation [2] ISDNAAccessRelatedInformation  OPTIONAL,
    calledFacilityGroup        [3] FacilityGroup              OPTIONAL,
    calledFacilityGroupMember  [4] FacilityGroupMember       OPTIONAL,
    originalCalledPartyID       [5] OriginalCalledPartyID     OPTIONAL,
    extensions                  [6] SEQUENCE SIZE(1..numOfExtensions) OF
                               ExtensionField                OPTIONAL
--
    ...
}

```

-- OPTIONAL parameters are only provided if modifications desired to basic call processing values.

```

SelectRouteArg ::= SEQUENCE {
    destinationRoutingAddress [0] DestinationRoutingAddress,
    alertingPattern [1] AlertingPattern OPTIONAL,
    correlationID [2] CorrelationID OPTIONAL,
    iSDNAccessRelatedInformation [3] ISDNAccessRelatedInformation OPTIONAL,
    originalCalledPartyID [4] OriginalCalledPartyID OPTIONAL,
    routeList [5] RouteList OPTIONAL,
    scfID [6] ScfID OPTIONAL,
    travellingClassMark [7] TravellingClassMark OPTIONAL,
    extensions [8] SEQUENCE SIZE(1..numOfExtensions) OF
        ExtensionField OPTIONAL,
    carrier [9] Carrier OPTIONAL
--
    ...
}

```

-- OPTIONAL parameters are only provided if modifications desired to basic call processing values.

```

SendChargingInformationArg ::= SEQUENCE {
    sCIBillingChargingCharacteristics [0] SCIBillingChargingCharacteristics,
    partyToCharge [1] LegID,
    extensions [2] SEQUENCE SIZE(1..numOfExtensions) OF
        ExtensionField OPTIONAL
--
    ...
}

```

```

ServiceFilteringResponseArg ::= SEQUENCE {
    countersValue [0] CountersValue,
    filteringCriteria [1] FilteringCriteria,
    extensions [2] SEQUENCE SIZE(1..numOfExtensions) OF
        ExtensionField OPTIONAL,
    responseCondition [3] ResponseCondition OPTIONAL
--
    ...
}

```

SpecializedResourceReportArg ::= NULL

```

StatusReportArg ::= SEQUENCE {
    resourceStatus [0] ResourceStatus OPTIONAL,
    correlationID [1] CorrelationID OPTIONAL,
    resourceID [2] ResourceID OPTIONAL,
    extensions [3] SEQUENCE SIZE(1..numOfExtensions) OF
        ExtensionField OPTIONAL,
    reportCondition [4] ReportCondition OPTIONAL
--
    ...
}

```

-- For correlationID, OPTIONAL denotes network operator optional.

-- resourceID is required when the SSF sends a report as an answer to a previous request when the

-- correlationID was present.

```

TAnswerArg ::= SEQUENCE {
    dpSpecificCommonParameters [0] DpSpecificCommonParameters,
    calledPartyBusinessGroupID [1] CalledPartyBusinessGroupID OPTIONAL,
    calledPartySubaddress [2] CalledPartySubaddress OPTIONAL,
    calledFacilityGroup [3] FacilityGroup OPTIONAL,
    calledFacilityGroupMember [4] FacilityGroupMember OPTIONAL,
    extensions [5] SEQUENCE SIZE(1..numOfExtensions) OF
        ExtensionField OPTIONAL
--
    ...
}

```

```

TBusyArg ::= SEQUENCE {
    dpSpecificCommonParameters [0] DpSpecificCommonParameters,
    busyCause [1] Cause OPTIONAL,
    calledPartyBusinessGroupID [2] CalledPartyBusinessGroupID OPTIONAL,
    calledPartySubaddress [3] CalledPartySubaddress OPTIONAL,
    originalCalledPartyID [4] OriginalCalledPartyID OPTIONAL,
    redirectingPartyID [5] RedirectingPartyID OPTIONAL

```

redirectionInformation	[6]	RedirectionInformation	OPTIONAL,
routeList	[7]	RouteList	OPTIONAL,
travellingClassMark	[8]	TravellingClassMark	OPTIONAL,
extensions	[9]	SEQUENCE SIZE(1..numOfExtensions) ExtensionField	OF OPTIONAL

-- ...
}

-- For the OPTIONAL parameters, refer to clause 3 for the trigger detection point processing rules
-- to specify when these parameters are included in the message.

TDisconnectArg ::= SEQUENCE {

dpSpecificCommonParameters	[0]	DpSpecificCommonParameters,	
calledPartyBusinessGroupID	[1]	CalledPartyBusinessGroupID	OPTIONAL,
calledPartySubaddress	[2]	CalledPartySubaddress	OPTIONAL,
calledFacilityGroup	[3]	FacilityGroup	OPTIONAL,
calledFacilityGroupMember	[4]	FacilityGroupMember	OPTIONAL,
releaseCause	[5]	Cause	OPTIONAL,
extensions	[6]	SEQUENCE SIZE(1..numOfExtensions) ExtensionField	OF OPTIONAL,
connectTime	[7]	Integer4	OPTIONAL

-- ...
}

TermAttemptAuthorizedArg ::= SEQUENCE {

dpSpecificCommonParameters	[0]	DpSpecificCommonParameters,	
calledPartyBusinessGroupID	[1]	CalledPartyBusinessGroupID	OPTIONAL,
calledPartySubaddress	[2]	CalledPartySubaddress	OPTIONAL,
callingPartyBusinessGroupID	[3]	CallingPartyBusinessGroupID	OPTIONAL,
originalCalledPartyID	[4]	OriginalCalledPartyID	OPTIONAL,
redirectingPartyID	[5]	RedirectingPartyID	OPTIONAL,
redirectionInformation	[6]	RedirectionInformation	OPTIONAL,
routeList	[7]	RouteList	OPTIONAL,
travellingClassMark	[8]	TravellingClassMark	OPTIONAL,
extensions	[9]	SEQUENCE SIZE(1..numOfExtensions) ExtensionField	OF OPTIONAL

-- ...
}

TNoAnswerArg ::= SEQUENCE {

dpSpecificCommonParameters	[0]	DpSpecificCommonParameters,	
calledPartyBusinessGroupID	[1]	CalledPartyBusinessGroupID	OPTIONAL,
calledPartySubaddress	[2]	CalledPartySubaddress	OPTIONAL,
calledFacilityGroup	[3]	FacilityGroup	OPTIONAL,
calledFacilityGroupMember	[4]	FacilityGroupMember	OPTIONAL,
originalCalledPartyID	[5]	OriginalCalledPartyID	OPTIONAL,
redirectingPartyID	[6]	RedirectingPartyID	OPTIONAL,
redirectionInformation	[7]	RedirectionInformation	OPTIONAL,
travellingClassMark	[8]	TravellingClassMark	OPTIONAL,
extensions	[9]	SEQUENCE SIZE(1..numOfExtensions) ExtensionField	OF OPTIONAL

-- ...
}

-- The Definition of Common Data Types

AccessCode ::= LocationNumber

-- An access code from a business group dialling plan attendant access codes, access codes to escape
-- to the public network, access code to access a private facility/network, and feature access codes.
-- Uses the LocationNumber format which is based on the Q.763 Location Number format.
-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).
-- The Numbering Plan Indicator field shall be set to "Spare" (value 000).
-- Of local significance.

**AChBillingChargingCharacteristics ::= OCTET STRING (SIZE (minAChBillingChargingLength..
maxAChBillingChargingLength))**

-- The AChBillingChargingCharacteristics parameter specifies the charging related information
-- to be provided by the SSF and the conditions on which this information has to be reported
-- back to the SCF with the ApplyChargingReport operation.
-- Examples of charging related information to be provided by the SSF may be: bulk counter
-- values, costs, tariff change and time of charge, time stamps, durations, etc.

-- Examples of conditions on which the charging related information are to be reported may be:
-- threshold value reached, timer expiration, tariff change, end of connection configuration, etc.

AdditionalCallingPartyNumber ::= Digits

-- Indicates the Additional Calling Party Number. Refer to Recommendation Q.763 for encoding.

AlertingPattern ::= OCTET STRING (SIZE(3))

-- Indicates a specific pattern that is used to alert a subscriber (e.g. distinctive ringing, tones, etc.).
-- Only applies if SSF is the terminating local exchange for the subscriber. Refer to the Q.931
-- Signal parameter for encoding.

ApplicationTimer ::= INTEGER (0..2047)

-- Used by the SCF to set a timer in the SSF. The timer is in seconds.

AssistingSSPIPRoutingAddress ::= Digits

-- Indicates the destination address of the SRF for the assist procedure.

BCSMEvent ::= SEQUENCE {

eventTypeBCSM	[0] EventTypeBCSM,	
monitorMode	[1] MonitorMode,	
legID	[2] LegID	OPTIONAL,
dpSpecificCriteria	[30] DpSpecificCriteria	OPTIONAL

}

-- Indicates the BCSM Event information for monitoring.

BearerCapability ::= CHOICE {

bearerCap	[0] OCTET STRING (SIZE(2..maxBearerCapabilityLength)),
tmr	[1] OCTET STRING (SIZE(1))

}

-- Indicates the type of bearer capability connection to the user. For bearerCapability, either
-- DSS 1 (Q.931) or the ISUP User Service Information (Q.763) encoding can be used. Refer
-- to the Q.763 Transmission Medium Requirement parameter for tmr encoding.

CalledPartyBusinessGroupID ::= OCTET STRING

-- Indicates the business group of the called party. The value of this octet string is network
-- operator specific.

**CalledPartyNumber ::= OCTET STRING (SIZE (minCalledPartyNumberLength..
maxCalledPartyNumberLength))**

-- Indicates the Called Party Number. Refer to Recommendation Q.763 for encoding.

CalledPartySubaddress ::= OCTET STRING

-- Indicates the Called Party Subaddress. Refer to Recommendation Q.931 for encoding.

CallingPartyBusinessGroupID ::= OCTET STRING

-- Indicates the business group of the calling party. The value of this octet string is network
-- operator specific.

**CallingPartyNumber ::= OCTET STRING (SIZE (minCallingPartyNumberLength..
maxCallingPartyNumberLength))**

-- Indicates the Calling Party Number. Refer to Recommendation Q.763 for encoding.

CallingPartySubaddress ::= OCTET STRING

-- Indicates the Calling Party Subaddress. Refer to Recommendation Q.931 for encoding.

CallingPartysCategory ::= OCTET STRING (SIZE(1))

-- Indicates the type of calling party (e.g. operator, payphone, ordinary subscriber).
-- Refer to Recommendation Q.763 for encoding.

**CallResult ::= OCTET STRING (SIZE (minCallResultLength..
maxCallResultLength))**

-- This parameter provides the SCF with the charging related information previously requested
-- using the ApplyCharging operation. This shall include the partyToCharge parameter as
-- received in the related ApplyCharging operation to correlate the result to the request.
-- The remaining content is network operator specific.

- Examples of charging related information to be provided by the SSF may be: bulk counter values,
- costs, tariff change and time of change, time stamps, durations, etc.
- Examples of conditions on which the charging related information are to be reported may be:
- threshold value reached, timer expiration, tariff change, end of connection configuration, etc.

Carrier ::= OCTET STRING

- Contains the carrier selection and carrier ID fields.
- Carrier selection is one octet and is encoded as:
- 00000000 No indication
- 00000001 Selected carrier code pre subscribed and not input by calling party
- 00000010 Selected carrier identification code pre subscribed and input by calling party
- 00000011 Selected carrier identification code pre subscribed, no indication of whether input by calling party
- 00000100 Selected carrier identification code not pre subscribed and input by calling party
- 00000101
- to Spare
- 11111110
- 11111111 Reserved
-
- Carrier ID has a one octet field indicating the number of digits followed by the digits encoded using BCD.
- Detailed coding is for further study. It is of local significance and carrying it through the ISUP is for further
- study.

Cause ::= OCTET STRING (SIZE (minCauseLength..maxCauseLength))

- Indicates the cause for interface related information. Refer to the Q.763 Cause parameter for
- encoding.
- For the use of cause and location values refer to Recommendation Q.850.

CGEncountered ::= ENUMERATED {

```

    noCGEncountered(0),
    manualCGEncountered(1),
    scpOverload(2)
}
```

- Indicates the type of automatic call gapping encountered, if any.

ChargeNumber ::= LocationNumber

- Information sent in either direction indicating the chargeable number for the call and consisting
- of the odd/even indicator, nature of address indicator, numbering plan indicator, and address signals.
- Uses the LocationNumber format which is based on the Q.763 Location Number format.
- For example, the ChargeNumber may be a third party number to which a call is billed for the 3rd party
- billing service. In this case, the calling party may request operator assistance to charge the call to,
- for example, their home number.

ChargingEvent ::= SEQUENCE {

```

    eventTypeCharging          [0] EventTypeCharging,
    monitorMode                [1] MonitorMode,
    legID                       [2] LegID                OPTIONAL
}
```

- This parameter indicates the charging event type and corresponding monitor mode and LegID.

CollectedDigits ::= SEQUENCE {

```

    minimumNbOfDigits          [0] INTEGER (1..127)          DEFAULT 1,
    maximumNbOfDigits          [1] INTEGER (1..127),
    endOfReplyDigit            [2] OCTET STRING (SIZE (1..2))    OPTIONAL,
    cancelDigit                 [3] OCTET STRING (SIZE (1..2))    OPTIONAL,
    startDigit                  [4] OCTET STRING (SIZE (1..2))    OPTIONAL,
    firstDigitTimeOut           [5] INTEGER (1..127)              OPTIONAL,
    interDigitTimeOut           [6] INTEGER (1..127)              OPTIONAL,
    errorTreatment              [7] ErrorTreatment                DEFAULT
                                reportErrorToScf,
    interruptableAnnInd         [8] BOOLEAN                       DEFAULT TRUE,
    voiceInformation            [9] BOOLEAN                       DEFAULT FALSE,
    voiceBack                   [10] BOOLEAN                     DEFAULT FALSE
}
```



```

DpSpecificCriteria ::= CHOICE {
    numberOfDigits           [0] NumberOfDigits,
    applicationTimer       [1] ApplicationTimer
}
-- The SCF may specify the number of digits to be collected by the SSF for the CollectedInfo event.
-- When all digits are collected, the SSF reports the event to the SCF.
-- The SCF may set a timer in the SSF for the No Answer event. If the user does not answer the call
-- within the allotted time, the SSF reports the event to the SCF.

```

```

Duration ::= INTEGER (-2..86400)
-- Values are seconds.

```

```

ErrorTreatment ::= ENUMERATED {
    reportErrorToScf(0),
    help(1),
    repeatPrompt(2)
}
-- reportErrorToScf means returning the "ImproperCallerResponse" error in the event of an error
-- condition during collection of user info.

```

```

EventSpecificInformationBCSM ::= CHOICE {
    collectedInfoSpecificInfo   [0] SEQUENCE {
        calledPartynumber       [0] CalledPartyNumber
        -- ... --
    },
    analyzedInfoSpecificInfo    [1] SEQUENCE {
        calledPartynumber       [0] CalledPartyNumber
        -- ... --
    },
    routeSelectFailureSpecificInfo [2] SEQUENCE {
        failureCause           [0] Cause           OPTIONAL
        -- ... --
    },
    oCalledPartyBusySpecificInfo [3] SEQUENCE {
        busyCause              [0] Cause           OPTIONAL
        -- ... --
    },
    oNoAnswerSpecificInfo       [4] SEQUENCE {
        -- no specific info defined --
        -- ... --
    },
    oAnswerSpecificInfo        [5] SEQUENCE {
        -- no specific info defined --
        -- ... --
    },
    oMidCallSpecificInfo       [6] SEQUENCE {
        connectTime            [0] Integer4       OPTIONAL
        -- ... --
    },
    oDisconnectSpecificInfo    [7] SEQUENCE {
        releaseCause           [0] Cause           OPTIONAL,
        connectTime           [1] Integer4       OPTIONAL
        -- ... --
    },
    tBusySpecificInfo          [8] SEQUENCE {
        busyCause              [0] Cause           OPTIONAL
        -- ... --
    },
    tNoAnswerSpecificInfo     [9] SEQUENCE {
        -- no specific info defined --
        -- ... --
    },
    tAnswerSpecificInfo       [10] SEQUENCE {
        -- no specific info defined --
        -- ... --
    },

```

```

tMidCallSpecificInfo      [11] SEQUENCE {
    connectTime           [0] Integer4    OPTIONAL
    -- ... --
    },
tDisconnectSpecificInfo  [12] SEQUENCE {
    releaseCause          [0] Cause       OPTIONAL,
    connectTime           [1] Integer4    OPTIONAL
    -- ... --
    }
}

```

-- Indicates the call related information specific to the event.
-- The connectTime indicates the duration between the received answer indication from the called party side
-- and the release of the connection for ODisconnect, OException, TDisconnect, or TException events.
-- The unit for the connectTime is 100 milliseconds.

EventSpecificInformationCharging ::= OCTET STRING (SIZE (minEventSpecificInformationChargingLength..maxEventSpecificInformationChargingLength))

-- defined by network operator.
-- Indicates the charging related information specific to the event.
-- An example data type definition for this parameter is given below:
-- chargePulses [0] Integer4,
-- chargeMessages [1] OCTET STRING (SIZE (min..max))

EventTypeBCSM ::= ENUMERATED {
origAttemptAuthorized(1),
collectedInfo(2),
analysedInformation(3),
routeSelectFailure(4),
oCalledPartyBusy(5),
oNoAnswer(6),
oAnswer(7),
oMidCall(8),
oDisconnect(9),
oAbandon(10),
termAttemptAuthorized(12),
tBusy(13),
tNoAnswer(14),
tAnswer(15),
tMidCall(16),
tDisconnect(17),
tAbandon(18)
}

-- Indicates the BCSM detection point event. Refer to 4.2.2.2/Q.1214 for additional information on the
-- events. Values origAttemptAuthorized and termAttemptAuthorized can only be used for TDPs.

EventTypeCharging ::= OCTET STRING (SIZE (minEventTypeChargingLength..maxEventTypeChargingLength))

-- This parameter indicates the charging event type. Its content is network operator specific.
--
-- An example data type definition for this parameter is given below:
-- EventTypeCharging ::= ENUMERATED {
-- chargePulses (0),
-- chargeMessages (1)
-- }

ExtensionField ::= SEQUENCE {
type INTEGER, -- shall identify the value of an EXTENSION type
criticality ENUMERATED {
ignore (0),
abort (1)
} DEFAULT ignore,
value [1] ANY DEFINED BY type
}

-- This parameter indicates an extension of an argument data type. Its content is network operator specific.

```

FacilityGroup ::= CHOICE {
    trunkGroupID [0] INTEGER,
    privateFacilityID [1] INTEGER,
    huntGroup [2] OCTET STRING,
    routeIndex [3] OCTET STRING
}

```

-- Indicates the particular group of facilities to route the call. huntGroup and routeIndex are encoded as network operator specific.

```

FacilityGroupMember ::= INTEGER

```

-- Indicates the specific member of a trunk group or multi-line hunt group.

```

FCIBillingChargingCharacteristics ::= OCTET STRING (SIZE (minFCIBillingChargingLength..
maxFCIBillingChargingLength))

```

-- This parameter indicates the billing and/or charging characteristics. Its content is network operator specific. An example datatype definition for this parameter is given below:

```

FCIBillingChargingCharacteristics ::= CHOICE {
    completeChargingrecord [0] OCTET STRING (SIZE (min..max)),
    correlationID [1] CorrelationID,
    scenario2Dot3 [2] SEQUENCE {
        chargeParty [0] LegID OPTIONAL,
        chargeLevel [1] OCTET STRING (SIZE (min..max))
OPTIONAL,
        chargeItems [2] SET OF Attribute OPTIONAL
}
}

```

-- Depending on the applied charging scenario, the following information elements can be included (refer to Q.1214 Appendix II):

- complete charging record (scenario 2.2)
- charge party (scenario 2.3)
- charge level (scenario 2.3)
- charge items (scenario 2.3)
- correlationID (scenario 2.4)

```

FeatureCode ::= LocationNumber

```

- The two-digit feature code preceded by "*" or "11".
- Uses the LocationNumber format which is based on the Q.763 Location Number format.
- The Nature of Address indicator field shall be set to "Spare" (value 00000000).
- The Numbering Plan Indicator field shall be set to "Spare" (value 000).
- Used for stimulus signalling (Q.932).

```

FeatureRequestIndicator ::= ENUMERATED {

```

```

    hold(0),
    retrieve(1),
    featureActivation(2),
    spare1(3),
    sparen(127)
}

```

-- Indicates the feature activated (e.g. a switch-hook flash, feature activation). Spare values reserved for future use.

```

FilteredCallTreatment ::= SEQUENCE {
    sFBillingChargingCharacteristics [0] SFBillingChargingCharacteristics,
    informationToSend [1] InformationToSend OPTIONAL,
    maximumNumberOfCounters [2] MaximumNumberOfCounters OPTIONAL,
    releaseCause [3] Cause OPTIONAL
}

```

- If releaseCause is not present, the default value is the same as the ISUP cause value decimal 31.
- If informationToSend is present, the call will be released after the end of the announcement with the indicated or default releaseCause.
- If maximumNumberOfCounters is not present, ServiceFilteringResponse will be sent with CountersValue ::= SEQUENCE SIZE (0) OF CountersAndValue.

```

FilteringCharacteristics ::= CHOICE {
    interval [0] INTEGER (-1..32000),
    numberOfCalls [1] Integer4
}

```

-- Indicates the severity of the filtering and the point in time when the ServiceFilteringResponse is to be sent.
 -- If = interval, every interval of time the next call leads to an InitialDP and a ServiceFilteringResponse is sent to the SCF. The interval is specified in seconds.
 -- If = NumberOfCalls, every N calls the Nth call leads to an InitialDP and a ServiceFilteringResponse is sent to the SCF.
 -- If ActivateServiceFiltering implies several counters – filtering on several dialled numbers –, the numberOfCalls would include calls to all the dialled numbers.

```

FilteringCriteria ::= CHOICE {
    dialledNumber           [0] Digits,
    callingLineID          [1] Digits,
    serviceKey             [2] ServiceKey,
    addressAndService     [30] SEQUENCE {
        calledAddressValue [0] Digits,
        serviceKey         [1] ServiceKey,
        callingAddressValue [2] Digits OPTIONAL,
        locationNumber     [3] LocationNumber OPTIONAL
    }
}
  
```

-- In case calledAddressValue is specified, the numbers to be filtered are from calledAddressValue up to and including calledAddressValue + maximumNumberOfCounters-1.
 -- The last two digits of calledAddressvalue can not exceed 100-maximumNumberOfCounters.

```

FilteringTimeOut ::= CHOICE {
    duration           [0] Duration,
    stopTime          [1] DateAndTime
}
  
```

-- Indicates the maximum duration of the filtering. When the timer expires, a ServiceFilteringResponse is sent to the SCF.

```

ForwardCallIndicators ::= OCTET STRING (SIZE(2))
  
```

-- Indicates the Forward Call Indicators. Refer to Recommendation Q.763 for encoding.

```

ForwardingCondition ::= ENUMERATED {
    busy(0),
    noanswer(1),
    any(2)
}
  
```

-- Indicates the condition that must be met to complete the connect.

```

GapCriteria ::= CHOICE {
    calledAddressValue [0] Digits,
    gapOnService       [2] GapOnService,
    calledAddressAndService [29] SEQUENCE {
        calledAddressValue [0] Digits,
        serviceKey         [1] ServiceKey
    },
    callingAddressAndService [30] SEQUENCE {
        callingAddressValue [0] Digits,
        serviceKey         [1] ServiceKey,
        locationNumber     [2] LocationNumber OPTIONAL
    }
}
  
```

-- Both calledAddressValue and callingAddressValue can be incomplete numbers, in the sense that a limited amount of digits can be given.

-- For the handling of numbers starting with the same digit string, refer to the detailed procedure of the CallGap operation in 3.3.

```

GapOnService ::= SEQUENCE {
    serviceKey [0] ServiceKey,
    dpCriteria [1] EventTypeBCSM OPTIONAL
}
  
```

```

GapIndicators ::= SEQUENCE {
    duration [0] Duration,
    gapInterval [1] Interval
}
-- Indicates the gapping characteristics. No gapping when gapInterval equals 0, and gap all calls when
-- gapInterval equals 1.

GapTreatment ::= CHOICE {
    informationToSend [0] InformationToSend,
    releaseCause [1] Cause,
    both [2] SEQUENCE {
        informationToSend [0] InformationToSend,
        releaseCause [1] Cause
    }
}
-- The default value for Cause is the same as in ISUP.

HighLayerCompatibility ::= OCTET STRING (SIZE (highLayerCompatibilityLength))
-- Indicates the teleservice. For encoding, DSS 1 (Q.931) is used.

HoldCause ::= OCTET STRING -- defined by network operator.
-- Indicates the cause for holding the call.

InbandInfo ::= SEQUENCE {
    messageID [0] MessageID,
    numberOfRepetitions [1] INTEGER (1..127) OPTIONAL,
    duration [2] INTEGER (0..32767) OPTIONAL,
    interval [3] INTEGER (0..32767) OPTIONAL
}
-- Interval is the time in seconds between each repeated announcement. Duration is the total
-- amount of time in seconds, including repetitions and intervals.
-- The end of announcement is either the end of duration or numberOfRepetitions, whatever comes first.
-- Duration with value 0 indicates infinite duration.

InformationToSend ::= CHOICE {
    inbandInfo [0] InbandInfo,
    tone [1] Tone,
    displayInformation [2] DisplayInformation
}

Integer4 ::= INTEGER (0..2147483647)

Interval ::= INTEGER (-1..60000)
-- Units are milliseconds. A -1 value denotes infinite.

InvokeID ::= InvokeIdType
-- Operation invoke identifier.

IPAvailable ::= OCTET STRING (SIZE (minIPAvailableLength..maxIPAvailableLength))
-- defined by network operator.
-- Indicates that the resource is available.

IPRoutingAddress ::= CalledPartyNumber
-- Indicates the routing address for the IP.

IPSSPCapabilities ::= OCTET STRING (SIZE (minIPSSPCapabilitiesLength..maxIPSSPCapabilitiesLength))
-- defined by network operator.
-- Indicates the SRF resources available at the SSP.

ISDNAccessRelatedInformation ::= OCTET STRING
-- Indicates the destination user network interface related information. Refer to the Q.763 Access
-- Transport parameter for encoding.

LegID ::= CHOICE {
    sendingSideID [0] LegType,
    receivingSideID [1] LegType
}

```

-- Indicates a reference to a specific party in a call. *OPTIONAL* denotes network operator specific use
 -- with a choice of unilateral ID assignment or bilateral ID assignment.
 -- *OPTIONAL* for LegID also denotes the following:
 -- -- when only one party exists in the call, this parameter is not needed (as no ambiguity exists);
 -- -- when more than one party exists in the call, one of the following alternatives applies:
 -- 1. LegID is present and indicates which party is concerned.
 -- 2. LegID is not present and a default value is assumed (e.g. calling party in the case of the
 -- ApplyCharging operation).
 -- Choice between these two alternatives is kept a network operator option.

LegType ::= OCTET STRING (SIZE(1))

leg1 LegType ::= '01'H

leg2 LegType ::= '02'H

**LocationNumber ::= OCTET STRING (SIZE (minLocationNumberLength..
 maxLocationNumberLength))**

-- Indicates the Location Number for the calling party. Refer to Recommendation Q.763 (White book) for encoding.

MaximumNumberOfCounters ::= INTEGER (1..numOfCounters)

MessageID ::= CHOICE {
 elementaryMessageID **[0] Integer4,**
 text **[1] SEQUENCE {**
 messageContent **[0] IA5String (SIZE**
 (minMessageContentLength..
 attributes **[1] OCTET STRING (SIZE (**
 minAttributesLength..
 maxAttributesLength))
 OPTIONAL
 },
 elementaryMessageIDs **[29] SEQUENCE SIZE (1..numOfMessageIDs) OF Integer4,**
 variableMessage **[30] SEQUENCE {**
 elementaryMessageID **[0] Integer4,**
 variableParts **[1] SEQUENCE SIZE (1..5)**
 OF VariablePart
 },
}

-- *OPTIONAL* denotes network operator specific use.

MiscCallInfo ::= SEQUENCE {
 messageType **[0] ENUMERATED {**
 request(0),
 notification(1)
 },
 dpAssignment **[1] ENUMERATED {**
 individualLine(0),
 groupBased(1),
 officeBased(2)
 },
OPTIONAL
}

-- Indicates detection point related information.

MonitorMode ::= ENUMERATED {
 interrupted(0),
 notifyAndContinue(1),
 transparent(2)
}

-- Indicates the event is relayed and/or processed by the SSP.
 -- If this parameter is used in the context of charging events, the following definitions apply for the
 -- handling of charging events:
 -- Interrupted means that the SSF notifies the SCF of the charging event using
 -- EventNotificationCharging, does not process the event but discard it.

-- *NotifyAndContinue* means that SSF notifies the SCF of the charging event using
 -- *EventNotificationCharging*, and continues processing the event or signal without waiting for SCF
 -- instructions. *Transparent* means that the SSF does not notify the SCF of the event. This value is used to
 -- end the monitoring of a previously requested charging event. Previously requested charging events are
 -- monitored until ended by a transparent monitor mode, or until the end of the connection configuration.
 -- For the use of this parameter in the context of BCSM events refer to 3.3.39.

NumberingPlan ::= OCTET STRING (SIZE(1))

-- Indicates the numbering plan for collecting the user information. Refer to the Q.763 Numbering Plan.
 -- Indicator field for encoding.

NumberOfDigits ::= INTEGER (1..255)

-- Indicates the number of digits to be collected

**OriginalCalledPartyID ::= OCTET STRING (SIZE (minOriginalCalledPartyIDLength..
 maxOriginalCalledPartyIDLength))**

-- Indicates the original called number. Refer to the Q.763 Original Called Number for encoding.

**RedirectingPartyID ::= OCTET STRING (SIZE (minRedirectingPartyIDLength..
 maxRedirectingPartyIDLength))**

-- Indicates redirecting number. Refer to the Q.763 Redirecting number for encoding.

RedirectionInformation ::= OCTET STRING (SIZE(2))

-- Indicates redirection information. Refer to the Q.763 Redirection Information for encoding.

ReportCondition ::= ENUMERATED {
 statusReport(0),
 timerExpired(1),
 canceled(2)
}

-- *ReportCondition* specifies the cause of sending "StatusReport" operation to the SCF.

RequestedInformationList ::= SEQUENCE SIZE (1..numOfInfoItems) OF RequestedInformation

**RequestedInformationTypeList ::= SEQUENCE SIZE (1..numOfInfoItems) OF
 RequestedInformationType**

RequestedInformation ::= SEQUENCE {
 requestedInformationType [0] RequestedInformationType,
 requestedInformationValue [1] RequestedInformationValue
}

RequestedInformationType ::= ENUMERATED {
 callAttemptElapsedTime(0),
 callStopTime(1),
 callConnectedElapsedTime(2),
 calledAddress(3),
 releaseCause(30)
}

RequestedInformationValue ::= CHOICE {
 callAttemptElapsedTimeValue [0] INTEGER (0..255),
 callStopTimeValue [1] DateAndTime,
 callConnectedElapsedTimeValue [2] Integer4,
 calledAddressValue [3] Digits,
 releaseCauseValue [30] Cause
}

-- *callAttemptElapsedTimeValue* is specified in seconds. The unit for the
 -- *callConnectedElapsedTimeValue* is 100 milliseconds.

ResourceID ::= CHOICE {
 lineID [0] Digits,
 facilityGroupID [1] FacilityGroup,

```

    facilityGroupMemberID      [2]  INTEGER,
    trunkGroupID              [3]  INTEGER
}

```

-- Indicates a logical identifier for the physical termination resource.

```

ResourceStatus ::= ENUMERATED {
    busy(0),
    idle(1)
}

```

```

ResponseCondition ::= ENUMERATED {
    intermediateResponse(0),
    lastResponse(1)
}

```

-- additional values are for further study.
}

-- ResponseCondition is used to identify the reason why ServiceFilteringResponse operation is sent.
-- intermediateresponse identifies that service filtering is running and the interval time is expired and
-- a call is received, or that service filtering is running and the threshold value is reached.
-- lastResponse identifies that the duration time is expired and service filtering has been finished or
-- that the stop time is met and service filtering has been finished.

```

RouteList ::= SEQUENCE SIZE(1..3) OF OCTET STRING (SIZE
    (minRouteListLength..maxRouteListLength))

```

-- Indicates a list of trunk groups or a route index. See Recommendation Q.1214 for additional information on this item.

```

ScfID ::= OCTET STRING (SIZE (minScfIDLength..maxScfIDLength))

```

-- defined by network operator.
-- Indicates the SCF identifier.

```

SCIBillingChargingCharacteristics ::= OCTET STRING (SIZE (minSCIBillingChargingLength..
    maxSCIBillingChargingLength))

```

-- This parameter indicates the billing and/or charging characteristics. Its content is network operator
-- specific. An example datatype definition for this parameter is given below:

```

-- SCIBillingChargingCharacteristics ::= CHOICE {
--     chargeLevel          [0] OCTET STRING (SIZE (min..max)),
--     chargePulses        [1] Integer4,
--     chargeMessages      [2] OCTET STRING (SIZE (min..max))
-- }

```

-- Depending on the applied charging scenario the following information elements

-- can be included (refer to Appendix II/Q.1214):

-- chargeLevel (scenario 3.2)
-- chargePulses (scenario 3.2)
-- chargeMessages (scenario 3.2)

```

ServiceAddressInformation ::= SEQUENCE {
    serviceKey          [0]  ServiceKey          OPTIONAL,
    miscCallInfo       [1]  MiscCallInfo,
    triggerType        [2]  TriggerType          OPTIONAL
}

```

-- Information that represents the result of trigger analysis and allows the SCF to choose the appropriate
-- service logic.

```

ServiceInteractionIndicators ::= OCTET STRING (SIZE ( minServiceInteractionIndicatorsLength..
    maxServiceInteractionIndicatorsLength))

```

-- Indicators which are exchanged between SSP and SCP to resolve interactions between IN based services
-- and network based services, respectively between different IN based services.

-- The contents are network specific and identified as a subject for further study with respect to INAP.

-- The following example is listed to illustrate the use of this parameter:

```

--     CallToBeDiverted          Allowed/NotAllowed Indicator

```

-- If the CallToBeDiverted indicator is set to NotAllowed, the destination exchange shall not allow any diversion on the subjected call. By this, each service can pass the applicable indicators to inform the destination exchange of how specific services are to be handled.

ServiceKey ::= Integer4

-- Information that allows the SCF to choose the appropriate service logic.

ServiceProfileIdentifier ::= OCTET STRING

-- Indicates a particular ISDN terminal. Refer to Recommendation Q.932 for encoding.

ServingAreaID ::= LocationNumber

-- Identifies the local serving area where a network provider operates. Uses the LocationNumber

-- format which is based on the Q.763 Location Number format.

-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).

-- The Numbering Plan Indicator field shall be set to "Spare" (value 000).

-- Defined by the network operator.

SFBillingChargingCharacteristics ::= OCTET STRING (SIZE (minSFBillingChargingLength..maxSFBillingChargingLength))

-- This parameter indicates the billing and/or charging characteristics for filtered calls.

-- Its content is network operator specific.

TerminalType ::= ENUMERATED {

unknown(0),
dialPulse(1),
dtmf(2),
isdn(3),
isdnNoDtmf(4),
spare(16)
}

-- Identifies the terminal type so that the SCF can specify, to the SRF, the appropriate type of capability

-- (voice recognition, DTMF, display capability, etc.). Since present signalling systems do not convey

-- terminal type, this parameter applies only at originating or terminating local exchanges.

TimerID ::= ENUMERATED {

tssf(0)
-- others ffs
}

-- Indicates the timer to be reset.

TimerValue ::= Integer4

-- Indicates the timer value (in seconds).

Tone ::= SEQUENCE {

toneID [0] **Integer4,**
duration [1] **Integer4** **OPTIONAL**
}

-- The duration specifies the length of the tone in seconds, value 0 indicates infinite duration.

TravellingClassMark ::= LocationNumber

-- Indicates travelling class mark information.

-- Uses the LocationNumber format which is based on the Q.763 Location Number format.

-- The Nature of Address indicator field shall be set to "Spare" (value 00000000).

-- The Numbering Plan Indicator field shall be set to "Spare" (value 000).

-- Maximum 2 digits.

TriggerType ::= ENUMERATED {

featureActivation(0),
verticalServiceCode(1),
customizedAccess(2),
}

```

    customizedIntercom(3),
    emergencyService(12),
    aFR(13),
    sharedIOTrunk(14),
    offHookDelay(17),
    channelSetupPRI(18),
    tNoAnswer(25),
    tBusy(26),
    oCalledPartyBusy(27),
    oNoAnswer(29),
    originationAttemptAuthorized(30),
    oAnswer(31),
    oDisconnect(32),
    termAttemptAuthorized(33),
    tAnswer(34),
    tDisconnect(35)
-- Private (ffs)
}

```

-- The type of trigger which caused call suspension
-- 4-11: Reserved; 15,16: Reserved; 19-24: Reserved

```

UnavailableNetworkResource ::= ENUMERATED {
    unavailableResources(0),
    componentFailure(1),
    basicCallProcessingException(2),
    resourceStatusFailure(3),
    endUserFailure(4)
}

```

-- Indicates the network resource that failed.

```

VariablePart ::= CHOICE {
    integer                [0] Integer4,
    number                 [1] Digits,           -- Generic digits
    time                   [2] OCTET STRING (SIZE(2)), -- HH:MM, BCD coded
    date                   [3] OCTET STRING (SIZE(3)), -- YYMMDD, BCD coded
    price                  [4] OCTET STRING (SIZE(4))
}

```

-- Indicates the variable part of the message.
-- BCD coded variable parts are encoded as described in the examples below.
-- For example, time = 12:15 would be encoded as:

```

--      Bits           HGFE       DCBA
--      leading octet      2         1
--                          5         1

```

-- date = 1993 September 30th would be encoded as:

```

--      Bits           HGFE       DCBA
--      leading octet      3         9
--                          9         0
--                          0         3

```

-- The Definition of range of constants Follows

```

highLayerCompatibilityLength      INTEGER ::= 2
minAChBillingChargingLength      INTEGER ::= -- network specific
maxAChBillingChargingLength      INTEGER ::= -- network specific
minAttributesLength              INTEGER ::= -- network specific
maxAttributesLength              INTEGER ::= -- network specific
maxBearerCapabilityLength        INTEGER ::= -- network specific
minCalledPartyNumberLength       INTEGER ::= -- network specific
maxCalledPartyNumberLength       INTEGER ::= -- network specific
minCallingPartyNumberLength      INTEGER ::= -- network specific
maxCallingPartyNumberLength      INTEGER ::= -- network specific
minCallResultLength              INTEGER ::= -- network specific
maxCallResultLength              INTEGER ::= -- network specific

```

```

minCauseLength          INTEGER ::= 2
maxCauseLength          INTEGER ::= -- network specific
minDigitsLength         INTEGER ::= -- network specific
maxDigitsLength         INTEGER ::= -- network specific
minDisplayInformationLength INTEGER ::= -- network specific
maxDisplayInformationLength INTEGER ::= -- network specific
minEventSpecificInformationChargingLength INTEGER ::= -- network specific
maxEventSpecificInformationChargingLength INTEGER ::= -- network specific
minEventTypeChargingLength INTEGER ::= -- network specific
maxEventTypeChargingLength INTEGER ::= -- network specific
minFCIBillingChargingLength INTEGER ::= -- network specific
maxFCIBillingChargingLength INTEGER ::= -- network specific
minIPAvailableLength   INTEGER ::= -- network specific
maxIPAvailableLength   INTEGER ::= -- network specific
minIPSSPCapabilitiesLength INTEGER ::= -- network specific
maxIPSSPCapabilitiesLength INTEGER ::= -- network specific
minLocationNumberLength INTEGER ::= -- network specific
maxLocationNumberLength INTEGER ::= -- network specific
minMessageContentLength INTEGER ::= -- network specific
maxMessageContentLength INTEGER ::= -- network specific
minOriginalCalledPartyIDLength INTEGER ::= -- network specific
maxOriginalCalledPartyIDLength INTEGER ::= -- network specific
minRedirectingPartyIDLength INTEGER ::= -- network specific
maxRedirectingPartyIDLength INTEGER ::= -- network specific
minRouteListLength     INTEGER ::= -- network specific
maxRouteListLength     INTEGER ::= -- network specific
minScfIDLength         INTEGER ::= -- network specific
maxScfIDLength         INTEGER ::= -- network specific
minSCIBillingChargingLength INTEGER ::= -- network specific
maxSCIBillingChargingLength INTEGER ::= -- network specific
minServiceInteractionIndicatorsLength INTEGER ::= -- network specific
maxServiceInteractionIndicatorsLength INTEGER ::= -- network specific
minSFBillingChargingLength INTEGER ::= -- network specific
maxSFBillingChargingLength INTEGER ::= -- network specific
numOfBCSMEvents        INTEGER ::= -- network specific
numOfChargingEvents    INTEGER ::= -- network specific
numOfCounters          INTEGER ::= 100
numOfExtensions        INTEGER ::= -- network specific
numOfInfoItems         INTEGER ::= 5
numOfMessageIDs        INTEGER ::= -- network specific

```

END

2.1.4 IN CS-1 application protocol (operation and error codes)

-- This module contains the operation and error code assignments for the IN CS-1 application protocol.

IN-CS-1-Codes { ccitt recommendation q 1218 modules(0) cs-1-codes(3) version1(0) }

DEFINITIONS ::=

BEGIN

-- OPERATION AND ERROR CODE ASSIGNMENTS FOR THE IN CS-1 PROTOCOL

-- FOLLOW

IMPORTS

-- macros

APPLICATION-SERVICE-ELEMENT

FROM Remote-Operations-Notation-Extension

{joint-iso-ccitt remote-operations(4)notation-extension(2)}

-- operation types

ActivateServiceFiltering,
ActivityTest,
AnalysedInformation,
AnalyseInformation,
ApplyCharging,
ApplyChargingReport,
AssistRequestInstructions,
CallGap,
CallInformationReport,
CallInformationRequest,
Cancel,
CancelStatusReportRequest,
CollectedInformation,
CollectInformation,
Connect,
ConnectToResource,
Continue,
DisconnectForwardConnection,
EstablishTemporaryConnection,
EventNotificationCharging,
EventReportBCSM,
FurnishChargingInformation,
HoldCallInNetwork,
InitialDP,
InitiateCallAttempt,
OAnswer,
OCalledPartyBusy,
ODisconnect,
OMidCall,
ONoAnswer,
OriginationAttemptAuthorized,
PlayAnnouncement,
PromptAndCollectUserInformation,
ReleaseCall,
RequestCurrentStatusReport,
RequestEveryStatusChangeReport,
RequestFirstStatusMatchReport,
RequestNotificationChargingEvent,
RequestReportBCSMEvent,
ResetTimer,
RouteSelectFailure,
SelectFacility,
SelectRoute,
SendChargingInformation,
ServiceFilteringResponse,
SpecializedResourceReport,
StatusReport,
TAnswer,
TBusy,
TDisconnect,
TermAttemptAuthorized,
TMidCall,
TNoAnswer

FROM IN-CS-1-Operations { ccitt recommendation q 1218 modules(0) cs-1-operations(0) version1(0) }

-- error types

Canceled,
CancelFailed,
ETCFailed,
ImproperCallerResponse,
MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
RequestedInfoError,

<i>-- Charging Event handling ASE</i>		
requestNotificationChargingEvent	RequestNotificationChargingEvent	::= localValue 25
eventNotificationCharging	EventNotificationCharging	::= localValue 26
<i>-- SSF call processing ASE</i>		
collectInformation	CollectInformation	::= localValue 27
analyseInformation	AnalyseInformation	::= localValue 28
selectRoute	SelectRoute	::= localValue 29
selectFacility	SelectFacility	::= localValue 30
continue	Continue	::= localValue 31
<i>-- SCF call initiation ASE</i>		
initiateCallAttempt	InitiateCallAttempt	::= localValue 32
<i>-- Timer ASE</i>		
resetTimer	ResetTimer	::= localValue 33
<i>-- Billing ASE</i>		
furnishChargingInformation	FurnishChargingInformation	::= localValue 34
<i>-- Charging ASE</i>		
applyCharging	ApplyCharging	::= localValue 35
applyChargingReport	ApplyChargingReport	::= localValue 36
<i>-- Status reporting ASE</i>		
requestCurrentStatusReport	RequestCurrentStatusReport	::= localValue 37
requestEveryStatusChangeReport	RequestEveryStatusChangeReport	::= localValue 38
requestFirstStatusMatchReport	RequestFirstStatusMatchReport	::= localValue 39
statusReport	StatusReport	::= localValue 40
<i>-- Traffic management ASE</i>		
callGap	CallGap	::= localValue 41
<i>-- Service management ASE</i>		
activateServiceFiltering	ActivateServiceFiltering	::= localValue 42
serviceFilteringResponse	ServiceFilteringResponse	::= localValue 43
<i>-- Call report ASE</i>		
callInformationReport	CallInformationReport	::= localValue 44
callInformationRequest	CallInformationRequest	::= localValue 45
<i>-- Signalling control ASE</i>		
sendChargingInformation	SendChargingInformation	::= localValue 46
<i>-- Specialized resource control ASE</i>		
playAnnouncement	PlayAnnouncement	::= localValue 47
promptAndCollectUserInformation	PromptAndCollectUserInformation	::= localValue 48
specializedResourceReport	SpecializedResourceReport	::= localValue 49
<i>-- Cancel ASE</i>		
cancel	Cancel	::= localValue 53
cancelStatusReportRequest	CancelStatusReportRequest	::= localValue 54
<i>-- Activity Test ASE</i>		
activityTest	ActivityTest	::= localValue 55

-- ERROR codes

cancelled	Cancelled	::= localValue 0
cancelFailed	CancelFailed	::= localValue 1
eTCFailed	ETCFailed	::= localValue 3
improperCallerResponse	ImproperCallerResponse	::= localValue 4
missingCustomerRecord	MissingCustomerRecord	::= localValue 6
missingParameter	MissingParameter	::= localValue 7
parameterOutOfRange	ParameterOutOfRange	::= localValue 8
requestedInfoError	RequestedInfoError	::= localValue 10
systemFailure	SystemFailure	::= localValue 11
taskRefused	TaskRefused	::= localValue 12
unavailableResource	UnavailableResource	::= localValue 13
unexpectedComponentSequence	UnexpectedComponentSequence	::= localValue 14
unexpectedDataValue	UnexpectedDataValue	::= localValue 15
unexpectedParameter	UnexpectedParameter	::= localValue 16
unknownLegID	UnknownLegID	::= localValue 17
unknownResource	UnknownResource	::= localValue 18

-- APPLICATION SERVICE ELEMENTS

SCF-activation-ASE ::= APPLICATION-SERVICE-ELEMENT

-- consumer is SSF

CONSUMER INVOKES {
 initialDP
}

-- supplier is SCF

SUPPLIER INVOKES

SCF-SRF-activation-of-assist-ASE ::= APPLICATION-SERVICE-ELEMENT

-- consumer is SSF/SRF

CONSUMER INVOKES {
 assistRequestInstructions
}

-- supplier is SCF

SUPPLIER INVOKES

Assist-connection-establishment-ASE ::= APPLICATION-SERVICE-ELEMENT

-- consumer is SSF

CONSUMER INVOKES

-- supplier is SCF

SUPPLIER INVOKES {
 establishTemporaryConnection
}

Generic-disconnect-resource-ASE ::= APPLICATION-SERVICE-ELEMENT

-- consumer is SSF

CONSUMER INVOKES

-- supplier is SCF

SUPPLIER INVOKES {
 disconnectForwardConnection
}

Non-assisted-connection-establishment-ASE ::= APPLICATION-SERVICE-ELEMENT

-- consumer is SSF

CONSUMER INVOKES

-- supplier is SCF

SUPPLIER INVOKES {
 connectToResource
}

Connect-ASE ::= APPLICATION-SERVICE-ELEMENT

-- consumer is SSF

CONSUMER INVOKES

-- supplier is SCF

SUPPLIER INVOKES {
 connect
}

```

BCSM-event-handling-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES {
    eventReportBCSM
}
-- supplier is SCF
SUPPLIER INVOKES {
    requestReportBCSMEvent
}

DP-specific-event-handling-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES {
    originationAttemptAuthorized,
    collectedInformation,
    analyzedInformation,
    routeSelectFailure,
    oCalledPartyBusy,
    oNoAnswer,
    oAnswer,
    oDisconnect,
    termAttemptAuthorized,
    TBusy,
    tNoAnswer,
    tAnswer,
    tDisconnect,
    oMidCall,
    tMidCall
}
-- supplier is SCF
SUPPLIER INVOKES {
    requestReportBCSMEven
}

Charging-event-handling-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES {
    eventNotificationCharging
}
-- supplier is SCF
SUPPLIER INVOKES {
    requestNotificationChargingEvent
}

SCF-call-initiation-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES
-- supplier is SCF
SUPPLIER INVOKES {
    initiateCallAttempt
}

Timer-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF/SRF
CONSUMER INVOKES
-- supplier is SCF
SUPPLIER INVOKES {
    resetTimer
}

Billing-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES
-- supplier is SCF
SUPPLIER INVOKES {
    furnishChargingInformation
}

```

```

Charging-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF
  CONSUMER INVOKES {
    applyChargingReport
  }
  -- supplier is SCF
  SUPPLIER INVOKES {
    applyCharging
  }

Traffic-management-ASE ::= APPLICATION-SERVICE-ELEMENT
  --consumer is SSF
  CONSUMER INVOKES
  -- supplier is SCF
  SUPPLIER INVOKES {
    callGap
  }

Service-management-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF
  CONSUMER INVOKES {
    serviceFilteringResponse
  }
  -- supplier is SCF
  SUPPLIER INVOKES {
    activateServiceFiltering
  }

Call-report-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF
  CONSUMER INVOKES {
    callInformationReport
  }
  -- supplier is SCF
  SUPPLIER INVOKES {
    callInformationRequest
  }

Signalling-control-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF
  CONSUMER INVOKES
  -- supplier is SCF
  SUPPLIER INVOKES {
    sendChargingInformation
  }

Specialized-resource-control-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF/SRF
  CONSUMER INVOKES {
    specializedResourceReport
  }
  -- supplier is SCF
  SUPPLIER INVOKES {
    playAnnouncement,
    promptAndCollectUserInformation
  }

Activity-test-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF/SRF
  CONSUMER INVOKES
  -- supplier is SCF
  SUPPLIER INVOKES {
    activityTest
  }

Basic-BCP-DP-ASE ::= APPLICATION-SERVICE-ELEMENT
  -- consumer is SSF
  CONSUMER INVOKES {
    originationAttemptAuthorized,
    collectedInformation,

```

```

analyzedInformation,
routeSelectFailure,
oCalledPartyBusy,
oNoAnswer,
oAnswer,
oDisconnect,
termAttemptAuthorized,
tCalledPartyBusy,
tNoAnswer,
tAnswer,
tDisconnect
}
-- supplier is SCF
SUPPLIER INVOKES

```

```

Advanced-BCP-DP-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES {
    oMidCall,
    tMidCall
}
-- supplier is SCF
SUPPLIER INVOKES

```

```

SSF-Call-Processing-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES
-- supplier is SCF
SUPPLIER INVOKES {
    collectInformation,
    analyzeInformation,
    selectRoute,
    selectFacility,
    continue
}

```

```

Status-reporting-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES {
    statusReport
}
-- supplier is SCF
SUPPLIER INVOKES {
    requestCurrentStatusReport,
    requestEveryStatusChangeReport,
    requestFirstStatusMatchReport
}

```

```

Cancel-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF/SRF
CONSUMER INVOKES
-- supplier is SCF
SUPPLIER INVOKES {
    cancel,
    cancelStatusReportRequest
}

```

```

Call-handling-ASE ::= APPLICATION-SERVICE-ELEMENT
-- consumer is SSF
CONSUMER INVOKES
-- supplier is SCF
SUPPLIER INVOKES {
    holdCallInNetwork,
    releaseCall
}

```

END

2.1.5 Application Contexts

APPLICATION-CONTEXT MACRO::=

BEGIN

TYPE NOTATION ::= Symmetric | InitiatorConsumerOfResponderConsumerOf | empty

VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)

Symmetric ::= "OPERATIONS OF" "{ASEList}"

InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{ASEList}" | empty

ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{ASEList}" | empty

ACEList ::= ASE | ASEList "," ASE

ASE ::= type - shall reference an APPLICATION-SERVICE-ELEMENT type

END

IN-CS1-SSF-to-SCF-Generic-AC APPLICATION-CONTEXT

-- *dialogue initiated by SSF with InitialDP operation*

INITIATOR CONSUMER OF {

SCF-activation-ASE,
Assist-connection-establishment-ASE,
Generic-disconnect-resource-ASE,
Non-assisted-connection-establishment-ASE,
Connect-ASE,
Call-handling-ASE,
BCSM-event-handling-ASE,
Charging-event-handling-ASE,
SSF-call-processing-ASE,
Timer-ASE,
Billing-ASE,
Charging-ASE,
Traffic-management-ASE,
Status-reporting-ASE,
Call-report-ASE,
Signalling-control-ASE,
Specialized-resource-control-ASE,
Cancel-ASE,
Activity-test-ASE
}

RESPONDER CONSUMER OF

::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) generic-ssf-to-scf(0) version 1(0)};

IN-CS1-SSF-to-SCF-DPspecific-AC APPLICATION-CONTEXT

-- *dialogue initiated by SSF with DP Specific Initial Operation*

INITIATOR CONSUMER OF {

Basic-BCP-DP-ASE,
Advanced-BCP-DP-ASE,
Assist-connection-establishment-ASE,
Generic-disconnect-resource-ASE,
Non-assisted-connection-establishment-ASE,
Connect-ASE,
Call-handling-ASE,
DP-specific-event-handling-ASE,
Charging-event-handling-ASE,
SSF-call-processing-ASE,
Timer-ASE,
Billing-ASE,
Charging-ASE,
Traffic-management-ASE,
Status-reporting-ASE,
Call-report-ASE,
Signalling-control-ASE,
Specialized-resource-control-ASE,
Cancel-ASE,
Activity-test-ASE
}

RESPONDER CONSUMER OF

::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) dp-specific-ssf-to-scf(1) version 1(0)};

IN-CS1-assist-handoff-SSF-to-SCF-AC **APPLICATION-CONTEXT**
-- dialogue initiated by SSF with AssistRequestInstructions
INITIATOR CONSUMER OF {
 SCF-SRF-activation-of-assist-ASE,
 Generic-disconnect-resource-ASE,
 Non-assisted-connection-establishment-ASE,
 Call-handling ASE,
 Timer-ASE,
 Billing-ASE,
 Charging-ASE,
 Status-reporting-ASE,
 Specialized-resource-control-ASE,
 Cancel-ASE,
 Activity-test-ASE
}
RESPONDER CONSUMER OF
::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) assist-handoff-ssf-to-scf(2) version 1(0)};

IN-CS1-SRF-to-SCF-AC **APPLICATION-CONTEXT**
-- dialogue initiated by SRF with AssistRequestInstructions
INITIATOR CONSUMER OF {
 SCF-SRF-activation-of-assist-ASE,
 Specialized-resource-control-ASE,
 Cancel-ASE
}
RESPONDER CONSUMER OF
::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) srf-to-scf(3) version 1(0)};

IN-CS1-SCF-to-SSF-AC **APPLICATION-CONTEXT**
-- dialogue initiated by SCF with InitiateCallAttempt, Generic Case
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
 Assist-connection-establishment-ASE,
 Generic-disconnect-resource-ASE,
 Non-assisted-connection-establishment-ASE,
 Connect-ASE,
 Call-handling-ASE,
 BCSM-event-handling-ASE,
 Charging-event-handling-ASE,
 SSF-call-processing-ASE,
 SCF-call-initiation-ASE,
 Timer-ASE,
 Billing-ASE,
 Charging-ASE,
 Status-reporting-ASE,
 Call-report-ASE,
 Signalling-control-ASE,
 Specialized-resource-control-ASE,
 Cancel-ASE,
 Activity-test-ASE
}
::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) generic-scf-to-ssf(4) version 1(0)};

IN-CS1-SCF-to-SSF-AC **APPLICATION-CONTEXT**
-- dialogue initiated by SCF with InitiateCallAttempt, DP-specific Case
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
 Assist-connection-establishment-ASE,
 Generic-disconnect-resource-ASE,
 Non-assisted-connection-establishment-ASE,
 Connect-ASE,
 Call-handling-ASE,
 DP-specific-event-handling-ASE,
 Charging-event-handling-ASE,
 SSF-call-processing-ASE,
 SCF-call-initiation-ASE,
 Timer-ASE,
 Billing-ASE,
 Charging-ASE,
 Status-reporting-ASE,
}

```

    Call-report-ASE,
    Signalling-control-ASE,
    Specialized-resource-control-ASE,
    Cancel-ASE,
    Activity-test-ASE
  }
 ::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) dp-specific-scf-to-ssf(5) version 1(0)};

IN-CS1-SCF-to-SSF-traffic-management-AC      APPLICATION-CONTEXT
-- dialogue initiated by SCF with CallGap
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
    Traffic-management-ASE
  }
 ::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) scf-to-ssf-traffic-management(6) version 1(0)};

IN-CS1-SCF-to-SSF-service-management-AC     APPLICATION-CONTEXT
-- dialogue initiated by SCF with ActivateServiceFiltering
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
    Service-management-ASE
  }
 ::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) scf-to-ssf-service-management(7) version 1(0)};

IN-CS1-SSF-to-SCF-service-management-AC     APPLICATION-CONTEXT
-- dialogue initiated by SSF with ServiceFilteringResponse
INITIATOR CONSUMER OF {
    Service-management-ASE
  }
RESPONDER CONSUMER OF
 ::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) scf-to-ssf-service-management(8) version 1(0)};

IN-CS1-SCF-to-SSF-status-reporting-AC       APPLICATION-CONTEXT
-- dialogue initiated by SCF with Status Reporting Operations
INITIATOR CONSUMER OF
RESPONDER CONSUMER OF {
    Cancel-ASE,
    Status-reporting-ASE
  }
 ::= {ccitt recommendation q1218 scf-ssf-srf-objects(1) scf-to-ssf-status-reporting(9) version 1(0)};

```

2.2 Interface SCF-SDF

2.2.1 Introduction au sous-ensemble DAP dans la Recommandation X.500 du réseau intelligent

2.2.1.1 Adaptation des concepts contenus dans la Recommandation X.500 en fonction du réseau intelligent

Les Recommandations X.500 sont utilisées pour définir l'interface SCF-SDF et le contenu de l'entité SDF. La plupart des concepts de la série X.500 sont directement utilisés dans l'environnement du réseau intelligent, mais des adaptations doivent cependant être apportées au niveau de la terminologie afin de s'assurer que les concepts qui ont été définis dans l'annuaire sont bien compris. Ce paragraphe a pour objectif d'apporter ces adaptations. C'est pourquoi il se limite aux termes qui présentent une ambiguïté dans l'environnement des réseaux intelligents.

Quand on examine la structure de l'entité SCF, la gestion des données de fonctionnement fait partie de l'entité SCF qui est responsable des interactions avec l'entité SDF. Elle peut être élaborée en vue de définir le concept d'agent utilisateur de l'annuaire (DUA). Quand une entité SCF à la demande d'un utilisateur veut réaliser une association avec une SDF, un exemple de DUA est créé dans le SLPI. Il disparaît quand une association cesse d'exister.

L'entité SDF est chargée de répondre aux demandes de la base de données. Cette unité fonctionnelle peut être organisée en différents agents du système de l'annuaire (DSA). Quand une association est créée entre une entité SCF et une entité SDF, une instance d'agent DSA est créée pour la durée de l'association.

L'annuaire est un ensemble d'agents DSA et d'entités SDF. Cet ensemble peut être utilisé pour un service spécifique ou pour des services divers. La notion d'annuaire est l'équivalent du concept de bases de données dans le réseau intelligent (RI).

L'annuaire peut également être considéré comme un répertoire de données. Les services du réseau intelligent fournissent aux utilisateurs différents types d'accès de données. Une entrée est un ensemble d'informations pouvant être identifié (ou désigné par un nom). Quand elle représente un objet (contenant par exemple des informations essentielles relatives à un objet), cette entrée s'appelle "entrée objet".

On appelle objet tout ce qui peut être désigné par un nom, pour lequel il est intéressant de garder des informations sur la base de données. Un exemple caractéristique d'objet est un utilisateur. Les objets peuvent être décrits par plusieurs entrées. Chaque élément d'information isolé utilisé pour décrire un objet est un attribut. Les informations sont associées en vue de constituer des entrées.

Dans un environnement de réseau intelligent, celui qui fournit un service est responsable de la gestion et de l'administration des données contenues dans une DSA. De ce fait, celui qui fournit le service joue un rôle d'administrateur. Il joue un rôle d'administrateur dans la terminologie relative à la Recommandation X.500. Le fournisseur de service met en application la procédure de sécurité (authentification et contrôle d'accès).

2.2.1.2 Utilisation d'un sous-ensemble limité de Recommandations X.500

L'objectif initial des Recommandations X.500 était de fournir un service à l'annuaire et non une description de l'interface SCF-SDF telle que la Commission d'études 11 voulait le faire. Les fonctionnalités de la Recommandation X.500 vont au-delà des fonctionnalités nécessaires à l'ensemble de capacités 1 du réseau intelligent. Le présent paragraphe s'efforce d'indiquer quels aspects du service abstrait d'annuaire devraient être envisagés et mis en place par les fabricants dans le domaine d'application de l'ensemble de capacités 1 du réseau intelligent. Il indique également quelle attitude doit être adoptée quand on rencontre un paramètre imprévu. Une notion de profil est utilisée comme moyen de présenter les valeurs des différents paramètres.

Il est important de mentionner que le nombre de paramètres d'un message devrait être réduit au minimum, parce que chacun d'eux augmente la charge dans le trafic de signalisation et le temps de traitement. C'est la raison pour laquelle les paramètres ne sont pas transmis, sauf si ils sont absolument indispensables. A la réception, les paramètres omis ne devraient pas être analysés, mais les paramètres convenus seront repris par l'entité réceptrice. Cela facilitera l'évolution future conformément à la description qui est faite dans l'édition 1993 de l'annuaire.

Pour plus de commodité et de clarté, ce profil est défini en utilisant les possibilités de sous-typage de la syntaxe ASN.1, ces définitions ne constituant cependant pas une spécification de protocole. Cela indique simplement quels paramètres n'ont pas à être envoyés, mais cela ne modifie pas le comportement de l'entité réceptrice qui sera toujours en état de décoder les valeurs conformes à la définition originale du service abstrait d'annuaire. Néanmoins, les éléments qui sont exclus par le sous-typage ne devraient pas être pris en compte.

2.2.1.3 Hypothèses de travail

Essai de révision

Plusieurs hypothèses de travail ont été utilisées pour concevoir le profil du service abstrait d'annuaire relatif à l'ensemble de capacités 1 du réseau intelligent. Les hypothèses suivantes sont faites au 2.2.2.1:

- **Hypothèse 1:** la version du service abstrait d'annuaire utilisée dans l'ensemble de capacités 1 est la version de 1993. Les paramètres utilisés uniquement dans la version de 1988 ne seront pas pris en compte. Les fonctionnalités qui pourraient être nécessaires dans les futurs ensembles de capacités devraient être au moins examinées si elles ne sont pas retenues.
- **Hypothèse 2:** les mécanismes de répartition de données ne sont pas pris en compte à l'intérieur de l'ensemble de capacités 1. Cela repose sur l'hypothèse selon laquelle les mécanismes de répartition utilisent une interface SDF-SDF ne faisant pas partie de l'ensemble de capacités 1. Ces mécanismes comprennent le chaînage et la diffusion de demandes, de même que le renvoi. Bien qu'elles n'aient pas besoin d'être retenues, ces indications relatives à l'emploi de tels mécanismes devraient être prises en compte (voir hypothèse 1).
- **Hypothèse 3:** les mécanismes de copie d'annuaire ne sont pas pris en compte dans l'ensemble de capacités 1. Comme les mécanismes de répartition de données, ils utilisent l'interface SDF-SDF. Ils impliquent des mécanismes complexes de maintenance qui n'entrent pas dans le domaine d'application de l'ensemble de capacités 1.
- **Hypothèse 4:** les opérations utilisées pour l'ensemble de capacités 1 ne nécessitent pas de signature. Etant donné qu'aucune exigence de sécurité n'a été exprimée quant à l'utilisation d'opérations signées comme la définition d'un algorithme pour la signature, cette fonctionnalité n'est pas couverte par le domaine d'application de l'ensemble de capacités 1.

- **Hypothèse 5:** les mécanismes d'authentification forte ne sont pas couverts par le présent profil. Ces mécanismes sont utilisés pour les systèmes de sécurité à clés publiques. Etant donné l'état actuel des travaux sur la sécurité, une telle fonctionnalité n'est pas prévue pour l'ensemble de capacités 1.
- **Hypothèse 6:** les entrées alias dans le réseau intelligent sont juste un moyen de donner un autre nom à un objet et elles doivent être déréférencées si nécessaire.
- **Hypothèse 7:** une opération SCF-SDF ne peut pas être abandonnée. Si une opération prend trop de temps, son temporisateur vient à expiration et il n'est pas nécessaire de l'abandonner.

2.2.2 Le sous-ensemble protocole d'accès à l'annuaire X.500 du réseau intelligent

2.2.2.1 Examen de X.511 devant être utilisé dans le réseau intelligent

Les profils de la Recommandation X.511 fournis dans ce qui suit sont fondés sur la Recommandation X.511 et sur ses amendements (projet d'amendement 2 sur les contextes et projet d'amendement 3 sur les extensions mineures).

2.2.2.1.1 Types d'information et procédures communes

2.2.2.1.1.1 Arguments communs

```
IN-CommonArguments ::= CommonArguments (
    WITH COMPONENTS {
        serviceControls          (IN-ServiceControls....),
        securityParameters      OPTIONAL,
        requestor                ABSENT,
        operationProgress        ({nameResolutionPhase notStarted}....),
        aliasedRDNs              ABSENT,
        criticalExtensions       OPTIONAL
        referenceType            ABSENT,
        entryOnly(TRUE....),
        exclusions                ABSENT,
        nameResolveOnMaster      (FALSE....)
        operationContexts       OPTIONAL,
        ...)

```

L'état de chaque composante du paramètre **CommonArguments** (argument commun) est le suivant:

- la composante **serviceControls** (contrôles de service) (voir 2.2.2.1.1.3);
- la composante **securityParameters** (paramètres de sécurité) – La composante **securityParameters** régit les caractéristiques de sécurité associées à une opération sur l'annuaire. Etant donné que les opérations n'ont pas de signature dans l'ensemble de capacités 1 (hypothèse 4), la composante **securityParameters** ne devrait pas exister;
- la composante **requestor** (demandeur) fournit le nom de l'auteur d'une opération particulière. Cette composante est utilisée principalement pour les demandes signées, et en tant que telle ne devrait pas être envoyée (hypothèse 4);
- les composantes **operationProgress**, **referenceType**, **entryOnly**, **exclusions** et **nameResolveOnMaster** sont utilisées quand l'agent DUA opère sur une instruction de poursuite émanant d'un agent DSA. Etant donné que les mécanismes de répartition ne sont pas pris en charge (hypothèse 2), ces paramètres ne devraient pas être envoyés;
- les composantes **aliased RDNs** (noms RDN alias) existent dans la version de 1993, uniquement pour des raisons de compatibilité. Elles devraient toujours être omises dans les réalisations de l'annuaire de 1993 (hypothèse 1);
- la composante **criticalExtensions** (extensions critiques) doit être utilisée pour prévenir les réalisations conformes aux éditions de 1988 du caractère critique d'un nombre d'extensions qui sont disponibles dans les versions de 1993. Etant donné que la version de 1993 est utilisée dans l'ensemble de capacités 1 (hypothèse 1), la composante ne devrait pas être envoyée. Cependant, les extensions suivantes devraient être prises en considération dans le réseau intelligent: sous-entrées, modifyRightsRequest, matchedValuesOnly, useAliasOnUpdate, useContexts, selectionOnModify, extendedEntryMods;
- la composante **operationContexts** (contextes d'opération) fournit un ensemble d'assertions de contexte qui sont appliquées aux assertions de valeur d'attribut et à l'information d'entrée au cours de l'opération. Puisque l'utilisation de contextes fait partie des prescriptions du réseau intelligent, la composante est présente chaque fois que cela est nécessaire.

2.2.2.1.1.2 Résultats communs

```
IN-CommonResults ::= CommonResults (  
    WITH COMPONENTS {  
        securityParameters    ABSENT,  
        performer             ABSENT,  
        aliasDereferenced})
```

L'état de chaque composante du paramètre **CommonResults** (résultats communs) est le suivant:

- la composante **securityParameters** (paramètres de sécurité) – Les opérations ne sont pas signées, c'est pourquoi cette composante ne devrait pas être envoyée;
- la composante **performer** (exécutant) fournit le nom de l'exécutant d'une opération donnée. Elle s'emploie principalement dans le cas de résultats signés et ne devrait pas être envoyée en tant que telle (hypothèse 4);
- la composante **aliasDereferenced** (alias déréférencé) est mis à l'état vrai quand un alias a été rencontré et déréférencé pendant le traitement d'une opération.

2.2.2.1.1.3 Contrôles de service

```
IN-ServiceControls ::= ServiceControls  
(WITH COMPONENTS {  
    options                ({chainingProhibited, localScope}  
                           ({changingProhibited, localScope}  
                           {changingProhibited, localScope, dontDereferenceAliases, subentries}  
                           {chainingProhibited, localScope, subentries}),  
    priority               ,  
    timeLimit              ABSENT,  
    sizeLimit              ABSENT,  
    scopeOfReferral        ABSENT,  
    attributeSizeLimit     ABSENT})
```

L'état de chaque composante du paramètre **ServiceControls** (contrôles de service) est le suivant:

- la composante **options** (options) contient un certain nombre d'indications:
 - **preferChaining** devrait être absent parce qu'il fait appel à la répartition (hypothèse 2);
 - **chaining Prohibited** et **localScope** devraient toujours être fixés, parce que le chaînage n'est pas autorisé dans l'ensemble de capacités 1 (hypothèse 2);
 - **dontUseCopy** et **copyShallDo** font référence à des mécanismes de copie et ne devraient pas être présents (hypothèse 3);
 - **dontDereferenceAliases** devrait uniquement être inclus quand l'objet en question est un alias (par exemple déplacer un alias);
 - **les sous-entrées** devraient être fixées en fonction des entrées ou des sous-entrées à traiter;
- la composante **priority** (priorité) est utilisée pour spécifier quelle est la priorité du service qui doit être apportée. Ce paramètre pourrait être utilisé pour gérer les encombrements dans l'agent DSA et peut s'avérer nécessaire;
- la composante **timeLimit** (temps maximal) indique le temps maximal écoulé pour satisfaire une demande. Elle fait double emploi avec les temporisateurs de TCAP et n'est donc pas nécessaire;
- la composante **sizeLimit** (limites de taille en termes d'objets) et la composante **attributeSizeLimit** (limites de taille en termes d'attribut) fixent certaines limites de taille aux résultats, soit en termes d'objets, soit en termes d'attributs. Ceci est utile quand on s'attend à ce que les demandes soient générales (le demandeur ne connaît pas la structure de la DSA), mais dans le cas du réseau intelligent, ce type de restriction ne semble pas applicable;
- la composante **scopeOfReferral** (champ d'une affectation) indique le champ s'appliquant à une affectation. Etant donné que l'affectation n'est pas une des caractéristiques examinées dans l'ensemble de capacités 1 (hypothèse 2), ce paramètre ne devrait pas être présent.

2.2.2.1.1.4 Sélection de l'information d'entrée

```
IN-EntryInformationSelection ::= EntryInformationSelection  
(WITH COMPONENTS {  
    attributes              ,  
    infoTypes              (attributeTypesAndValues...),  
    extraAttributes        OPTIONAL
```

```

contextSelection      OPTIONAL,
returnContexts      },
...)
```

L'état de chaque composante du paramètre **EntryInformationSelection** (sélection de l'information d'entrée) est le suivant:

- la composante **attributes** (attributs) spécifie les attributs qui doivent être renvoyés dans un service de consultation. L'option **allUserAttributes** (attributs pour tous les utilisateurs) est conservée même si il est conseillé aux spécificateurs de service d'éviter son utilisation qui génère plus de trafic que nécessaire. L'option **select** (sélection), qui nomme avec précision les attributs demandés, doit être préférée;
- la composante **infoTypes** (types d'information) spécifie si les types et les valeurs d'attributs ou si seulement les types doivent être renvoyés. Les services du réseau intelligent sont principalement intéressés par les valeurs de l'attribut qui sont en rapport avec le traitement du service. Etant donné sa valeur par défaut, il est inutile d'indiquer cette composante;
- la composante **extraAttributes** a une utilisation similaire à la composante **attributes**. La seule différence est l'option **allUserAttributes** qui est remplacée par l'option **allOperationalAttributes**.
- la composante **contextSelection** (sélection de contexte) sert à spécifier les valeurs d'attribut à renvoyer pour les attributs sélectionnés par la composante **attributes** lorsque les contextes sont utilisés. Cette composante est donc importante pour les spécifications de réseau intelligent;
- la composante **returnContexts** (contextes de renvoi) sert à demander à l'entité SDF de renvoyer les valeurs d'attribut avec les listes de contextes associés. La composante peut être utilisée pour certains services de réseau intelligent.

2.2.2.1.1.5 Information d'entrée

IN-EntryInformation ::= EntryInformation

```

(WITH COMPONENTS {
  name                PRESENT,
  fromEntry           (TRUE...),
  information         (WITH COMPONENTS {
                        attributeType  ABSENT,
                        attribute       PRESENT}....) OPTIONAL,
  incompleteEntry    ,
  partialNameResolution (FALSE...),
...)
```

L'état de chaque composante du paramètre **EntryInformation** (information d'entrée) est le suivant:

- la composante **name** (nom) donne le nom de l'entrée renvoyée. C'est une composante obligatoire;
- la composante **fromEntry** (de l'entrée) indique si l'entrée elle-même ou une copie de celle-ci est renvoyée. Puisque l'ensemble de capacités 1 du réseau intelligent n'utilise pas les mécanismes de copie (hypothèse 3), seule la valeur par défaut de cette composante doit être utilisée;
- le paramètre **information** contient l'information utile qui est renvoyée. Du fait du choix effectué pour la composante **infoTypes** (voir 2.2.2.1.1.4), seule l'option **attribute** doit être utilisée;
- le paramètre **incompleteEntry** (entrée incomplète) indique que le résultat renvoyé n'est pas complet à cause, par exemple, de certaines limitations de droit d'accès. Il est nécessaire de prendre en charge un tel paramètre;
- la composante **partialNameResolution** (résolution de nom partielle) indique si le nom a été partiellement ou complètement résolu. Cette composante ne correspond à aucune prescription du réseau intelligent et n'est pas incluse.

2.2.2.1.1.6 Paramètres optionnels signés

UNSIGNED {Type} ::= OPTIONALLY-SIGNED {Type}

```

(WITH COMPONENTS {
  unsigned           PRESENT,
  signed             ABSENT})
```

Les opérations pour l'ensemble de capacités 1 ne sont pas signées (hypothèse 4). Cela signifie que seules les options signées doivent être utilisées.

2.2.2.1.2 Opération de rattachement

```
in-DirectoryBind OPERATION ::= {  
    ARGUMENT      IN-DirectoryBindArgument  
    RESULT        IN-DirectoryBindResult  
    ERRORS        {in-DirectoryBindError}}
```

Les opérations **DirectoryBind** (rattachement à l'annuaire) et **DirectoryUnbind** (détachement de l'annuaire) sont utilisées au début et à la fin de chaque période d'accès à l'annuaire.

2.2.2.1.2.1 Arguments et résultats de l'opération de rattachement

```
IN-DirectoryBindArgument ::= DirectoryBindArgument  
    (WITH COMPONENTS {  
        credentials      (IN-Credentials) OPTIONAL,  
        versions         })
```

```
IN-Credentials ::= Credentials  
    (WITH COMPONENTS {  
        simple           ,  
        strong           ABSENT,  
        externalProcedure })
```

```
IN-DirectoryBindResult ::= IN-DirectoryBindArgument
```

L'état de chaque paramètre **DirectoryBindArgument** (argument de rattachement à l'annuaire) est le suivant:

- le paramètre **credentials** (pouvoir) est utilisé pour établir l'identité de l'utilisateur. Il est nécessaire pour les services qui requièrent une authentification. Cependant le pouvoir **strong** (pouvoir étendu) ne doit pas être utilisé parce qu'il correspond à l'utilisation d'un système à clé publique (hypothèse 5);
- le paramètre **versions** identifie la version du service d'annuaire utilisée.

Les mêmes considérations s'appliquent au paramètre **DirectoryBindResult** (résultat de rattachement à l'annuaire).

2.2.2.1.2.2 Erreurs de l'opération de rattachement

```
in-DirectoryBindError ERROR ::= {  
    PARAMETER IN-DirectoryBindErrorParameter}
```

```
IN-DirectoryBindErrorParameter ::= DirectoryBindErrorParameter  
    (WITH COMPONENTS {  
        versions         ,  
        error           (WITH COMPONENTS {  
            securityError      (SecurityProblem (1/2/7....)),  
            serviceError       (ServiceProblem (2))}))
```

Le message d'erreur **DirectoryBindError** (erreur de rattachement à l'annuaire) reste défini pour le service abstrait d'annuaire. Tous les messages d'erreur possibles doivent être pris en charge dès leur réception afin d'interpréter la cause d'une erreur de recherche.

2.2.2.1.3 Opération de recherche

```
in-Search OPERATION ::= {  
    ARGUMENT      IN-SearchArgument  
    RESULT        IN-SearchResult  
    ERRORS        {in-NameError | in-ServiceError | in-SecurityError | attributeError}  
    CODE          id-opcode-in-search}
```

L'opération de recherche sert à chercher les entrées répondant à un critère donné dans une portion de l'arbre d'informations de l'annuaire (DIT).

2.2.2.1.3.1 Arguments et erreurs de l'opération de recherche

```
IN-SearchArgument ::= UNSIGNED {SearchArgument  
    (WITH COMPONENTS {  
        baseObject      PRESENT,  
        subset          ,  
        filter          ,  
        searchAliases   (TRUE....),  
        selection        (IN-EntryInformationSelection....),  
        pagedResults     ABSENT,
```

```

matchedValuesOnly      ,
extendedFilter        ABSENT,
checkOverspecified   },
...)}

```

L'état de chaque paramètre de la composante **SearchArgument** (argument de recherche) est le suivant:

- le paramètre **baseObject** (objet de base) identifie l'entrée d'objet pour laquelle l'opération de recherche doit prendre place. Ce paramètre est obligatoire;
- le paramètre **subset** (sous-ensemble) indique quelle partie de l'arbre d'informations de l'annuaire doit être utilisée pour l'opération de recherche. Ce paramètre est nécessaire même si, pour éviter que la temporisation de l'opération expire: l'utilisation du sous-arbre **wholeSubtree** (sous-arbre complet) doit être elle-même évitée;
- le paramètre **filter** (filtre) est utilisé pour éliminer des entrées de l'espace de recherche. Cependant le paramètre **extendedFilter** (filtre étendu) a été ajouté dans la version 1993 de l'annuaire pour des raisons de compatibilité et ne doit par conséquent pas être envoyé. Seul le paramètre **filter** doit être envoyé;
- le paramètre **searchAliases** (alias de recherche) indique si les alias rencontrés dans l'espace de recherche (à l'exception de l'objet de base) doivent être pris en compte. Puisque les alias du réseau intelligent sont toujours déréférencés lors de la recherche, ce paramètre doit être utilisé uniquement avec sa valeur par défaut;
- le paramètre **sélection** indique l'information des entrées qui sont demandées (voir 2.2.2.1.1.4), types et valeurs par exemple;
- le paramètre **pagedResults** (résultats paginés) est utilisé pour demander un résultat page par page. Le paramètre **pagedResults** est utilisé pour présenter les résultats d'une opération de recherche sous un format page. Ce type d'information n'est pas nécessaire dans un ensemble de capacités 1 puisque l'entité SCF traite les résultats;
- le paramètre **matchedValuesOnly** est un service complémentaire utilisé pour renvoyer uniquement les valeurs qui remplissent la condition définie par le paramètre **filter**. Cette fonctionnalité peut être nécessaire pour un ensemble de capacités 1 du réseau intelligent;
- le paramètre **checkOverspecified** sert à demander à l'entité SDF de renvoyer s'il y a lieu un item **overspecFilter** dans le résultat de recherche. Les services du réseau intelligent peuvent avoir besoin d'un item **overspecFilter** et ce paramètre doit donc être inclus.

Les messages d'erreurs **abandoned** (erreur d'abandon) et **referral** (erreur de renvoi) ne sont pas pris en charge.

2.2.2.1.3.2 Résultats de l'opération de recherche

```

IN-SearchResult ::= UNSIGNED {SearchResult
  (WITH COMPONENTS {
    searchInfo      (WITH COMPONENTS {
      name          ,
      entries      (WITH COMPONENT (IN-EntryInformation...)...),
      partialOutcomeQualifier (PartialOutcomeQualifier
        (WITH COMPONENTS {
          limitProblem          OPTIONAL,
          unexplored            ABSENT,
          unavailableCriticalExtensions ,
          unknownErrors        ABSENT,
          queryReference       ABSENT,
          overspecFilter       OPTIONAL},
          ... )...}),
        ...),
      uncorrelatedSearchInfo ABSENT},
    ...})

```

L'état de chaque composante du paramètre **SearchResult** (résultat de recherche) est le suivant:

- le paramètre **name** indique le nom des entrées renvoyées s'il diffère du nom du paramètre **baseObject**. Il est nécessaire lorsque l'opération de recherche produit un résultat comportant des entrées différentes de l'objet de base;
- le paramètre **entries** (entrées) contient les entrées qui satisfont au filtre (voir 2.2.2.1.1.5);

- le paramètre **partialOutcomeQualifier** (qualificatif des résultats partiels) est présent lorsque l'opération de recherche n'a pas été menée à terme. Il indique les raisons pour lesquelles l'opération de recherche ne s'est pas terminée et l'endroit où elle a été interrompue:
 - le paramètre **limitProblem** (problème de limite) indique quel type de limite a été atteinte (administrative). Il est nécessaire pour acheminer de telles indications;
 - le paramètre **unexplored** (inexploré) indique où l'opération de recherche a été arrêtée et donne des indications concernant l'état de l'opération (objet considéré et indication de déroulement). Si l'opération doit être réexécutée, un tel paramètre est nécessaire. Il est équivalent à une référence et comme telle doit être absent (hypothèse 2);
 - le paramètre **unavailableCriticalExtensions** (extensions critiques non disponibles) indique que certaines extensions critiques n'ont pas été disponibles;
 - le paramètre **unknownErrors** (erreurs non répertoriées) correspond à la réception d'erreurs non répertoriées provenant d'autres entités SDF. Il implique l'utilisation d'un mécanisme de chaînage qui n'est pas disponible dans l'ensemble de capacités 1 (hypothèse 2), et n'est donc pas utilisé;
 - le paramètre **queryReference** (consultation de référence) est utilisé lorsque des résultats paginés ont été demandés et n'est donc pas nécessaire;
 - le paramètre **overspecFilter** est utilisé lorsque le résultat de recherche renvoyé est vide du fait d'un filtrage trop restrictif. Le filtre renvoyé correspond à une partie du filtre initial que vérifient encore un certain nombre d'entrées. Cette fonctionnalité pourrait être utilisée pour les services du réseau intelligent;
- le paramètre **uncorrelatedSearchInfo** (information de recherche non corrélée) contient les résultats obligatoirement signés et qui ne l'étaient pas. Puisque les opérations ne sont pas signées dans l'ensemble de capacités 1 du réseau intelligent, ce paramètre ne doit pas être utilisé.

2.2.2.1.4 Opération d'ajout d'entrée

```

in-AddEntry OPERATION ::= {
  ARGUMENT  IN-AddEntryArgument
  RESULT    AddEntryResult
  ERRORS    {nameError | in-ServiceError | in-SecurityError | attributeError |
             in-UpdateError}
  CODE      id-opcode-in-addEntry}

```

L'opération **addEntry** (ajout d'entrée) sert à ajouter une entrée feuille à l'arbre d'informations de l'annuaire (DIT).

2.2.2.1.4.1 Arguments, résultats et erreurs de l'opération d'ajout d'entrée

```

IN-AddEntryArgument ::= UNSIGNED {AddEntryArgument
  (WITH COMPONENTS {
    object          PRESENT,
    entry           PRESENT,
    targetSystem    ABSENT})}

```

L'état de chaque composante du paramètre **AddEntryArgument** (argument d'ajout d'entrée) est le suivant:

- le paramètre **object** (objet) identifie l'entrée qui doit être ajoutée. Ce paramètre est obligatoire;
- le paramètre **entry** décrit l'entrée qui doit être ajoutée. Ce paramètre est aussi obligatoire;
- le paramètre **targetSystem** (système cible) contient le nom de l'entité SDF où l'entrée a été ajoutée. Lorsque aucune distribution n'est considérée (hypothèse 2), l'entité SDF est la même que celle d'où l'opération a été envoyée. Le paramètre n'est donc pas nécessaire.

Le message d'erreur **referral** (erreur de référence) n'est pas pris en charge.

2.2.2.1.5 Opération de suppression entrée

```

in-RemoveEntry OPERATION ::= {
  ARGUMENT  IN-RemoveEntryArgument
  RESULT    RemoveEntryResult
  ERRORS    {nameError | in-ServiceError | in-SecurityError | in-UpdateError}
  CODE      id-opcode-in-removeEntry}

```

L'opération **removeEntry** (suppression d'entrée) sert à supprimer une entrée feuille de l'arbre d'informations de l'annuaire (DIT).

2.2.2.1.5.1 Arguments, résultats et erreurs de l'opération de modification d'entrée

IN-RemoveEntryArgument ::= UNSIGNED {RemoveEntryArgument}

Les paramètres se limitent au nom de l'objet à supprimer. Ce paramètre est obligatoire. On doit garder l'argument de l'opération comme il est décrit dans l'annuaire.

Le message d'erreur **referral** (erreur de référence) n'est pas pris en charge.

2.2.2.1.6 Opération de modification d'entrée

in-ModifyEntry OPERATION ::= {
 ARGUMENT **IN-ModifyEntryArgument**
 RESULT **IN-ModifyEntryResult**
 ERRORS {**nameError** | **in-ServiceError** | **in-SecurityError** | **attributeError** |
 in-UpdateError}
 CODE **id-opcode-in-modifyEntry**
}

L'opération **modifyEntry** (modification d'entrée) est utilisée pour appliquer une série de modifications à une entrée.

2.2.2.1.6.1 Arguments et erreurs de l'opération de modification d'entrée

IN-ModifyEntryArgument ::= UNSIGNED {ModifyEntryArgument
 (WITH COMPONENTS {
 object **PRESENT,**
 changes **PRESENT,**
 selection **(IN-EntryInformationSelection))}}**

Les paramètres sont le nom de l'objet à modifier, la liste des modifications et le paramètre **sélection** qui spécifie certains attributs et valeurs à renvoyer. Les deux premiers sont obligatoires et le troisième correspond à celui du 2.2.2.1.1.4. Puisque toutes les modifications disponibles dans le paramètre **changes** (modifications) sont utiles pour l'ensemble de capacités 1 du réseau intelligent, l'argument de l'opération doit rester tel qu'il est décrit dans l'annuaire à l'exception du paramètre **sélection**.

Le message d'erreur **referral** n'est pas pris en charge.

2.2.2.1.6.2 Résultats de l'opération de modification d'entrée

IN-ModifyEntryResult ::= ModifyEntryResult
 (WITH COMPONENTS {
 null **,**
 information **UNSIGNED {Information**
 (WITH COMPONENTS {
 entry **(IN-EntryInformation))}}**
)}

Si l'opération **modifyEntry** (modification d'entrée) n'a aucune information à extraire, elle renvoie le résultat **null** (néant). Sinon l'information doit être renvoyée dans la composante **entry** du résultat **information**. Pour l'ensemble de capacités 1 du réseau intelligent, cette composante est spécifiée dans 2.2.2.1.1.5.

2.2.2.1.7 Erreurs

La règle de préséance définie dans l'annuaire s'applique.

Les messages d'erreurs **abandoned**, **abandonFailed** et **referral** ne sont pas pris en considération, car ils ne sont pas pris en charge par l'ensemble de capacités 1 (hypothèse 1 et hypothèse 7).

2.2.2.1.7.1 Erreur d'attribut

Tous les problèmes d'attributs cités dans le message d'erreur **attributeError** (erreur d'attribut) peuvent être utilisés dans l'ensemble de capacités 1, et ce message d'erreur ne doit donc pas être modifié et doit être pris directement de l'annuaire.

2.2.2.1.7.2 Erreur de nom

Tous les problèmes de dénomination cités dans le message d'erreur **nameError** (erreur de nom) peuvent être utilisés dans l'ensemble de capacités 1, et ce message d'erreur ne doit donc pas être modifié et doit être pris directement de l'annuaire.

2.2.2.1.7.3 Erreur de sécurité

```
in-SecurityError ERROR ::= {  
    PARAMETER    IN-SecurityErrorParameter  
    CODE         id-errcode-in-securityError}
```

```
IN-SecurityErrorParameter ::= SecurityErrorParameter  
    (WITH COMPONENTS {  
        problem    (SecurityProblem (1 | 2 | 3 | 6))})
```

Le message d'erreur **securityError** (erreur de sécurité) signale un problème de sécurité qui a été rencontré durant l'exécution d'une opération.

Il contient uniquement le paramètre **problem** qui indique quel type de problème a été rencontré. Parmi les problèmes possibles, les problèmes **invalidSignature** (signature non valide) et **protectionRequired** (protection requise) ne doivent pas être envoyés, car ils sont utilisés en conjonction avec des opérations signées (hypothèse 4).

2.2.2.1.7.4 Erreur de service

```
in-ServiceError ERROR ::= {  
    PARAMETER    IN-ServiceErrorParameter  
    CODE         id-errcode-in-serviceError}
```

```
IN-ServiceErrorParameter ::= ServiceErrorParameter  
    (WITH COMPONENTS {  
        problem    (ServiceProblem (1 | 2 | 3 | 5 | 10 | 8 | 9 | 12))})
```

Le message d'erreur **serviceError** (erreur de service) signale un problème relatif à la fourniture du service.

Il contient uniquement le paramètre **problem** qui indique le type de problème rencontré. Parmi les problèmes possibles, plusieurs ne doivent pas être envoyés parce qu'ils sont liés à l'exploitation répartie (hypothèse 2), à l'incompatibilité des versions (hypothèse 1) et à l'utilisation de résultats paginés. Ce sont les problèmes **chainingRequired**, **invalidReference**, **timeLimitExceeded**, **outOfScope** et **invalidQueryReference**.

2.2.2.1.7.5 Erreur de mise à jour

```
in-UpdateError ERROR ::= {  
    PARAMETER    IN-UpdateErrorParameter  
    CODE         id-errcode-in-updateError}
```

```
IN-UpdateErrorParameter ::= UpdateErrorParameter  
    (WITH COMPONENTS {  
        problem    (UpdateProblem (1 | 2 | 3 | 4 | 5 | 7))})
```

Le message d'erreur **UpdateError** (erreur de mise à jour) signale des problèmes relatifs à des tentatives d'ajout, de suppression ou de modification en formation dans la base de données de l'annuaire. Il contient uniquement le paramètre **problem** qui indique le type de problème rencontré. Parmi les problèmes possibles, le problème **affectsMultipleDSAs** ne doit pas être envoyé car est lié à l'exploitation répartie (hypothèse 2).

2.2.2.2 Sous-ensemble de protocole d'accès à l'annuaire

2.2.2.2.1 Aperçu général du protocole

2.2.2.2.1.1 Opérations distantes – Spécification et réalisation du système sémaphore

La Rec. X.880 de l'UIT-T | ISO/IEC 13712-1 définit plusieurs classes d'opération sur des objets qui sont utilisés dans la spécification de protocoles d'application pour des opérations distantes telles que les différents protocoles d'annuaire définis dans cette spécification. Un grand nombre de ces classes est utilisé dans des articles ultérieurs. Les techniques de spécification figurant dans la Rec. X.880 de l'UIT-T | ISO/IEC 13712-1 sont utilisées pour définir un protocole générique entre les objets. Une fois qu'ils ont été mis en œuvre dans le protocole d'une couche application du réseau sémaphore, les concepts de la Rec. X.880 de l'UIT-T | ISO/IEC 13712-1 sont mis en correspondance avec ceux du réseau sémaphore décrits dans la Recommandation Q.775. Le protocole de la couche application du réseau sémaphore défini dans cette Recommandation est un protocole destiné à établir une communication entre deux processus de couche application. Dans l'environnement du réseau sémaphore, cela est représenté par la communication entre deux entités d'application (AE) (*application entities*) utilisant les capacités de transaction (TC) (*transaction capabilities*). La fonction d'une entité d'application est fournie par un ensemble d'éléments de service d'application (ASE) (*application-service elements*). L'interaction entre les entités d'application est décrite en termes de leur utilisation des services apportés par les éléments de service d'application. Tous les services fournis par les éléments de service d'application sont contenus dans

une entité d'application unique. Le gestionnaire de composants (CHA) (*component handler*) des capacités de transaction (TC) (*transaction capabilities*) assure le paradigme demande/réponse de l'opération. Les éléments de service d'application de l'annuaire fournissent la fonction de correspondance entre la notation de syntaxe abstraite des progiciels d'exploitation d'annuaire et les services fournis par les capacités de transaction. Le gestionnaire de dialogue (DHA) (*dialogue handler*) de la capacité de transaction assure l'établissement et l'abandon d'une association d'application appelée "dialogue" entre deux entités d'application. Les dialogues entre une DUA et une DSA peuvent être établis uniquement par la DUA.

2.2.2.2.1.2 Objets et contrats pour les opérations effectuées distantes (ROS) sur l'annuaire

La Rec. X.511 de l'UIT-T | ISO/IEC 9594-3 définit le service abstrait entre une DUA et l'annuaire qui fournit un point d'accès en vue d'établir des services d'annuaire. Le paragraphe 2.2.2 définit le sous-ensemble de ce service abstrait utilisé dans le contexte des réseaux intelligents.

La classe **dua** (agent d'utilisateur d'annuaire) d'objets pour les opérations distantes décrit une DUA, qui est une instance de cette classe, comme étant à l'origine du contrat **dapContract** (contrat du protocole d'accès à l'annuaire). Dans ces spécifications d'annuaire, il est fait référence à ce contrat comme le service abstrait de l'annuaire. Il est spécifié au 2.2.2.2.1.3 comme étant un objet d'informations utilisant les opérations distantes (ROS).

```
dua      ROS-OBJECT-CLASS ::= {
        INITIATES   {dapContract}
        ID           id-rosObject-dua}
```

La classe annuaire de l'objet ROS décrit le fournisseur du service abstrait d'annuaire. Ce fournisseur est l'entité répondeuse du contrat **dapContract**.

```
directory  ROS-OBJECT-CLASS ::= {
        RESPONDS    {dapContract}
        ID           id-rosObject-directory}
```

L'annuaire est de plus modélisé comme étant représenté auprès d'un agent DUA par un agent DSA qui prend en charge le point d'accès particulier concerné. Dans le contexte de réseaux intelligents, chaque agent DSA forme potentiellement un point d'accès à l'annuaire.

L'objet **annuaire** se présente comme un ensemble d'agents DSA (résidant dans une entité SDF). Chaque agent DSA contenu dans l'**annuaire** est une instance de la classe **dap-dsa**. Un objet **dap-dsa** assume le rôle d'entité répondeuse dans le contrat **dapContract**.

```
dap-dsa   ROS-OBJECT-CLASS ::= {
        RESPONDS    {dapContract}
        ID           id-rosObject-dapDSA}
```

Les versions ultérieures de la présente Recommandation permettront aux agents DAS d'interagir entre eux afin d'atteindre des objectifs variés.

2.2.2.2.1.3 Contrat et blocs du protocole DAP

Le contrat **dapContract** est défini comme un objet informationnel de la classe **CONTRACT**.

```
dapContract  CONTRACT ::= {
        CONNECTION           dapConnectionPackage
        INITIATOR CONSUMER OF {searchPackage | modifyPackage}
        ID                    id-contract-dap}
```

Lorsque l'agent DUA et l'agent DSA relèvent d'entités physiques différentes du réseau intelligent, ce contrat d'association sera réalisé sous la forme d'un protocole de couche d'application du système SS7, et appelé protocole d'accès à l'annuaire (DAP) (*Directory Access Protocol*) du réseau intelligent. La définition de ce protocole en termes d'un contexte d'application SS7 est fournie dans le 7.2.

Le contrat **dapContract** est composé d'un bloc de connexion, **dapConnectionPackage**, et de deux blocs d'opération, **searchPackage** et **modifyPackage**. Le bloc de connexion **dapConnectionPackage** est défini comme un objet informationnel de la classe **CONNECTION-PACKAGE**. Les opérations de rattachement et de détachement de ce bloc de connexion, **directoryBind** et **directoryUnbind**, sont définies dans la Rec. UIT-T X.511 | ISO/IEC 9594-3.

```
dapConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND           directoryBind
        UNBIND         directoryUnbind
        ID             id-package-dapConnection}
```

Les blocs d'opérations **searchPackage** et **modifyPackage** sont définis comme des objets informationnels de la classe OPERATION-PACKAGE. Les opérations de ces blocs d'opération sont définies dans la Rec. UIT-T X.511 | ISO/IEC 9594-3. Celle-ci définit des opérations supplémentaires pour prendre en charge l'accès à l'annuaire. De telles opérations ne sont pas utilisées dans le contexte des réseaux intelligents.

```
searchPackage    OPERATION-PACKAGE ::= {
    CONSUMER INVOKES    {search}
    ID                    id-package-search}
modifyPackage    OPERATION-PACKAGE ::= {
    CONSUMER INVOKES    {addEntry | removeEntry | modifyEntry}
    ID                    id-package-modify}
```

NOTE – Ces blocs, lorsqu'ils sont réalisés sous forme d'éléments ASE, servent à la construction des contextes d'application définis dans la présente Specification. Ils n'ont pas pour objet d'étager les déclarations de conformité relatives à des éléments ASE isolés ou combinés.

Puisque l'agent DUA est l'initiateur du contrat **dapContract**, il assume le rôle de consommateur des blocs d'opération du contrat. Cela signifie que seul l'agent DUA peut invoquer les opérations de ce contrat et de sa réalisation SS7.

2.2.2.2.1.4 Utilisation des sous-jacents systèmes

Le protocole DAP utilise comme suit les services sous-jacents:

2.2.2.2.1.4.1 Utilisation des services de traitement de composante

Les éléments ASE d'annuaire sont des utilisateurs des services de traitement de composante des capacités de transaction, à l'exception des services TC-L-REJECT et TC-L-CANCEL qui sont utilisés par le processus d'application. La réception d'une primitive d'indication TC-L-REJECT ou TC-L-CANCEL, conduit le processus d'application à abandonner le dialogue (c'est-à-dire qu'il émet alors une primitive de demande TC-U-ABORT).

Le service TC-U-CANCEL n'est jamais utilisé.

2.2.2.2.1.4.2 Utilisation des services de traitement du dialogue

Les services de traitement du dialogue servent à prendre en charge les opérations DirectoryBind et DirectoryUnbind, et à déclencher l'envoi d'unités PDU d'application associées aux opérations intervenant dans les blocs de l'annuaire.

Le groupement de composantes est exécuté sous le contrôle du processus d'application par une utilisation appropriée des services TC-BEGIN et TC-CONTINUE.

Le service TC-END est utilisé uniquement pour prendre en charge la procédure de détachement (c'est-à-dire qu'il n'est jamais utilisé pour déclencher l'envoi de composantes).

Si l'entité SDF ne reçoit qu'une primitive de demande TC-CONTINUE vide, elle l'ignorera.

Si l'entité SDF reçoit une primitive de demande TC-END avec une requête de base de données, elle n'exécutera pas la requête et considérera la demande de service TC-END comme une procédure de détachement. Le dialogue est alors terminé (voir 2.2.2.2.3.1.2).

Si l'entité SDF reçoit une primitive d'indication TC-U-REJECT, elle l'ignorera.

Si l'entité SDF reçoit une primitive d'indication TC-R-REJECT, elle abandonnera le dialogue avec une primitive de demande TC-U-ABORT.

Si l'entité SDF déteste des situations de rejet, elle enverra une primitive de demande TC-U-REJECT suivie d'une primitive de demande TC-CONTINUE.

La procédure de terminaison préconvenue n'est jamais utilisée.

Le processus d'application est l'unique utilisateur des services TC-P-ABORT et TC-NOTICE.

La réception d'une primitive d'indication TC-U-ABORT ou d'une primitive d'indication TC-P-ABORT au cours d'un dialogue met fin à toutes les requêtes en cours. Il appartient au processus d'application de confirmer si les modifications demandées ont été réalisées.

Il appartient au processus d'application de fournir dans la primitive de demande TC-BEGIN une adresse de destination qui puisse être utilisée par le sous-système SCCP pour acheminer le message à l'entité SDF appropriée.

2.2.2.2.2 Syntaxe abstraite du protocole d'annuaire

2.2.2.2.2.1 Syntaxes abstraites

La présente version du protocole d'accès à l'annuaire exige la prise en charge de trois syntaxes abstraites:

- la syntaxe abstraite des unités de données protocolaires de commande de dialogue TC, **dialogue-abstract-syntax** (syntaxe abstraite de dialogue), qui est nécessaire à l'établissement des dialogues entre unités SCF et unités SDF, et qui est spécifié dans la Recommandation Q.773;
- la syntaxe abstraite d'acheminement des unités de données protocolaires, destinées à invoquer pour appeler les opérations **directoryBind** et **directoryUnbind** et à en communiquer les résultats;
- la syntaxe abstraite d'acheminement des unités de données protocolaires destinées à invoquer les opérations intervenant dans les blocs d'opérations spécifiés au 2.2.2.2.1.3 et à en communiquer les résultats.

Le type ASN.1 dont dérivent les valeurs de la deuxième syntaxe abstraite est spécifié au moyen des types paramétrés, **Bind** {} et **Unbind** {} définis dans la Recommandation X.880.

Le type ASN.1 dont dérivent les valeurs de la dernière syntaxe abstraite est spécifié au moyen des types paramétrés **TCAPMessage** {} définis dans la Recommandation Q.773.

Toutes ces syntaxes abstraites seront (au minimum) codées selon les règles de codage de base ASN.1 avec les restrictions énumérées dans la Recommandation Q.773.

2.2.2.2.2.1.1 Syntaxe abstraite du protocole DAP

Les éléments ASE d'annuaire qui effectuent les blocs d'opération spécifiés au 2.2.2.2.1.3 partagent une même syntaxe abstraite, **directoryOperationsAbstractSyntax**. Celle-ci est spécifiée comme un objet informationnel de la classe ABSTRACT-SYNTAX.

```
inDirectoryOperationsAbstractSyntax      ABSTRACT-SYNTAX ::= {
    BasicDAP-PDUs
    IDENTIFIED BY                          id-as-directoryOperationsAS}
```

```
BasicDAP-PDUs ::= TCAPMessage {{DAP-Invokable},{DAP-Returnable}}
```

```
DAP-Invokable      OPERATION      ::= {search | addEntry | removeEntry | modifyEntry}
```

```
DAP-Returnable     OPERATION      ::= {search | addEntry | removeEntry | modifyEntry}
```

La réalisation du bloc de connexion spécifié au 2.2.2.2.1.3 utilise une syntaxe abstraite distincte, **directoryBindingAbstractSyntax**. Celle-ci est spécifiée comme un objet informationnel de la classe ABSTRACT-SYNTAX.

```
InDirectoryBindingAbstractSyntax        ABSTRACT-SYNTAX ::= {
    DAPBinding-PDUs
    IDENTIFIED BY                          id-as-directoryBindingAS}
```

```
DAPBinding-PDUs ::= CHOICE {
    bind          Bind {directoryBind},
    unbind       Unbind {directoryUnbind}}
```

2.2.2.2.2.2 Contextes d'application d'annuaire

2.2.2.2.2.2.1 Contexte d'application d'accès à l'annuaire

Le contrat **dapContract** (contrat de protocole d'accès à l'annuaire) est réalisé sous la forme du contexte d'application **inDirectoryAccessAC**. Ce contexte d'application est spécifié comme un objet informationnel de la classe APPLICATION-CONTEXT

```
inDirectoryAccessAC      APPLICATION-CONTEXT ::= {
    CONTRACT                dapContract
    DIALOGUE MODE           structured
    TERMINATION             basic
    ABSTRACT SYNTAXES      {dialogue-abstract-syntax |
                            inDirectoryOperationsAbstractSyntax |
                            inDirectoryBindingAbstractSyntax }
    APPLICATION CONTEXT NAME id-ac-directoryAccessAC}
```

2.2.2.2.3 Codes d'opération

Les opérations intervenant dans les blocs définis dans la présente Recommandation sont spécifiées dans la Recommandation X.519; les codes d'opérations affectés correspondants sont importés de la Recommandation X.519.

2.2.2.2.4 Codes d'erreur

Les messages d'erreurs intervenant dans les blocs définis dans la présente Recommandation sont spécifiés dans la Recommandation X.519; les codes d'erreurs affectés correspondants sont importés de la Recommandation X.519.

2.2.2.2.5 Versions et règles d'extensibilité

L'annuaire peut être réparti et plusieurs entités d'application d'annuaire peuvent interagir pour satisfaire une demande. Les entités d'application d'annuaire peuvent être réalisées selon différentes versions des spécifications d'annuaire, représentée ou non par leurs numéros de version de protocole. Le numéro de version négocié par chaque couple d'entités d'application d'annuaire directement rattachées correspond au plus grand numéro commun de version.

2.2.2.2.5.1 Négociation de version

Lors de l'acceptation d'une association, c'est-à-dire d'un rattachement, au moyen du protocole DAP, la version négociée n'affectera que les éléments point à point du protocole échangés entre l'agent DUA et l'agent DSA auquel il est connecté. Les demandes ou réponses ultérieures transmises sur cette association ne seront pas contraintes par la version négociée.

NOTE – Il n'existe pas d'aspects point à point du protocole DAP qui relèvent actuellement de versions protocolaires différentes.

2.2.2.2.5.2 Côté agent DUA

2.2.2.2.5.2.1 Traitement de demande et de réponse côté agent DUA

L'agent DUA peut lancer des requêtes selon la version la plus récente des spécifications qu'il prend en charge pour cette demande. Si un ou plusieurs éléments de la requête sont critiques, il indique le numéro d'extension dans le paramètre critical **Extensions**.

NOTE 1 – Si l'information que l'extension a remplacée dans un type choix, énuméré ou entier (utilisé comme énuméré) est essentielle au bon fonctionnement d'un agent DSA réalisé selon une version plus ancienne de la Spécification, il est recommandé de marquer l'extension comme étant critique.

Lors du traitement d'une réponse, un agent DUA:

- a) ignorera toutes les affectations de bit qui lui sont inconnues dans une chaîne;
- b) ignorera tous les numéros qui lui sont inconnus dans un type énuméré ou un type entier utilisé comme un énuméré, sous réserve que le numéro fourni corresponde à un élément optionnel de type ensemble ou séquence;
- c) ignorera les éléments qui lui sont inconnus dans un type ensemble, à la fin d'un type séquence, ou dans un type choix lorsque celui-ci est un élément optionnel d'une structure de type ensemble ou séquence;

NOTE 2 – Les applications peuvent par choix local ignorer certains éléments supplémentaires dans une unité PDU d'annuaire. En particulier, il leur est possible d'ignorer certains numéros nommés et des types choix non répertoriés dans les éléments obligatoires de type ensemble et séquence sans mettre en cause la validité de l'opération. L'identification de tels éléments nécessite un complément d'étude.

- d) ne considérera pas la réception de types d'attribut et de valeurs d'attributs non répertoriés comme une violation du protocole;
- e) pourra en option signaler à l'utilisateur les types d'attributs et les valeurs d'attributs non répertoriés.

2.2.2.2.5.2.2 Règles d'extensibilité pour le traitement d'erreurs côté agent DUA

Lorsqu'il traite un type d'erreur répertorié indiquant un problème ou comportant un paramètre non répertorié, un agent DUA:

- a) ne considérera pas la réception du problème ou du paramètre non répertorié comme une violation du protocole (c'est-à-dire, qu'il n'émettra pas de message TC-U-REJECT ni n'interrompra le dialogue);
- b) pourra en option signaler l'information d'erreur supplémentaire à l'utilisateur.

Lorsqu'il traite un type d'erreur non répertorié, l'agent DUA:

- a) ne considérera pas la réception du type d'erreur non répertorié comme une violation du protocole (c'est-à-dire, qu'il n'émettra pas de message TC-U-REJECT ni n'interrompra l'association d'application);
- b) pourra en option signaler l'erreur à l'utilisateur.

2.2.2.2.5.3 Traitement de la requête côté agent DSA

Si un agent DSA exécutant une opération détecte un élément **criticalExtensions** (extensions critiques) dont la sémantique ne lui est pas connue, il renverra un message d'erreur avec le motif **unavailableCriticalExtension** (extension critique non disponible).

NOTE 1 – Les valeurs nulles dans une chaîne **criticalExtensions** indiquent soit que les extensions correspondantes ne sont pas présentes ou qu'elles ne sont pas critiques. La présence d'une valeur nulle dans une chaîne **criticalExtensions** n'impliquera ni la présence ni l'absence de l'extension correspondante dans l'unité APDU.

Sinon, lorsqu'il traite une demande émanant d'un agent DUA, un agent DSA:

- a) ignorera tous les bits nommés qui lui sont inconnus dans une chaîne binaire;
- b) ignorera les numéros nommés qui lui sont inconnus dans un type énuméré ou dans un type entier utilisé comme un énuméré, pour autant que le numéro figure en tant qu'élément optionnel dans un type ensemble ou séquence;
- c) ignorera les éléments qui lui sont inconnus dans les types ensemble, à la fin des types séquence, et dans le type choix lorsque celui-ci est en élément optionnel dans un type ensemble ou séquence.

NOTE 2 – Les applications peuvent sur choix local ignorer certains éléments supplémentaires dans une unité PDU d'annuaire. En particulier, des numéros nommés et des types choix non répertoriés dans les éléments obligatoires des types ensemble et séquence peuvent être ignorés sans porter atteinte à la validité de l'opération. L'identification de tels éléments nécessite un complément d'étude.

2.2.2.2.3 Mappage sur les services utilisés

Ce paragraphe définit le mappage du protocole DAP sur les services de capacités de transaction (TC).

2.2.2.2.3.1 Mappage sur les services de dialogue

Ce paragraphe définit le mappage des services DirectoryBind et DirectoryUnbind sur les services de traitement de dialogue de capacité de transaction définis dans la Recommandation Q.771.

2.2.2.2.3.1.1 Rattachement

Le service DirectoryBind est mappé sur les services de capacité de transaction de la manière suivante:

- a) le service TC-BEGIN est utilisé pour invoquer l'opération **DirectoryBind**;
- b) le service TC-CONTINUE est utilisé pour signaler le succès de l'opération **DirectoryBind**.
- c) le service TC-U-ABORT est utilisé pour signaler l'échec de l'opération **DirectoryBind**.

L'utilisation des paramètres de ces services est spécifiée dans les paragraphes suivants:

2.2.2.2.3.1.1.1 Mappage sur le service TC-BEGIN

2.2.2.2.3.1.1.1.1 Qualité de service

Ce paramètre sera utilisé conformément à la Recommandation Q.771. Aucune contrainte particulière n'est imposée par la présente Recommandation.

2.2.2.2.3.1.1.1.2 Adresse de destination

Ce paramètre contiendra l'adresse de l'entité DSF qui joue le rôle de l'agent DSA.

2.2.2.2.3.1.1.1.3 Nom du contexte d'application

Ce paramètre prendra la valeur du champ nom du contexte d'application de l'objet **iNdirectoryAccessAC**.

2.2.2.2.3.1.1.1.4 Adresse d'origine

Ce paramètre contiendra l'adresse de l'entité SCF dans laquelle l'agent DUA réside.

2.2.2.2.3.1.1.1.5 Identificateur de dialogue

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.1.1.6 User Information

Ce paramètre contiendra une valeur du type **DirectoryBindArgument**.

2.2.2.2.3.1.1.1.7 Composante Present (présent)

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.1.2 Mappage sur le service TC-CONTINUE

2.2.2.2.3.1.1.2.1 Qualité de service

Ce paramètre sera utilisé conformément à la Recommandation Q.771. Aucune contrainte particulière n'est imposée par la présente Recommandation.

2.2.2.2.3.1.1.2.2 Adresse d'origine

Ce paramètre sera utilisé conformément à la Recommandation Q.771. S'il est présent, il contiendra l'adresse de l'entité SDF qui joue le rôle de l'agent DSA.

2.2.2.2.3.1.1.2.3 Nom du contexte d'application

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.1.2.4 Identificateur de dialogue

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

L'identificateur de lien autorisé peut être mappé sur l'identificateur de dialogue TCAP.

2.2.2.2.3.1.1.2.5 User Information

Ce paramètre contiendra une valeur de type **DirectoryBindResult**.

2.2.2.2.3.1.1.2.6 Composante Present

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.1.3 Mappage sur le système TC-U-ABORT

2.2.2.2.3.1.1.3.1 Qualité de service

Ce paramètre sera utilisé conformément à la Recommandation Q.771. Aucune contrainte particulière n'est imposée par la présente Recommandation.

2.2.2.2.3.1.1.3.2 Identificateur de dialogue

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.1.3.3 Motif d'abandon

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.1.3.4 Nom du contexte d'application

Ce paramètre sera utilisé conformément à la Recommandation Q.771. Lorsque l'entité SDF refuse un dialogue parce qu'elle ne prend pas en charge le nom du contexte d'application reçu ce paramètre aura la valeur du champ du nom du contexte d'application de l'objet **iNdirectoryAccessAC**.

2.2.2.2.3.1.1.3.5 Information utilisateur

Lorsque le paramètre motif d'abandon a la valeur "dialogue refusé", ce paramètre contiendra une valeur du type **DirectoryBindError**. Sinon il sera absent.

2.2.2.2.3.1.2 Détachement

Le service **DirectoryUnbind** est projeté sur le service TC-END. L'utilisation des paramètres du service TC-END sont spécifiés dans les paragraphes suivants:

2.2.2.2.3.1.2.1 Qualité de service

Ce paramètre sera utilisé conformément à la Recommandation Q.771. Aucune contrainte particulière n'est imposée par la présente Recommandation.

2.2.2.2.3.1.2.2 Nom du contexte d'application

Ce paramètre n'est pas utilisé dans cette phase de dialogue.

2.2.2.2.3.1.2.3 Identificateur de dialogue

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.1.2.4 Information utilisateur

Ce paramètre sera vide.

2.2.2.2.3.1.2.5 Composante Present

Ce paramètre sera utilisé conformément à la Recommandation Q.771.

2.2.2.2.3.2 Projection sur les services de traitement de composante

Les services ASE de l'annuaire sont projetés sur les services de traitement de composante des capacités de transaction. La projection des opérations et des erreurs sur les services des capacités de transaction est définie dans la Recommandation Q.774.

Le paramètre de temporisation des primitives de demande TC-INVOKE est réglé selon le Tableau 3.

TABLEAU 3/Q.1218

Valeurs de temporisation des capacités de transaction des opérations du protocole DAP

Opération	Temporisation
search (recherche)	à étudier
modifyEntry (modification d'entrée)	à étudier
addEntry (ajout d'entrée)	à étudier
removeEntry (suppression d'entrée)	à étudier

2.2.2.2.4 Conformité

Ce paragraphe définit les prescriptions de conformité de la présente Spécification.

2.2.2.2.4.1 Conformité des entités SCF

Une réalisation d'entité SCF se prétendant conforme à la présente Spécification satisfera les prescriptions définies du 2.2.2.2.4.1.1 au 2.2.2.2.4.1.3.

2.2.2.2.4.1.1 Prescriptions de déclaration

La déclaration de conformité comportera les éléments suivants:

- les opérations du contexte d'application **iNdirectoryAccessAC** que l'entité SCF est capable d'invoquer et pour lesquelles la conformité est revendiquée;
- le ou les niveaux de sécurité pour lesquels la conformité est revendiquée (aucun, simple, élevé);
- les extensions énumérées dans le tableau du 7.3.1 de la Rec. UIT-T X.511 | ISO/IEC 9594-3, que l'entité SCF est capable d'initier et pour lesquelles la conformité est revendiquée.

2.2.2.2.4.1.2 Prescriptions statiques

Une entité SCF:

- aura la capacité de prendre en charge le contexte d'application **iNdirectoryAccessAC** défini par sa syntaxe abstraite au 2.2.2.2.2;
- sera conforme aux extensions pour lesquelles elle prétend à conformité comme indiqué dans l'alinéa c) du 2.2.2.2.4.1.1.

2.2.2.2.4.1.3 Dynamic requirements

An SCF shall:

- a) conform to the mapping onto used services defined in 2.2.2.2.3;
- b) shall conform to the rules of extensibility procedures defined in 2.2.2.2.5.2.

2.2.2.2.4.2 Conformance by SDFs

An SDF implementation claiming conformance to this Specification shall satisfy the requirements specified in 2.2.2.2.4.2.1 through 2.2.2.2.4.2.3.

2.2.2.2.4.2.1 Statement requirements

The following shall be stated:

- a) The application-context for which conformance is claimed. The present version of this Recommendation only requires conformance to the **iNdirectoryAccessAC** application-context.

NOTE – An application context shall not be divided except as stated herein; in particular, conformance shall not be claimed to particular operations.
- b) The security-level(s) for which conformance is claimed (none, simple, strong).
- c) The attribute types for which conformance is claimed and whether for attributes based on the syntax **DirectoryString**, conformance is claimed for the **UNIVERSAL STRING** choice.
- d) The object classes, for which conformance is claimed.
- e) The extensions listed in the table of 7.3.1 of ITU-T Rec. X.511 | ISO/IEC 9594-3, that the SDF is capable of responding to for which conformance is claimed.
- f) Whether conformance is claimed for collective attributes as defined in 8.8 of ITU-T Rec. X.501 | ISO/IEC 9594-2 and 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3.
- g) Whether conformance is claimed for hierarchical attributes as defined in 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3.
- h) The operational attribute types defined in ITU-T Rec. X.501 | ISO/IEC 9594-2 and any other operational attribute types for which conformance is claimed.
- i) Whether conformance is claimed for return of alias names as described in 7.7.1 of ITU-T Rec. X.511 | ISO/IEC 9594-3.
- j) Whether conformance is claimed for indicating that returned entry information is complete, as described in 7.7.6 of ITU-T Rec. X.511 | ISO/IEC 9594-3.
- k) Whether conformance is claimed for modifying the object class attribute to add and/or remove values identifying auxiliary object classes, as described in 11.3.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3.
- l) Whether conformance is claimed to Basic Access Control.
- m) Whether conformance is claimed to Simplified Access Control.
- n) The name bindings for which conformance is claimed.
- o) Whether the SDF is capable of administering collective attributes, as defined in ITU-T Rec. X.501 | ISO/IEC 9594-2
- p) Whether conformance is claimed for contexts.

2.2.2.2.4.2.2 Static requirements

An SDF shall:

- a) have the capability of supporting the application-contexts for which conformance is claimed as defined by their abstract syntax in 2.2.2.2.2;
- b) have the capability of supporting the information framework defined by its abstract syntax in ITU-T Rec. X.501 | ISO/IEC 9594-2;
- c) have the capability of supporting the attribute types for which conformance is claimed; as defined by their abstract syntaxes;

- d) have the capability of supporting the object classes for which conformance is claimed, as defined by their abstract syntaxes;
- e) conform to the extensions for which conformance was claimed in 2.2.2.2.4.2.1;
- f) if conformance is claimed for collective attributes, have the capability of performing the related procedures defined in 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;
- g) if conformance is claimed for hierarchical attributes, have the capability of performing the related procedures defined in 7.6, 7.8.2 and 9.2.2 of ITU-T Rec. X.511 | ISO/IEC 9594-3;
- h) have the capability of supporting the operational attribute types for which conformance is claimed;
- i) if conformance is claimed to Basic Access Control, have the capability of holding ACI items that conform to the definitions of Basic Access Control;
- j) if conformance is claimed to Simplified Access Control, have the capability of holding ACI items that conform to the definitions of Simplified Access Control.

2.2.2.2.4.2.3 Dynamic requirements

An SDF shall:

- a) conform to the mapping onto used services defined in 2.2.2.2.3;
- b) conform to the rules of extensibility procedures defined in 2.2.2.2.2.5.3;
- c) if conformance is claimed to Basic Access Control, have the capability of protecting information within the SDF in accordance with the procedures of Basic Access Control;
- d) if conformance is claimed to Simplified Access Control, have the capability of protecting information within the SDF in accordance with the procedures of Simplified Access Control.

2.2.2.3 X.501 profile

The Recommendation X.501 provides a generic information model that is needed to support the service provided by the Directory. In the context of IN, the generic information model should conform to the part of the Recommendation going from clauses 1 to 7 of the Recommendation X.501. However certain aspects of this Recommendation need not to be supported. This includes the DIT content rules whose use is a local matter.

Some other points are outside the scope of this Recommendation. This concerns the distributed procedures, the replications procedures and other items associated with capabilities not covered by IN CS-1. Therefore the following parts of Rec. X.501 are not applicable:

- Subclause 3.3 Distributed operation definitions;
- Subclause 3.4 Replication definitions;
- last paragraph of 6.3.2;
- paragraphs f), h) and i) in 16.2.3;
- paragraph a) in 16.2.4. The compare operation is not used, the search operation is used instead. Therefore the FilterMatch permission replaces the Compare permission.

In this subclause when it is referred to Recommendation X.501, it refers to the modified version of the Recommendation X.501 including the amendments made to it (DAM2 on contexts and DAM3 on minor extensions).

2.2.2.4 ASN.1 Profile of the Directory Abstract Service for the IN CS-1

2.2.2.4.1 ASN.1 Module for IN X.500 profile

This subclause contains the full ASN.1 profiling of the Directory Abstract Service. It has been successfully compiled.

IN-DirectoryAbstractService {ccitt recommendation q 1218 modules (0) abstractService (15) version 1 (0)}

DEFINITIONS ::=

BEGIN

IMPORTS

attributeError, ServiceControls, EntryInformation, EntryInformationSelection, DirectoryBindArgument, Credentials, SearchArgument, SearchResult, SearchInfo, PartialOutcomeQualifier, AddEntryArgument, RemoveEntryArgument, ModifyEntryArgument, ServiceProblem, UpdateProblem, SecurityProblem, directoryBindError, AddEntryResult, BindErrorParameter, ModifyEntryResult, RemoveEntryResult, SecurityErrorParameter, ServiceErrorParameter, AttributeErrorParameter, UpdateErrorParameter,

FROM DirectoryAbstractService {joint-iso-ccitt ds (5) module (1) directoryAbstractService (2) 2}

Code, OPERATION, ERROR

FROM Remote-Operations-Information-Objects {joint-iso-ccitt remote-operations (4) informationObjects (5) version1 (0)};

IN-ServiceControls ::= ServiceControls

(WITH COMPONENTS {
options {{chainingProhibited, localScope}}
{chainingProhibited, localScope, dontDereferenceAliases, subentries}}
{chainingProhibited, localScope, dontDereferenceAliases}}
{chainingProhibited, localScope, subentries}...),
priority ,
timeLimit ABSENT,
sizeLimit ABSENT,
scopeOfReferral ABSENT,
attributeSizeLimit ABSENT},
...)

IN-EntryInformationSelection ::= EntryInformationSelection

(WITH COMPONENTS {
attributes ,
infoTypes (attributeTypesAndValues),
extraAttributes OPTIONAL,
contextSelection OPTIONAL,
returnContexts },
...)

IN-EntryInformation ::= EntryInformation

(WITH COMPONENTS {
name PRESENT,
fromEntry (TRUE),
information (WITH COMPONENTS {
attributeType ABSENT,
attribute PRESENT},
...) OPTIONAL,
incompleteEntry ,
partialNameResolution (FALSE ...)},
...)

UNSIGNED {Type} ::= OPTIONALLY-SIGNED {Type}

(WITH COMPONENTS {
unsigned PRESENT,
signed ABSENT} ...)

in-DirectoryBind OPERATION ::= {

ARGUMENT IN-DirectoryBindArgument
RESULT IN-DirectoryBindResult
ERRORS {in-DirectoryBindError}}

IN-DirectoryBindArgument ::= DirectoryBindArgument

(WITH COMPONENTS {
credentials (IN-Credentials) OPTIONAL,
versions } ...)

IN-Credentials ::= Credentials

(WITH COMPONENTS {
simple ,
strong ABSENT,
externalProcedure } ...)

IN-DirectoryBindResult ::= IN-DirectoryBindArgument

**in-DirectoryBindError ERROR ::= {
 PARAMETER IN-DirectoryBindErrorParameter}**

**IN-DirectoryBindErrorParameter ::= DirectoryBindErrorParameter
 (WITH COMPONENTS {
 versions ,
 error (WITH COMPONENTS {
 securityError (SecurityProblem (1|2|7 ...)),
 serviceError (ServiceProblem (2 ...))} ...)} ...)**

**in-Search OPERATION ::= {
 ARGUMENT IN-SearchArgument
 RESULT IN-SearchResult
 ERRORS {nameError | in-ServiceError | in-SecurityError | attributeError}
 CODE id-opcode-in-search}**

**IN-SearchArgument ::= UNSIGNED {SearchArgument
 (WITH COMPONENTS {
 baseObject PRESENT,
 subset ,
 filter ,
 searchAliases (TRUE ...),
 selection (IN-EntryInformationSelection ...),
 pagedResults ABSENT,
 matchedValuesOnly ,
 extendedFilter ABSENT
 serviceControls (IN-ServiceControls ...),
 securityParameters OPTIONAL,
 requestor ABSENT,
 operationProgress ({nameResolutionPhase notStarted} ...),
 aliasedRDNs ABSENT,
 criticalExtensions OPTIONAL,
 referenceType ABSENT,
 entryOnly (TRUE ...),
 exclusions ABSENT,
 nameResolveOnMaster (FALSE ...),
 checkOverespecified },
 ...)}
 ...}**

**IN-SearchResult ::= UNSIGNED {SearchResult
 (WITH COMPONENTS {
 searchInfo (WITH COMPONENTS {
 name ,
 entries (WITH COMPONENT (IN-EntryInformation ...) ...),
 partialOutcomeQualifier PartialOutcomeQualifier
 (WITH COMPONENTS {
 limitProblem OPTIONAL,
 unexplored ABSENT
 unavailableCriticalExtensions ,
 unknownErrors ABSENT,
 queryReference ABSENT,
 overspectFilter OPTIONAL,
 ...} ...),
 securityParameters ABSENT,
 performer ABSENT,
 aliasDereferenced },
 ...),
 uncorrelatedSearchInfo ABSENT},
 ...)}
 ...}**

**in-AddEntry OPERATION ::= {
 ARGUMENT IN-AddEntryArgument
 RESULT AddEntryResult
 ERRORS {nameError | in-ServiceError | in-SecurityError | attributeError | in-UpdateError}
 CODE id-opcode-in-addEntry}**

IN-AddEntryArgument ::= UNSIGNED {AddEntryArgument
(WITH COMPONENTS {
 object **PRESENT,**
 entry **PRESENT,**
 targetSystem **ABSENT,**
 serviceControls **(IN-ServiceControls),**
 securityParameters **OPTIONAL,**
 requestor **ABSENT,**
 operationProgress **({nameResolutionPhase notStarted}),**
 aliasedRDNs **ABSENT,**
 criticalExtensions **OPTIONAL,**
 referenceType **ABSENT,**
 entryOnly **(TRUE),**
 exclusions **ABSENT,**
 nameResolveOnMaster **(FALSE)},**
 ...}

in-RemoveEntry OPERATION ::= {
 ARGUMENT **IN-RemoveEntryArgument**
 RESULT **RemoveEntryResult**
 ERRORS **{nameError | in-ServiceError | in-SecurityError | in-UpdateError}**
 CODE **id-opcode-in-removeEntry}**

IN-RemoveEntryArgument ::= UNSIGNED {RemoveEntryArgument
(WITH COMPONENTS {
 object **PRESENT,**
 serviceControls **(IN-ServiceControls),**
 securityParameters **OPTIONAL,**
 requestor **ABSENT,**
 operationProgress **({nameResolutionPhase notStarted}),**
 aliasedRDNs **ABSENT,**
 criticalExtensions **OPTIONAL,**
 referenceType **ABSENT,**
 entryOnly **(TRUE),**
 exclusions **ABSENT,**
 nameResolveOnMaster **(FALSE)},**
 ...}

in-ModifyEntry OPERATION ::= {
 ARGUMENT **IN-ModifyEntryArgument**
 RESULT **ModifyEntryResult**
 ERRORS **{nameError | in-ServiceError | in-SecurityError | attributeError | in-UpdateError}**
 CODE **id-opcode-in-modifyEntry}**

IN-ModifyEntryArgument ::= UNSIGNED {ModifyEntryArgument
(WITH COMPONENTS {
 object **PRESENT,**
 changes **PRESENT,**
 selection **(IN-EntryInformationSelection),**
 serviceControls **(IN-ServiceControls),**
 securityParameters **OPTIONAL,**
 requestor **ABSENT,**
 operationProgress **({nameResolutionPhase notStarted}),**
 aliasedRDNs **ABSENT,**
 criticalExtensions **OPTIONAL,**
 referenceType **ABSENT,**
 entryOnly **(TRUE),**
 exclusions **ABSENT,**
 nameResolveOnMaster **(FALSE)},**
 ...}

IN-ModifyEntryResult ::= ModifyEntryResult
(WITH COMPONENTS {
 null **,**
 information **UNSIGNED {Information**
 (WITH COMPONENTS {

```

entry (IN-EntryInformation ....),
securityParameters OPTIONAL,
performer ABSENT,
aliasDereferenced },
...));
...

```

```

in-SecurityError ERROR ::= {
    PARAMETER IN-SecurityErrorParameter
    CODE      id-errcode-in-securityError}

```

```

IN-SecurityErrorParameter ::= SecurityErrorParameter
(WITH COMPONENTS {
    problem (SecurityProblem (1 | 2 | 3 | 6|7 ....) ....)
}

```

```

in-ServiceError ERROR ::= {
    PARAMETER IN-ServiceErrorParameter
    CODE      id-errcode-in-serviceError}

```

```

IN-ServiceErrorParameter ::= ServiceErrorParameter
(WITH COMPONENTS {
    problem (ServiceProblem (1 | 2 | 3 | 5 | 8 | 9 | 10 | 12 ....) ....)
}

```

```

in-UpdateError ERROR ::= {
    PARAMETER IN-UpdateErrorParameter
    CODE      id-errcode-in-updateError}

```

```

IN-UpdateErrorParameter ::= UpdateErrorParameter
(WITH COMPONENTS {
    problem (UpdateProblem (1|2|3|4|5|7 ....) ....)
}

```

```

id-errcode-in-serviceError Code ::= local:2
id-errcode-in-securityError Code ::= local:3
id-errcode-in-updateError Code ::= local:4

```

```

id-opcode-in-modifyEntry Code ::=local:1
id-opcode-in-addEntry Code ::=local:2
id-opcode-in-removeEntry Code ::=local:3
id-opcode-in-search Code ::=local:4

```

END

2.2.2.4.2 ASN.1 Modules for the DAP

This subclause includes all of the ASN.1 type and value definitions contained in this Directory Specification, in the form of the ASN.1 module, "DirectoryAccessProtocol". It also includes all of the ASN.1 Object Identifiers assigned in this Specification, in the form of a ASN.1 module, "ProtocolObjectIdentifiers".

```

INDirectoryAccessProtocol {ccitt recommendation q 1218 module(0) indap(12) version1( 0)}

```

```

DEFINITIONS ::=

```

```

BEGIN

```

```

-- EXPORTS All --

```

```

-- The types and values defined in this module are exported for use in the other ASN.1 modules contained
-- within the Directory Specifications, and for the use of other applications which will use them to access
-- Directory services. Other applications may use them for their own purposes, but this will not constrain
-- extensions and modifications needed to maintain or improve the Directory service.

```

```

IMPORTS

```

```

    directoryAbstractService
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 2}

```

```

    ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE,
    Code, OPERATION

```

```

    FROM Remote-Operations-Information-Objects
        {joint-iso-ccitt remote-operations(4) informationObjects(5) version1(0)}

```

```

    Bind{}, Unbind{}, InvokeID
    FROM Remote-Operations-Generic-ROS-PDUs
        {joint-iso-ccitt remote-operations(4) generic-ROS-PDUs(6) version1(0)}

```

```

TCAPMessage {}
    FROM TCAPMessages
        {ccitt recommendation q 773 modules(2) messages(1) version3(3)}

APPLICATION-CONTEXT
    FROM TC-Notation-Extension
        {ccitt recommendation q 775 modules(2) notation-extensions (4) version1(1)}

dialogue-as-id
    FROM DialoguePDUs
        {ccitt recommendation q 773 modules(2) dialoguePDUs (2) version1(1)}

id-ac-directoryAccessAC, id-rosObject-dua, id-rosObject-directory, id-rosObject-dapDSA,
id-contract-dap, id-package-dapConnection, id-package-search, id-package-modify,
id-as-directoryOperationsAS, id-as-directoryBindingAS
    FROM SDFProtocolObjectIdentifiers
        {ccitt recommendation q 1218 modules(0) sdfProtocolObjectIdentifiers (10) version1 (0)}

directoryUnbind FROM DirectoryAbstractService directoryAbstractService ;

directoryBind, search, addEntry, removeEntry, modifyEntry
    FROM ExtendedDirectoryAbstractService
        {ccitt recommendation q 1218 modules(0) informationFramework (11) version1( 0)}

```

-- application contexts --

```

inDirectoryAccessAC      APPLICATION-CONTEXT ::=
{
    CONTRACT                dapContract
    DIALOGUE MODE           structured
    TERMINATION              basic
    ABSTRACT SYNTAXES       { dialogue-abstract-syntax |
                            inDirectoryOperationsAbstractSyntax |
                            inDirectoryBindingAbstractSyntax }
    APPLICATION CONTEXT NAME id-ac-directoryAccessAC
}

```

-- ROS-objects --

```

dua                      ROS-OBJECT-CLASS ::=
{
    INITIATES               { dapContract }
    ID                      id-rosObject-dua
}

directory                ROS-OBJECT-CLASS ::=
{
    RESPONDS                { dapContract }
    ID                      id-rosObject-directory
}

dap-dsa                  ROS-OBJECT-CLASS ::=
{
    RESPONDS                { dapContract }
    ID                      id-rosObject-dapDSA
}

```

-- contracts --

```

dapContract              CONTRACT ::=
{
    CONNECTION              dapConnectionPackage
    INITIATOR CONSUMER OF  {searchPackage | modifyPackage }
    ID                      id-contract-dap
}

```

```

-- connection package --
dapConnectionPackage      CONNECTION-PACKAGE ::=
{
    BIND                directoryBind
    UNBIND              directoryUnbind
    ID                  id-package-dapConnection
}

-- search package --
searchPackage             OPERATION-PACKAGE ::=
{
    CONSUMER INVOKES   {search }
    ID                 id-package-search
}

-- modify package --
modifyPackage             OPERATION-PACKAGE ::=
{
    CONSUMER INVOKES   { addEntry | removeEntry | modifyEntry}
    ID                 id-package-modify
}

-- abstract-syntaxes --
inDirectoryOperationsAbstractSyntax  ABSTRACT-SYNTAX ::= {
    BasicDAP-PDUs
    IDENTIFIED BY      id-as-directoryOperationsAS }
BasicDAP-PDUs ::= TCAPMessage {{ DAP-Invokable }, { DAP-Returnable } }
DAP-Invokable      OPERATION      ::= {search | addEntry | removeEntry | modifyEntry}
DAP-Returnable     OPERATION      ::= {search | addEntry | removeEntry | modifyEntry}
inDirectoryBindingAbstractSyntax     ABSTRACT-SYNTAX ::= {
    DAPBinding-PDUs
    IDENTIFIED BY      id-as-directoryBindingAS }
DAPBinding-PDUs ::= CHOICE
{
    bind                Bind { directoryBind },
    unbind              Unbind { directoryUnbind }
}
END

SDFProtocolObjectIdentifiers
    {ccitt recommendation q 1218 module(0) sdfProtocolObjectIdentifiers(10) version1 (0)}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --

IMPORTS
-- useful definitions

in-ds                OBJECT IDENTIFIER ::=
    {ccitt recommendation q 1218 sdf-objects (10)}

id-rosObject         OBJECT IDENTIFIER ::= {in-ds 25}
id-contract          OBJECT IDENTIFIER ::= {in-ds 26}
id-package           OBJECT IDENTIFIER ::= {in-ds 27}
id-ac                OBJECT-IDENTIFIER ::= {in-ds 3}
id-as                OBJECT IDENTIFIER ::= {in-ds 5}

```

-- ROS Objects --

id-rosObject-dua	OBJECT IDENTIFIER	::=	{id-rosObject 1}
id-rosObject-directory	OBJECT IDENTIFIER	::=	{id-rosObject 2}
id-rosObject-dapDSA	OBJECT IDENTIFIER	::=	{id-rosObject 3}

-- contracts --

id-contract-dap	OBJECT IDENTIFIER	::=	{id-contract 1}
------------------------	--------------------------	------------	------------------------

-- packages --

id-package-search	OBJECT IDENTIFIER	::=	{id-package 2}
id-package-modify	OBJECT IDENTIFIER	::=	{id-package 3}
id-package-dapConnection	OBJECT IDENTIFIER	::=	{id-package 10}

-- application contexts --

id-ac-directoryAccessAC	OBJECT IDENTIFIER	::=	{id-ac 1}
--------------------------------	--------------------------	------------	------------------

-- abstract syntaxes --

id-as-directoryOperationsAS	OBJECT IDENTIFIER	::=	{id-as 1}
id-as-directoryBindingAS	OBJECT IDENTIFIER	::=	{id-as 2}

END

3 Semantics

3.1 Definition of procedures and entities

3.1.1 SSF application entity procedures

3.1.1.1 General

This clause provides the definition of the SSF Application Entity (AE) procedures related to the SSP-SCP interface. The procedures are based on the use of Common Channel Signalling System No. 7 (CCSS No. 7); other signalling systems can be used (e.g. DSS 1-layer 3).

Capabilities not explicitly covered by these procedures may be supported in an implementation dependent manner in the SSP, while remaining in line with clause 2.

The AE, following the architecture defined in the Recommendations Q.700, Q.771 and Q.1400, includes Transaction Capabilities Application Part (TCAP) and one or more ASEs called TC-users. The following subclauses define the TC-user ASE which interfaces with TCAP using the primitives specified in Recommendation Q.771; other signalling systems, such as DSS 1 (layer 3), may be used.

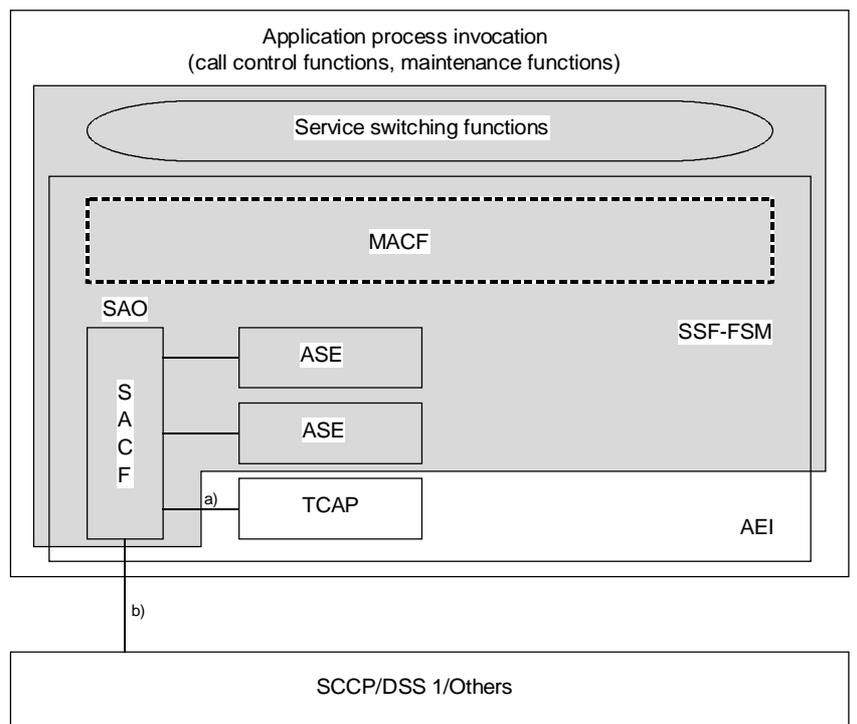
The procedure may equally be used with other signalling message transport systems supporting the application layer structures defined.

In case interpretations for the application entity procedures defined in the following differ from detailed procedures and the rules for using of TCAP service, the statements and rules contained in the detailed subclauses 3.3 and 3.4 shall be followed.

3.1.1.2 Model and interfaces

The functional model of the AE-SSF is shown in Figure 9; the ASEs interface to TCAP to communicate with the SCF, and interface to the call control function (CCF) and the maintenance functions already defined for switching systems. The scope of this Recommendation is limited to the shaded area in Figure 9.

The interfaces shown in Figure 9 use the TC-user ASE primitives specified in Recommendation Q.771 and N-Primitives specified in Recommendation Q.711. The operations and parameters of Intelligent Network Application Protocol (INAP) are defined in clause 2.



T1146720-92/d09

AEI Application Entity Invocation
 SSF Service Switching Functions
 FSM Finite State Machine
 MACF Multiple Association Control Function
 SACF Single Association Control Function
 SAO Single Association Object

a) TC-Primitives
 b) N-Primitives

NOTE – SSF-FSM includes several finite state machines.

FIGURE 9/Q.1218
Functional model of SSF-AE

3.1.1.3 Relations between SSF-FSM and the CCF and maintenance functions

The primitive interface between the SSF-FSM and the CCF/maintenance functions is an internal interface and is not subject to standardization in CS-1. Nevertheless this interface should be in line with the BCSM defined in 4.2.1.2/Q.1214.

The relationship between the BCSM and the SSF-FSM may be described as follows for the case of a call/attempt initiated by an end user, and the case of a call/attempt initiated by IN service logic:

- When a call/attempt is initiated by an end user and processed at an exchange, an instance of a BCSM is created. As the BCSM proceeds, it encounters detection points (DPs, see 4.2/Q.1214). If a DP is armed as a Trigger DP (TDP) an instance of an SSF-FSM is created.
- If an InitiateCallAttempt is received from the SCF, an instance of a BCSM is created, as well as an instance of an SSF-FSM.

The SSF logic should:

- perform the DP processing actions specified in 4.2.2.7/Q.1214, including if DP criteria are met;
- check if traffic mechanisms are active;
- check for SCF accessibility;
- handle service feature interactions.

The SSF hands control back to the CCF at least in the following cases:

- If call gapping is in effect – The SSF logic will instruct the CCF to terminate the call with the appropriate treatment.
- If service filtering is in effect – The call is counted (if required) and the SSF logic instructs the CCF to handle the call with the appropriate treatment.
- If a trigger (TDP) criteria match is not found (e.g. insufficient information to proceed) – The SSF logic returns call control to the CCF.
- If the call is abandoned – The SSF logic returns call control to the CCF and continues processing as described in 3.1.1.5.
- If the destination SCF is not accessible – The SSF logic will instruct the CCF to route the call if possible (e.g. default routing to a terminating announcement).
- If there is an existing control relationship for the call and the DP is armed as an EDP-R – The SSF returns call control to the CCF.

The management functions related to the execution of operations received from the SCF are executed by the SSF Management Entity (SSME). The SSME comprises a SSME-Control and several instances of SSME FSMs. The SSME-control interfaces the different SSF FSMs and SSME FSMs respectively and the Functional Entity Access Manager (FEAM). Figure 10 shows the SSF Interfaces.

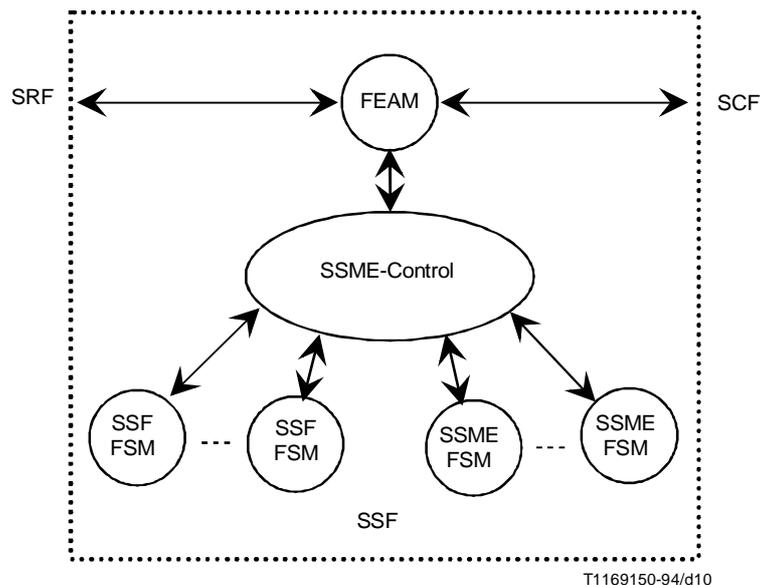


FIGURE 10/Q.1218
SSF interfaces

The Functional Entity Access Manager (FEAM) provides the low level interface maintenance functions including the following:

- 1) establishing and maintaining the interfaces to the SCF and SRF;
- 2) passing and queuing (when necessary) the messages received from the SCF and SRF to the SSME-Control;
- 3) formatting, queuing (when necessary), and sending the messages received from the SSME-Control to the SCF and SRF.

The SSME-control maintains the dialogues with the SCF and SRF on behalf of all instances of the SSF Finite State Machine (FSM). These instances of the SSF FSM occur concurrently and asynchronously as calls occur, which explains the need for a single entity that performs the task of creation, invocation and maintenance of the SSF-FSMs. In particular the SSME-control performs the following tasks:

- 1) Interprets the input messages from other FEs and translates them into corresponding SSF FSM events.
- 2) Translates the SSF-FSM outputs into corresponding messages to other FEs.
- 3) Captures asynchronous (with call processing) activities related to management or supervisory functions in the SSF and creates an instance of a SSME-FSM. For example, the SSME provides non-call associated treatment due to changes in Service Filtering or Call Gapping. Therefore, the SSME-control separates the SSF-FSM from the Call Gapping and Service Filtering functions by creating instances of SSME-FSMs for each context of management related operations.

The different contexts of the SSME-FSMs may be distinguished based on the address information provided in the initiating operations. In the case of service filtering this address information is given by filteringCriteria, i.e. all ActivateServiceFiltering operations using the same address, address the same SSME-FSM handling this specific service filtering instance. For example ActivateServiceFiltering operations providing different filtering Criteria cause the invocation of new SSME-FSMs.

The SSF-FSM passes call handling instructions to the related instances of the BCSM as needed. DPs may be dynamically armed as Event DPs, requiring the SSF-FSM to remain active. At some point, further interaction with the SCF is not needed, and the SSF-FSM may be terminated while the BCSM continues to handle the call as needed. A later TDP in the BCSM may result in a new instance of the SSF-FSM for the same call.

Consistent with the single-ended control characteristic of IN service features for CS-1, the SSF-FSM only applies to a functionally separate call portion (e.g. the originating BCSM or the terminating BCSM in a two-party call, but not both).

3.1.1.4 SSF Management Finite State Machine (SSME-FSM)

The SSME-FSM state diagram is described in Figure 11.

The SSME-FSM is independent of the individual SSF-FSMs.

The non-call associated treatment state is entered from the IdleManagement state when one of the following non-call associated operations is received (transition em1):

- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- ActivateServiceFiltering;
- CallGap;
- ActivityTest.

The CallGap operation may be received inside as well as outside a call context transaction. The ActivityTest operation applies to call context transactions only. The ActivateServiceFiltering operation may be received outside a call context only.

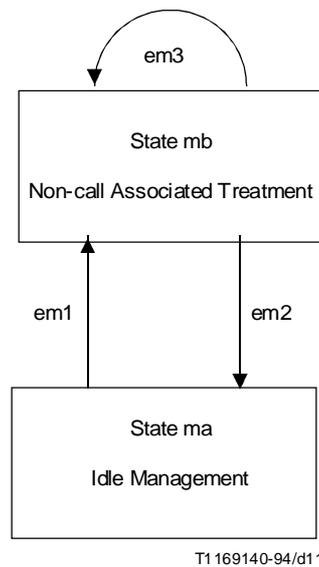


FIGURE 11/Q.1218
SSME-FSM state diagram

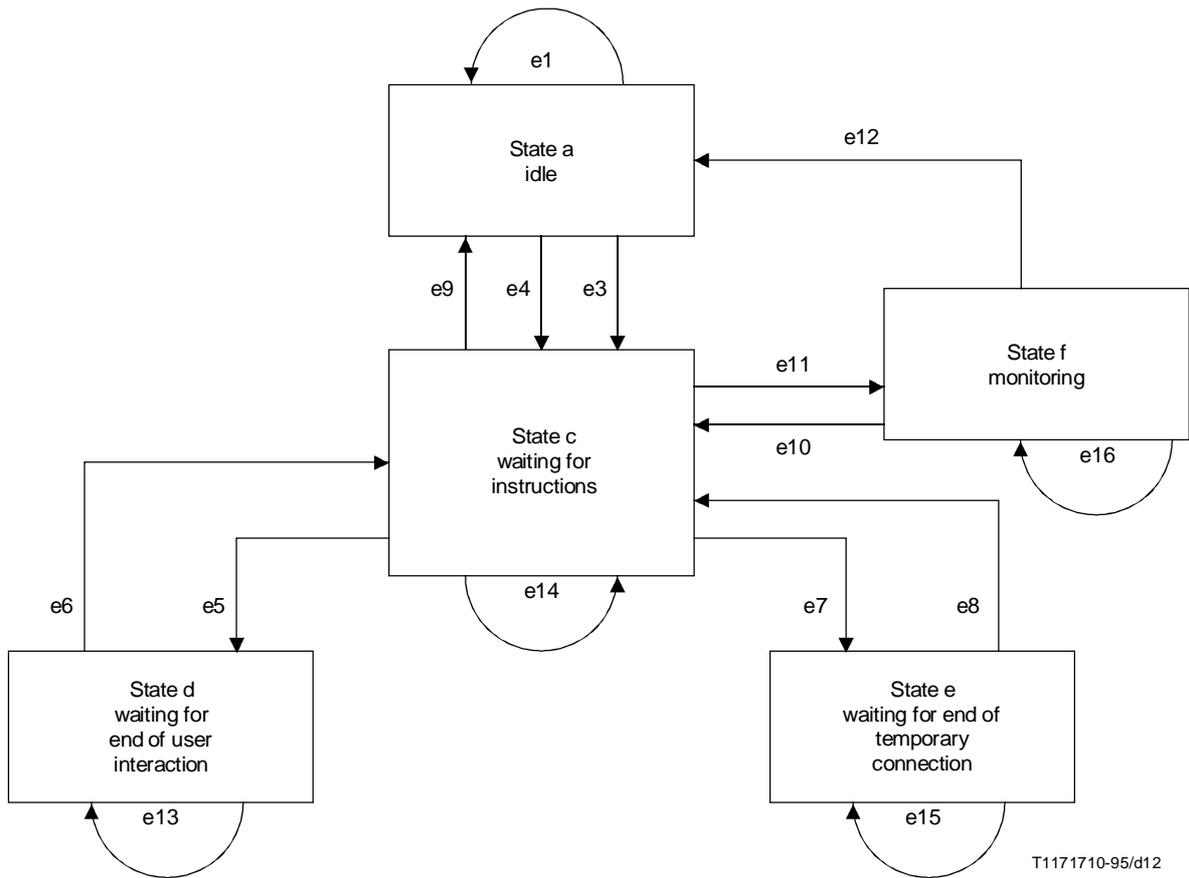
During this state the following events can occur:

- given that service filtering is active, the SSF needs to send a service filtering response to the SCF: the SSME-FSM remains in this state (transition em3);
- given that service filtering is active, the SSF needs to increment a counter: the SSME FSM remains in this state (transition em3);
- given that service filtering is active and the service filtering duration expires: the SSME should move to the Idle Management state (transition em2) and send a ServiceFilteringResponse operation to the SCF;
- given that status report is active previously requested by a RequestEveryStatusChangeReport operation, the SSF needs to send a StatusReport operation: the SSF remains in this state (transition em3);
- given that status report is active previously requested by a RequestFirstStatusMatchReport operation, the SSF needs to send a StatusReport operation: the SSF transits to Idle Management state (transition em2);
- given that status report is active previously requested by a RequestFirstStatusMatchReport or a RequestEveryStatusMatchReport operation, and the CancelStatusReportRequest operation is received by the SSF or the status report duration expires, the SSME-FSM should move to the Idle Management state (transition em2);
- given that status report is active previously requested by a RequestCurrentStatusReport operation, the SSF sends the StatusReport operation, the SSME-FSM should move to the Idle Management state (transition em2);
- if call gap related duration timer expires, the SSME-FSM should move to the Idle Management state (transition em2);
- given that call gap/service filtering is active, another CallGap/ActivateServiceFiltering operation could be received by the SSF, which has the same gapping/filtering criteria: the second “filter” or “gap” replace the first one (transition em3) unless the duration timer value is equal to zero, in which case the SSME-FSM should move to the Idle Management state (transition em2).

All other operations have no effect on the SSME-FSMs; the operations are passed by the SSME-Control to the relevant SSF-FSM.

3.1.1.5 SSF state transition diagram

Figure 12 shows the state diagram of the SSF part of the SSP during the processing of an IN call/attempt.



NOTE - The abandon and disconnect transition is not shown.

FIGURE 12/Q.1218
SSF finite state machine

Each state is discussed in the following subclauses. General rules applicable to more than one state are addressed here.

One or a sequence of components received in one or more TCAP messages may include a single operation or multiple operations, and is processed as follows:

- Process the operations in the order in which they are received.
- Each operation causes a state transition independent of whether or not a single operation or multiple operations are received in a message.

- The SSF examines subsequent operations in the sequence. As long as sequential execution of these operations would leave the FSM in the same state, it will execute them (e.g. RequestReportBCSMEvent). If a subsequent operation causes a transition out of the state, then the following operations should be buffered until the current operation has been executed. In all other cases, await an event that would cause a transition out of the current state (such an event would be the completion of operation being executed, or reception of an external event). An example of this is as follows:
 - The SSF receives the operations FurnishChargingInformation, ConnectToResource, and PlayAnnouncement in a component sequence inside a single TCAP message. Upon receipt of this message, these operations are executed up to and including ConnectToResource while the SSF is in the Waiting For Instructions state. As the ConnectToResource operation is executed (and when, or after the FurnishChargingInformation operation has been completed), the SSF-FSM will transit to the Waiting For End Of User Interaction state. The PlayAnnouncement operation is relayed to the SRF while the SSF is in Waiting For End Of User Interaction state.
- If there is an error in processing one of the operations in the sequence, the SSF-FSM processes the error (see below) and discards all remaining operations in the sequence.
- If an operation is not understood or is out of context (i.e. violates the SACF rules defined by the SSF-FSM) as described above, ABORT the interaction. For example, when the SSF-FSM applies to an originating BCSM, then receiving SelectFacility operation would be out of context since this applies only to the terminating half of the BCSM.

In any state, if there is an error in a received operation, the maintenance functions are informed and the SSF-FSM remains in the same state as when it received the erroneous operation; depending on the class of the operation, the error could be reported by the SSF to the SCF using the appropriate component (see Recommendation Q.774).

In any state (except idle), if the calling party abandons the call before it is answered (i.e. before the Active PIC in the BCSM), then the SSF-FSM should instruct the CCF to clear the call and ensure that any CCF resources allocated to the call have been de-allocated, then continue processing as follows:

- If the Abandon DP is not armed and there is no call information request pending, then transition to the idle state.
- If the Abandon DP is not armed and there is a CallInformationRequest pending, send a CallInformationReport, then transit to the Idle state.
- If the Abandon DP is armed as an EDP-R, send an EventReportBCSM or a DP specific operation, then transit to the Waiting For Instructions state. If a CallInformationRequest is pending, a CallInformationRequest shall be sent before the corresponding EventReportBCSM or a DP specific operation is sent.
- If the Abandon DP is armed as an EDP-N and there is no CallInformationRequest pending, send an EventReportBCSM or a DP specific operation, then transit to the Idle state.
- If the Abandon DP is armed as an EDP-N and there is a CallInformationRequest pending, send an EventReportBCSM or a DP specific operation, followed by a CallInformationReport, then transit to the Idle state.
- Other pending requests that should be treated in the same way as the CallInformationRequest in the above list is the ApplyCharging.

In any state (except Idle), if a call party disconnects from a stable call (i.e. from the Active PIC in the BCSM), then the SSF-FSM should process this event as follows:

- If the Disconnect DP is not armed for that specific leg and there is no CallInformationRequest pending, transit to the Idle state.
- If the Disconnect DP is not armed and there is a CallInformationRequest pending, send a CallInformationReport and transit to the Idle state.

- If the Disconnect DP is armed as an EDP-R for that specific leg, send an EventReportBCSM or a DP specific operation, then transit to the Waiting For Instructions state. If a Call Information Request is pending, a CallInformationReport shall be sent before the corresponding EventReportBCSM or a DP specific operation is sent.
- If the Disconnect DP is armed as an EDP-N and there is no CallInformationRequest pending, send an EventReportBCSM or a DP specific operation, then transit to the Idle state.
- If the Disconnect DP is armed as an EDP-N and there is a CallInformationRequest pending, send an EventReportBCSM or a DP specific operation and a CallInformationReport, then transit to the Idle state.
- Other pending requests that should be treated in the same way as the CallInformationRequest in the above list is the ApplyCharging.

In any state (except Idle), an EventNotificationCharging can be sent to the SCF, if previously requested by a RequestNotificationCharging and if the charging event has been detected by the CCF. In this case no state transition takes place.

The SSF has an application timer, T_{SSF} , whose purpose is to prevent excessive call suspension time and to guard the association between the SSF and the SCF.

Timer T_{SSF} is set in the following cases:

- When the SSF sends an InitialDP, DP specific (refer to 3.1.1.5.2 State c: Waiting For Instructions), or AssistRequestInstructions operation (refer to 3.1.1.6.2 State b: Waiting For Instructions in assisting/handoff case). While waiting for the first response from the SCF, the timer T_{SSF} can be reset only once by a ResetTimer operation. Subsequent to the first response, the timer can be reset any number of times.
- When the SSF enters the Waiting For Instructions state (refer to 3.1.1.5.2) under any other condition than the ones listed in the previous case. In this case the SCF may reset the T_{SSF} timer using the ResetTimer operation any number of times.
- When the SSF receives a HoldCallInNetwork operation (refer to 3.1.1.5.2 State c: Waiting For Instructions). In this case the SCF may reset T_{SSF} using the ResetTimer operation any number of times.
- When the SSF enters the Waiting For End of User Interaction state or the Waiting For End of Temporary Connection state (refer to 3.1.1.5.3 and 3.1.1.5.4). In these cases the SCF may reset T_{SSF} using the ResetTimer operation any number of times.(OPTIONAL).

NOTE – This "OPTIONAL" means that the application timer T_{SSF} is optionally set. Whether it is used or not depends on implementation. But it must be synchronized with $T_{SCF-SSF}$ in the SCSM.

In the four above cases T_{SSF} may respectively have four different values as defined by the application.

When receiving or sending any operation which is different from the above, the SSF should reset T_{SSF} to the last used set value. This value is either one associated to the four different cases as listed above, or received in a ResetTimer operation, whatever occurred last. In the monitoring state (refer to 3.1.1.5.5) T_{SSF} is not used.

On expiration of T_{SSF} the SSF-FSM transitions to the idle state, aborts the interaction with the SCF and the CCF progresses the BCSM if possible.

The SSF state diagram contains the following transitions (events):

- | | |
|----|---------------------------------------|
| e1 | TDP-N encountered |
| e3 | InitiateCallAttempt received |
| e4 | TDP-R encountered |
| e5 | User interaction requested |
| e6 | User interaction ended |
| e7 | Temporary connection created |
| e8 | Temporary connection ended |
| e9 | Idle return from wait for instruction |

- e10 EDP-R encountered
- e11 Routing instruction received
- e12 EDP-N last encountered or ReleaseCall received or Cancel(allRequests) received
- e13 Waiting for End of User Interaction state no change
- e14 Waiting for Instructions state no change
- e15 Waiting for End of Temporary Connection state no change
- e16 Monitoring state no change
- e17 Abandon (from any state) (not shown in the SSF state diagram)
- e18 Disconnect (from any state) (not shown in the SSF state diagram)
- e19 Non-Call Associated Treatment from any state (not shown in the SSF state diagram)

The SSF state diagram contains the following states:

- State a Idle
- State c Waiting for Instructions
- State d Waiting for End of User Interaction
- State e Waiting for End of Temporary Connection
- State f Monitoring

3.1.1.5.1 State a – Idle

The SSF-FSM enters the Idle state under a variety of conditions, as described below.

The SSF-FSM enters the Idle state when sending or receiving an ABORT TCAP primitive due to abnormal conditions in any state.

The SSF-FSM enters the Idle state when one of the following occurs:

- when the call is abandoned or one or more call parties disconnect in any other state under the conditions identified in 3.1.1.5;
- when a Connect or proceed call processing operation is processed in the Waiting for Instructions state, and no EDPs are armed and there are no outstanding report requests (transition e9);
- when the application timer T_{SSF} expires in one of the states: Waiting for Instructions, Waiting for End of User Interaction or Waiting for End of Temporary Connection;
- when a ReleaseCall operation is processed in Waiting for Instructions (transition e9) or Monitoring (transition e12);
- when the last EDP-N is encountered in the Monitoring state, and there are no EDP-Rs armed and no monitoring is active for charging events (transition e12);
- when the last charging event is encountered in the Monitoring state, and there are no EDPs armed (transition e12).

When transiting to the Idle state, if there is a CallInformationRequest pending (see 3.1.1.5), the SSF sends a CallInformationReport operation to the SCF before returning to Idle. Once in the Idle state, if status reporting is still active the SSF deactivates it, any outstanding responses to send to the SCF are discarded.

During this state the following call-associated events can occur:

- indication from the CCF that an armed TDP is encountered related to a possible IN call/service attempt, the SSF-FSM acts as described below:
 - if the DP is a TDP-N, send a generic InitialDetectionPoint or DP-specific operation (see Note) to the SCF, as determined from DP processing; there is no resulting transition to a different state (transition e1);

- if the DP is a TDP-R, send a generic InitialDetectionPoint or DP-specific operation (see Note) to the SCF, as determined from DP processing, and transit to the Waiting for Instructions state (transition e4);
 - the rules for DP processing are described in Recommendation Q.1214, section "DP Processing";
- a message related to a new transaction containing an InitiateCallAttempt operation is received from the SCF: in this case the SSF-FSM moves to the state Waiting for Instructions (transition e3).

Any other operation received from the SCF while the SSF is in Idle state should be treated as an error. The event should be reported to the maintenance functions and the transaction should be aborted according to the procedure specified in TCAP (see Recommendation Q.774).

NOTE – DP specific operations are the following (refer to clause 2): TAnswer, TDisconnect, TermAttemptAuthorized, TMidCall, TNoAnswer, AnalysedInformation, TBusy, CollectedInformation, OAnswer, OCalledPartyBusy, ODisconnect, OMidCall, ONoAnswer, OriginationAttemptAuthorized, RouteSelectFailure.

3.1.1.5.2 State c – Waiting for Instructions

This state is entered from the Idle state, as indicated above (transition e4), directly from the Idle state on receipt at the SSF of a TC-BEGIN indication primitive containing an InitiateCallAttempt operation from the SCF (transition e3), from the state Monitoring on detection of an EDP-R (transition e10), from the state Waiting for End of User Interaction on occurrence of call disconnect from/to the SRF (transition e6), or from the state Waiting for End of Temporary Connection on occurrence of disconnection of temporary connection (transition e8).

In this state the SSF-FSM is waiting for an instruction from the SCF; call handling is suspended and an application timer (T_{SSF}) should be set on entering this state.

During this state the following events can occur:

- The user dials additional digits (applies for open-ended numbering plans) – The CCF should store the additional digits dialled by the user.
- The user abandons or disconnects. This should be processed in accordance with the general rules in 3.1.1.5.
- The application Timer T_{SSF} expires – The SSF-FSM moves to the Idle state, the CCF routes the call if possible (e.g. default routing to a terminating announcement), the T_{SSF} expiration is reported to the maintenance functions and the transaction is aborted.
- An operation is received from the SCF – The SSF-FSM acts according to the operation received as described below.

The following operations may be received from the SCF and processed by the SSF with no resulting transition to a different state (transition e14):

- HoldCallInNetwork;
- RequestReportBCSMEEvent;
- RequestNotificationChargingEvent;
- ResetTimer;
- FurnishChargingInformation;
- ApplyCharging;
- CallInformationRequest;
- SendChargingInformation;
- Cancel.

The following operations may be received from the SCF and processed by the SSF, causing a state transition to Waiting for End of User Interaction state (transition e5):

- ConnectToResource.

The following operations may be received from the SCF and processed by the SSF, causing a state transition to Waiting for End of Temporary Connection state (transition e7):

- EstablishTemporaryConnection

The following operations may be received from the SCF and processed by the SSF, causing a state transition to either Monitoring state (if any EDPs were armed or any reports were requested) (transition e11) or Idle state (transition e9):

- Connect;
- CollectInformation;
- AnalyseInformation;
- SelectRoute;
- SelectFacility;
- Continue;
- ReleaseCall.

ReleaseCall operation may be received from the SCF. In this case, the SSF-FSM should instruct the CCF to clear the call and ensure that any CCF resources allocated to the call have been de-allocated, then continue processing as follows:

- if neither CallInformationRequest nor ApplyChargingReport operation has been requested, the SSF-FSM transits to the Idle state (transition e9);
- if CallInformationRequest or ApplyChargingReport operation has been requested, the SSF sends each operation which has been requested from SCF, and then the SSF-FSM transits to the Idle state (transition e9).

When processing the above operations, any necessary call handling information is provided to the Call Control Function (CCF).

Any other operation received in this state should be processed in accordance with the general rules in 3.1.1.5.

3.1.1.5.3 State d – Waiting for End of User Interaction

The SSF enters this state from the Waiting for Instructions state (transition e5) on the reception of one of the following operations:

- ConnectToResource.

During this state the following events can occur:

- A valid SCF-SRF operation [i.e. PlayAnnouncement, PromptAndCollectUserInformation, and Cancel(invokedID)] for relaying is received and is correct, the operation is transferred to the SRF for execution. The SSF-FSM remains in the Waiting for End of User Interaction state (transition e13).
- A valid SRF-SCF operation (i.e. SpecializedResourceReport and ReturnResult from PromptAndCollectUserInformation) for relaying is received and is correct, the operation is transferred to the SCF. The SSF-FSM remains in the Waiting for End of User Interaction state (transition e13).
- The application timer T_{SSF} expires (if it was set) – The SSF-FSM moves to the Idle state, the CCF routes the call if possible (e.g. default routing to a terminating announcement), the T_{SSF} expiration is reported to the maintenance functions and the transaction is aborted.
- An operation is received from the SCF – The SSF-FSM acts according to the operation received as described below.
- The user abandons – This should be processed in accordance with the general rules in 3.1.1.5.

The following operations may be received from the SCF and processed by the SSF with no resulting transition to a different state (transition e13):

- RequestNotificationChargingEvent;
- ResetTimer;
- FurnishChargingInformation;
- ApplyCharging;
- Send Charging Information.

L'opération DisconnectForwardConnection peut être reçue en provenance de l'entité SCF et traitée par l'entité SSF dans cet état. La déconnexion de l'appel peut aussi provenir de l'entité SRF. Dans les deux cas, elle entraîne la libération de la connexion vers l'entité SRF et une transition à l'état Waiting for Instructions. La déconnexion n'est pas transmise à l'autre partie (transition e6).

Toute autre opération reçue dans ce état doit être traitée selon les règles générales énoncées en 3.1.1.5.

3.1.1.5.4 Etat e – "Waiting for End of Temporary Connection"

L'entité SSF passe à cet état à partir de l'état Waiting for Instructions (transition e7) lorsqu'elle reçoit une opération de type EstablishTemporaryConnection.

L'appel est acheminé vers l'entité SSF/SRF assistante et son traitement est suspendu en attendant la fin de la procédure d'assistance. Le temporisateur T_{SSF} est actif dans cet état (qu'il soit utilisé ou pas est facultatif).

Pendant cet état, les événements suivants peuvent se produire:

- expiration de la temporisation d'application T_{SSF} (si elle a été fixée) – La machine FSM de l'entité SSF passe à l'état Idle, l'entité CCF achemine l'appel si possible (acheminement par défaut vers une annonce terminale, par exemple), la fin de la temporisation T_{SSF} est signalée aux fonctions de maintenance et la transaction est abandonnée;
- réception d'une indication de déconnexion de la connexion vers l'avant provenant de l'entité CCF – Dans ce cas, l'entité SSF passe à l'état Waiting for Instructions (transition e8). La déconnexion n'est pas transférée au demandeur;
- l'utilisateur abandonne l'appel et cette opération doit être traitée selon les règles générales énoncées en 3.1.1.5;
- une opération est reçue en provenance de l'entité SCF – l'entité SSF agit selon l'opération reçue comme cela est décrit ci-dessous.

Les opérations suivantes peuvent être reçues de l'entité SCF et traitées par l'entité SSF sans entraîner de transition d'état (transition e15):

- RequestNotificationChargingEvent;
- ResetTimer;
- FurnishChargingInformation;
- ApplyCharging;
- SendChargingInformation.

L'opération DisconnectForwardConnection peut être reçue de l'entité SCF et traitée par l'entité SSF dans cet état. La déconnexion de l'appel peut aussi provenir de l'entité SRF. Dans ces deux cas, elle entraîne la libération de la connexion vers l'entité SRF et la transition à l'état Waiting for Instructions. La déconnexion n'est pas transférée à l'autre partie (transition e8).

Toute autre opération reçue dans cet état doit être traitée selon les règles générales énoncées en 3.1.1.5.

3.1.1.5.5 Etat f – "Monitoring"

L'entité SSF parvient à cet état à partir de l'état "Waiting For Instructions" (transition e11) lorsqu'elle reçoit une opération de type Connect, CollectInformation, AnalyseInformation, SelectRoute, SelectFacility, Continue ou lorsqu'elle traite une opération de type CallAttempt quand un ou plusieurs EDP sont armés et/ou il y a d'autres rapports en attente (voir 3.1.1.5).

Dans cet état, le temporisateur T_{SSF} n'est pas utilisé, c'est-à-dire que la fin de cette temporisation n'a aucune influence sur la machine FSM de l'entité SSF.

Pendant cet état, les événements suivants peuvent se produire:

- une notification de point EDP-N doit être signalée à l'entité SCF en envoyant une opération de type EventReportBCSM ou une opération spécifique au point DP; la machine FSM de l'entité SSF doit rester dans l'état Monitoring (transition e16) si un ou plusieurs points EDP sont armés ou s'il y a des demandes de comptes rendus en attente. La machine FSM de l'entité SSF doit passer à l'état Idle (transition e12) s'il ne reste pas de point EDP armé ou s'il n'y a pas de demandes de comptes rendus en attente.

Si l'événement provoquant un CallInformationReport est également détecté par un point EDP-N armé, l'opération CallInformationReport sera envoyée immédiatement avant le EventReportBCSM correspondant ou une opération spécifique au point DP sera envoyée;

- une demande de point EDP-R doit être signalée à l'entité SCF en envoyant une opération EventReportBCSM ou une opération spécifique au point DP; la machine FSM de l'entité SSF devant passer à l'état "Waiting For Instructions" (transition e10);
- si l'événement provoquant un CallInformationReport est également détecté par un point EDP-R armé, le CallInformationReport sera envoyé immédiatement après le EventReportBCSM correspondant ou une opération spécifique au point DP est envoyée;
- la réception d'une primitive END (fin) ou ABORT (abandon) provenant du sous-système TCAP n'a pas d'effet sur l'appel; l'appel pouvant continuer à l'aide des informations disponibles. Dans ce cas, la machine FSM de l'entité SSF passe à l'état Idle (transition e12), se dissociant de l'appel;
- réception d'une opération provenant de l'entité SCF: la machine FSM de l'entité SSF se comporte conformément à cette opération, comme décrit précédemment;
- l'utilisateur abandonne l'appel ou se déconnecte. Cela devrait être traité conformément aux règles générales énoncées en 3.1.1.5.

Les opérations ci-après peuvent être reçues de l'entité SCF et traitées par l'entité SSF sans entraîner de transition d'état (transition e16):

- RequestNotificationChargingEvent;
- SendChargingInformation;
- FurnishChargingInformation;
- ApplyCharging.

L'opération RequestReportBCSMEvent peut provenir de l'entité SCF. Dans ce cas, la machine FSM de l'entité SSF transite comme suit:

- si un ou plusieurs points EDP sont armés ou si les points EDP sont en partie désarmés, la machine FSM de l'entité SSF revient au même état (transition e16);
- si tous les points EDP qui ont été armés sont désarmés et s'il y a des demandes de type CallInformationRequest, EventNotificationCharging ou ApplyChargingReport en cours, la machine FSM de l'entité SSF revient au même état (transition e16);
- si tous les points EDP armés sont désarmés et s'il y a des demandes de type CallInformationRequest, EventNotificationCharging ou ApplyChargingReport en cours, la machine FSM de l'entité SSF passe à l'état Idle (transition e12).

L'opération Cancel peut provenir de l'entité SCF et être traitée par l'entité SSF, entraînant une transition d'état vers l'état Idle (transition e12).

L'opération ReleaseCall peut provenir de l'entité SCF. Dans ce cas, la machine FSM de l'entité SSF doit donner l'ordre à l'unité CCF de libérer l'appel et de s'assurer que toutes les ressources CCF affectées à l'appel ont été libérées, puis poursuivre le traitement comme suit:

- si aucune des opérations CallInformationRequest ou ApplyChargingReport n'a été demandée, la machine FSM de l'entité SSF passe à l'état Idle (transition e12);
- si l'opération CallInformationRequest ou ApplyChargingReport a été demandée, l'entité SSF envoie chaque opération qui a été demandée par l'entité SCF, et puis la machine FSM de l'entité SSF passe à l'état Idle (transition e12).

Toute autre opération reçue dans cet état doit être traitée conformément aux règles générales énoncées en 3.1.1.5.

3.1.1.6 Machine FSM de transfert de tâche à une entité SSF assistante

Le présent paragraphe décrit la machine FSM d'une entité SSF assistante. Une entité SSF assistante est structurée comme décrit en 3.1.1.1 à 3.1.1.5. Dans l'ensemble de capacités CS-1, la machine FSM de transfert de tâche ne s'utilise que lorsqu'on applique le traitement final de l'appel.

A l'intérieur de ce paragraphe, le terme "SSF assistante" fait à la fois référence aux entités SSF assistante et de transfert, à moins que l'une ou l'autre de ces entités soit explicitement indiquée.

Le diagramme d'états de l'entité SSF assistante/de transfert de tâches comporte les transitions (événements) suivantes (voir la Figure 13):

- ea1 Détection de transfert de tâche/d'assistance
- ea2 Echech/aboutissement de transfert de tâche/d'assistance
- ea3 Interaction utilisateur demandée
- ea4 Interaction utilisateur terminée
- ea5 Transition nulle de l'état Waiting For Instructions
- ea6 Transition nulle de l'état Waiting For End of User Interaction
- ea7 Début de déconnexion de canal support amont

Le diagramme d'états de l'entité SSF assistante/de transfert de tâches comporte les états suivants:

- Etat a Idle (repos)
- Etat b Waiting for Instructions (attente d'instructions)
- Etat c Waiting for End of User Interaction (attente de fin d'interaction utilisateur)

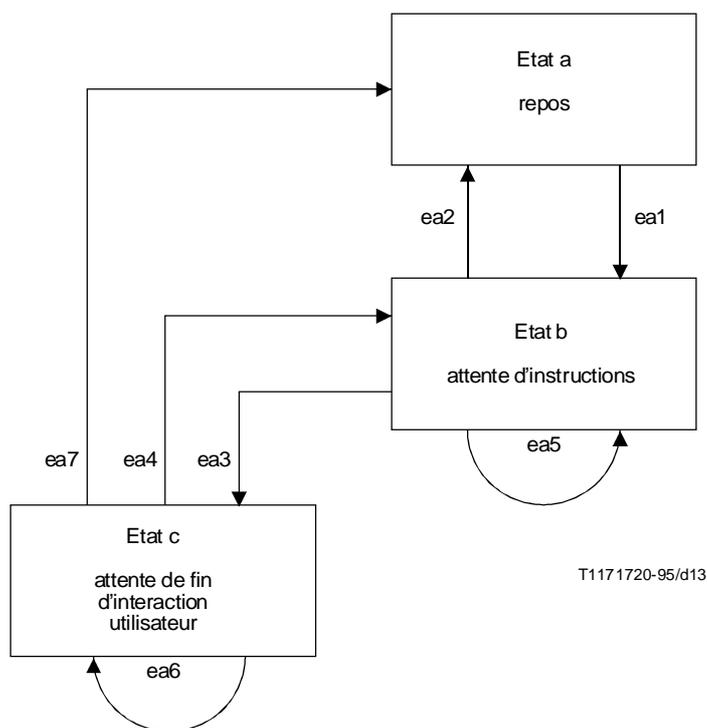


FIGURE 13/Q.1218

Machine à états finis de transfert de tâche à une entité SSF assistante

3.1.1.6.1 Etat a – "Idle" (repos)

La machine FSM de l'entité SSF passe à l'état Idle lorsque l'un des événements suivants intervient:

- réception ou émission d'une primitive ABORT TCAP (abandon du sous-système TCAP) due à des anomalies constatées dans un état quelconque;
- lorsqu'une connexion temporaire existe entre une entité SSF amont et l'entité SSF assistante en aval, lorsqu'une déconnexion de canal support est reçue à partir de l'entité SSF amont; (transition ea2).

Lorsque la machine FSM de l'entité SSF est à l'état Idle, les éventuelles réponses en attente à envoyer à l'entité SCF sont rejetées par l'entité SSF assistante.

La machine FSM de l'entité SSF assistante passe de l'état Idle à l'état Waiting for Instructions lorsque l'entité SSF assistante reçoit une indication d'assistance provenant d'une autre entité SSF (transition ea1).

Toute autre opération reçue de l'entité SCF pendant que l'entité SSF assistante est dans l'état Idle doit être traitée comme une erreur. Cet événement doit être signalé aux fonctions de maintenance et la transaction abandonnée selon la procédure spécifiée dans le sous-système TCAP (voir Recommandation Q.774).

3.1.1.6.2 Etat b – "Waiting for Instructions" (attente d'instructions)

Une entité SSF accède à cet état à partir de l'état Idle lorsqu'elle reçoit une connexion provenant d'une autre entité SSF demandant une assistance, selon un mécanisme de détection dépendant de la réalisation (transition ea1).

Avant d'accéder à cet état, l'entité SSF envoie une opération de type AssistRequestInstructions à l'entité SCF. Dans cet état, la machine FSM de l'entité SSF assistante attend une instruction de l'entité SCF; le traitement de l'appel est suspendu et un temporisateur d'application (T_{SSF}) doit être initialisé au moment du passage à cet état.

Pendant cet état, les événements suivants peuvent se produire:

- le temporisateur d'application T_{SSF} arrive à expiration – La machine FSM de l'entité SSF assistante passe à l'état Idle (transition ea2), la fin de la temporisation est signalée aux fonctions de maintenance et la transaction est abandonnée;
- une opération est reçue en provenance de l'entité SCF – La machine FSM de l'entité SSF agit selon l'opération reçue, comme décrit ci-dessus;
- une déconnexion de canal support est reçue de l'entité SSF amont et la machine FSM passe à l'état Idle (repos) (transition ea2).

Les opérations suivantes peuvent être reçues de l'entité SCF et traitées par l'entité SSF assistante sans entraîner de transition d'état (transition ea5):

- ResetTimer;
- FurnishChargingInformation;
- ApplyCharging;
- SendChargingInformation.

Les opérations suivantes peuvent être reçues de l'entité SCF et traitées par l'entité SSF assistante en entraînant une transition à l'état Waiting for End of User Interaction (transition ea3):

- ConnectToResource.

Au cas où l'entité SSF est en transfert de tâche, l'opération ReleaseCall peut provenir de l'entité SCF. La machine FSM de l'entité SSF devrait donner l'ordre à l'entité CCF de libérer l'appel et de s'assurer que toutes les ressources CCF affectées à l'appel ont été libérées, puis continuer le traitement de la façon suivante:

- si l'opération ApplyChargingReport n'est pas demandée, la machine FSM de l'entité SSF effectuant le transfert de tâche transite vers l'état Idle (repos) (transition ea2);
- si l'opération ApplyChargingReport a été demandée, l'entité SSF effectuant le transfert de tâche envoie ApplyChargingReport à l'entité SCF, puis la machine FSM de l'entité SSF effectuant le transfert de tâche transite vers l'état Idle (repos) (transition ea2).

Noter que l'opération ci-dessus n'est permise que dans l'entité SSF effectuant le transfert de tâche. Au cas où un équipement ne permettrait pas de faire la distinction entre une entité SSF effectuant un transfert de tâche et une entité SSF assistante, il pourrait effectuer l'opération ReleaseCall dans l'entité SSF assistante.

Toute autre opération reçue dans cet état doit être traitée selon les règles générales énoncées en 3.1.1.5.

Il est à noter que la présente Recommandation ne traite pas des procédures multiples de transfert de tâche.

3.1.1.6.3 Etat c – "Waiting for End of User Interaction"

L'entité SSF assistante passe à cet état à partir de l'état Waiting for Instructions (transition ea3) lorsqu'elle reçoit l'opération suivante:

- ConnectToResource.

Pendant cet état, les événements suivants peuvent se produire:

- réception d'une opération SCF-SRF de retransmission [c'est-à-dire d'une opération de type Play announcement, prompt and collect user information et cancel (announcement)] valide et correcte: cette opération est transférée à l'entité SRF pour exécution. La machine FSM de l'entité SSF reste dans l'état Waiting for End of User Interaction (transition ea6);
- quand l'entité SRF indique à l'entité SSF assistante la fin de l'interaction avec l'utilisateur en lançant une déconnexion, la machine FSM de l'entité SSF assistante revient à l'état "Attente des instructions" (uniquement en cas de transfert de tâche) (transition ea4);
- expiration de la temporisation d'application T_{SSF} – La machine FSM de l'entité SSF passe à l'état Idle, l'entité CCF achemine l'appel si possible (acheminement par défaut vers une annonce terminale, par exemple), la fin de la temporisation T_{SSF} est signalée aux fonctions de maintenance et la transaction est abandonnée;
- réception d'une opération provenant de l'entité SCF – La machine FSM de l'entité SSF se comporte selon l'opération reçue, comme décrit ci-dessous;
- une déconnexion de canal support provient de l'entité SSF à l'origine de la demande – La machine FSM de l'entité SSF passe à l'état Idle (repos), la connexion avec l'entité SRF est libérée et il est mis fin à la transaction (transition ea7).

Les opérations suivantes peuvent être reçues en provenance de l'entité SCF et traitées par l'entité SSF sans entraîner de transition d'état (transition ea6):

- ResetTimer.

L'opération DisconnectForwardConnection peut être reçue en provenance de l'entité SCF et traitée par l'entité SSF assistante et de transfert de tâche dans cet état, entraînant une transition à l'état Waiting for Instructions (Attente des instructions) (transition ea4). Cette procédure n'est valide que si une opération de type ConnectToResource a été traitée auparavant en vue de provoquer un passage à l'état Waiting for End of User Interaction (Attente de la fin de l'interaction avec l'utilisateur).

Toute autre opération reçue dans cet état doit être traitée selon les règles générales énoncées en 3.1.1.5.

3.1.2 Procédures relatives à l'entité application de l'entité SCF

3.1.2.1 Généralités

Ce paragraphe fournit la définition des procédures relatives à l'entité d'application (AE) (*application entity*) concernant les interfaces SCF-SSF/SRF/SDF. Ces procédures font référence au système de signalisation n° 7 (SS n° 7), mais on peut aussi utiliser d'autres systèmes de signalisation.

En outre, d'autres capacités peuvent être prises en charge selon la réalisation par les entités physiques SCP (*service control point*), AD (*adjunct*) ou SN (*service node*).

L'entité application, conforme à l'architecture définie dans les Recommandations Q.700, Q.771 et Q.1400, comprend le sous-système application des capacités de transaction (TCAP) (*transaction capabilities application part*) et un ou plusieurs éléments du service application (ASE) de l'utilisateur du gestionnaire de transactions. Les paragraphes ci-après définissent les règles applicables à l'utilisateur TC et SACF & MACF, qui assurent l'interface avec le sous-système TCAP au moyen de primitives spécifiées dans la Recommandation Q.771.

La procédure peut également être utilisée avec d'autres systèmes de signalisation prenant en charge les structures définies de la couche application. Le présent texte ne vise pas à imposer de contraintes aux programmes de logique de services (SLP).

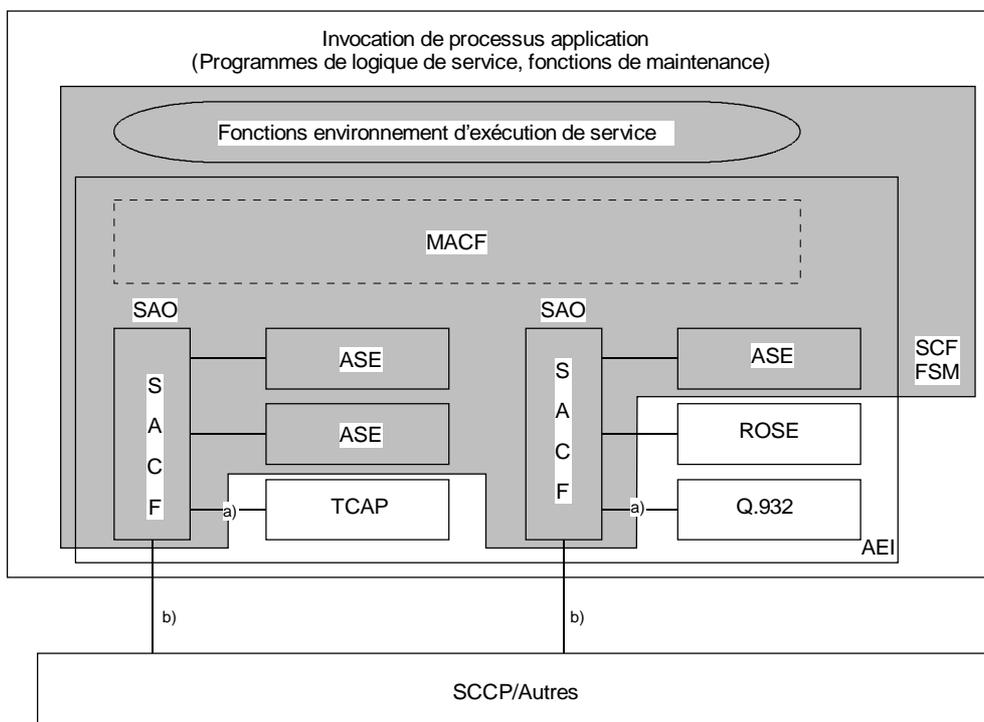
Au cas où les interprétations relatives aux procédures d'entité d'application et définies ci-après seraient différentes des procédures détaillées et des règles d'utilisation du service TCAP, il conviendrait de se conformer aux exposés et aux règles contenus en 3.3 et 3.4.

3.1.2.2 Modèle et interfaces

Le modèle fonctionnel de l'entité AE-SCF est représenté à la Figure 14; les éléments ASE assurent, d'une part, l'interface avec les couches de protocole de prise en charge nécessaires à la communication avec les entités SSF, SRF et SDF et, d'autre part, l'interface avec les programmes de logique de service et les fonctions de maintenance. Le domaine d'application de la présente Recommandation est limité à la zone ombrée de la Figure 14.

NOTE – La machine SCF-FSM comprend plusieurs machines FSM.

Les interfaces indiquées à la Figure 14 font appel aux primitives de l'élément ASE de l'utilisateur du gestionnaire de transactions spécifiées dans la Recommandation Q.771 [interface (1)] et aux primitives N spécifiées dans la Recommandation Q.711 [interface (2)]. Le fonctionnement et les paramètres associés au protocole d'application du réseau intelligent (protocole INAP) sont définis à l'article 2.



T1146770-92/d14

AEI Invocation de l'entité application (*application entity invocation*)
 SCF Fonction de commande de service (*service control function*)
 FSM Machine à états finis (*finite state machine*)
 MACF Fonction de commande d'association multiple (*multiple association control function*)
 SACF Fonction de commande d'association unique (*single association control function*)
 SAO Objet d'association unique (*single association object*)

a) Primitives TC ou primitives Q.932.

b) Primitives N.

NOTE – La machine SCF-FSM comporte plusieurs machines FSM.

FIGURE 14/Q.1218

Modèle fonctionnel d'entité application de la fonction de commande de service (SCF AE)

3.1.2.3 Relation entre la machine SCF-FSM et les fonctions de maintenance/programmes de logique de service

L'interface des primitives entre le SCF-FSM et les fonctions de maintenance des programmes de logique de service est une interface interne et ne fait pas l'objet de normalisation pour l'ensemble de capacités CS-1.

La relation entre le programme de logique de service et la machine SCF-FSM peut être décrite comme suit (dans le cas d'un appel déclenché par un utilisateur final et d'un appel déclenché par la logique de service du RI):

- lorsqu'une demande de traitement d'appel RI arrive en provenance de l'entité SSF, une instance de modèle d'état SCF (SCSM) (*SCF call state model*) est créée, et le programme de service de logique est appelé;
- lorsque le déclenchement d'un appel est demandé par la logique de service, une instance du modèle SCSM est créée.

Dans chaque cas, la machine FSM de l'entité SCF gère l'interaction avec la machine SSF-FSM (et la machine SRF-FSM et SDF-FSM) selon le cas, et signale les événements selon les besoins au programme de logique de service.

Les fonctions de gestion associées à l'exécution des opérations reçues en provenance de l'entité SCF sont prises en charge par l'entité de gestion de l'entité SCF (SCME) (*SCF management entity*). L'entité SCME comprend la commande SCME et différentes instances de machines SCME-FSM. L'entité SCME assure l'interface avec différents modèles d'état d'appel SCF (SCSM) et le gestionnaire d'accès d'entité fonctionnelle (FEAM) (*functional entity access manager*). La structure de la machine SCF-FSM est représentée à la Figure 15.

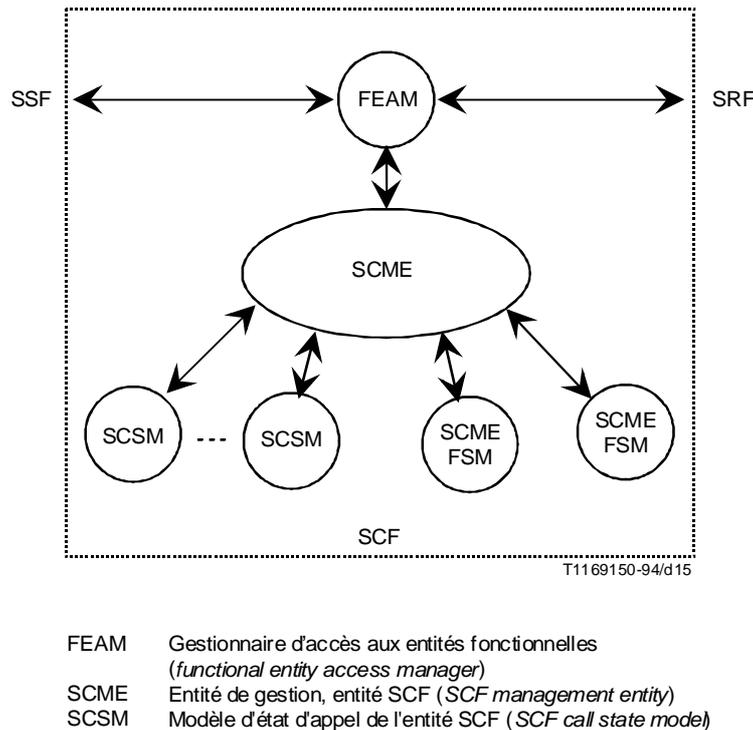


FIGURE 15/Q.1218
Structure de la machine FSM SCF

Le texte suivant décrit systématiquement les différents aspects des procédures relatives à l'interface entre l'entité SCF et d'autres entités fonctionnelles, le principal objectif étant de spécifier l'ordre des opérations plutôt que les capacités fonctionnelles des entités. En conséquence, la présente Recommandation décrit uniquement un sous-ensemble des capacités fonctionnelles de l'entité SCF.

Le modèle de procédure associe un modèle SCSM à chaque interrogation de l'entité SSF. Le modèle SCSM maintient les dialogues avec les entités SSF, SRF et SDF dans l'intérêt de la logique de service.

Plusieurs demandes peuvent être exécutées de manière parallèle et asynchrone par l'entité SCF, ce qui explique la nécessité de disposer d'une seule entité qui assure les tâches de création, d'appel et de maintenance des objets SCF-FSM. Cette entité est appelée entité de gestion SCF (Commande SCME) (*SCF management entity-control*). Outre les tâches mentionnées ci-dessus, le modèle SCME maintient les dialogues avec les SSF, SDF et SRF au nom de toutes les instances des FSM. En particulier, la commande du modèle SCME:

- 1) interprète les messages d'entrée provenant des autres entités fonctionnelles et les traduit en événements de modèle SCSM correspondants;
- 2) traduit les sorties du modèle SCSM en messages correspondants pour les autres entités fonctionnelles;

- 3) exécute quelques tâches asynchrones (avec le traitement d'appel) (une telle activité sera le contrôle de flux). Le modèle SCME a la responsabilité de déceler les surcharges nodales et d'envoyer l'indication de surcharge (par exemple, séquençement automatique des appels) à l'entité SSF pour exercer le contrôle de flux sur les demandes. Les autres activités sont les traitements des appels liés aux modifications de filtrage du service, de séquençement des appels et de l'état de la surveillance des ressources, de même que la fourniture de messages relatifs à l'état des ressources à l'entité SSF;
- 4) prend en charge les interactions durables entre la SCF et les autres entités fonctionnelles;
- 5) exécute toutes les tâches asynchrones (avec le traitement de appels) liées aux activités de gestion et de surveillance dans l'entité SCF et crée une instance de machine SCME-FSM. Par exemple, l'entité SCME fournit le traitement indépendant des appels en raison de changements dans le filtrage de service. De ce fait, le modèle SCME sépare le modèle SCSM du filtrage de service en créant des instances de machines SCME-FSM pour chaque contexte d'opérations connexes.

Les différents contextes de machines SCME-FSM se distinguent à partir des paramètres d'adresse qui sont fournis par les fonctions d'initialisation. Dans le cas du filtrage de service, ces informations d'adresse sont fournies par des critères de filtrage, c'est-à-dire que toutes les opérations `ActivateServiceFiltering` utilisant la même adresse, font appel à la même machine SCME-FSM gérant cette instance de filtrage de service spécifique. Par exemple, les traitements `ActivateServiceFiltering` apportant des critères différents de filtrage provoquent l'appel à de nouvelles machines SCME-FSM.

En dernier lieu, le gestionnaire d'accès d'entité fonctionnelle (gestionnaire FEAM) retire à l'entité SCME les fonctions d'interface de niveau inférieur. Les fonctions du gestionnaire FEAM sont les suivantes:

- 1) l'établissement et le maintien des interfaces avec les entités SSF, SRF et SDF;
- 2) la retransmission (et la mise en file d'attente, le cas échéant) des messages reçus en provenance des entités SSF, SRF et SDF destinés au modèle SCME;
- 3) le formatage, la mise en file d'attente (si nécessaire) et l'émission de messages provenant de l'entité SCME destinés aux entités SSF, SRF et SDF.

Il convient de noter que le gestionnaire SCSM dispose d'un état et de procédures relatives à la gestion des files d'attente, ce type de gestion des ressources ne représente qu'une façon de gérer les files d'attente des appels dans le réseau. Une autre possibilité est de laisser l'entité SSF/CCF gérer les files d'attente; cependant, les détails techniques sur la manière dont l'entité SSF/CCF assure la gestion des files d'attente ne relèvent pas du domaine du RI. A cet égard, l'objet de commande de ressource ou objet RCO (*resource control object*) (voir 3.1.2.4.5) et l'état de mise en file d'attente du modèle SCSM (état 2.2), ainsi que ses sous-états, événements et procédures associés, ne sont nécessaires et applicables qu'au cas où la gestion de files d'attente est assurée par l'entité SCF.

3.1.2.4 Diagramme de transition d'état partiel de l'entité de gestion SCF (modèle SCME)

Les deux parties essentielles du diagramme d'état de l'entité de gestion SCF (SCME) (*SCF management entity*) sont représentées aux Figures 16 et 17.

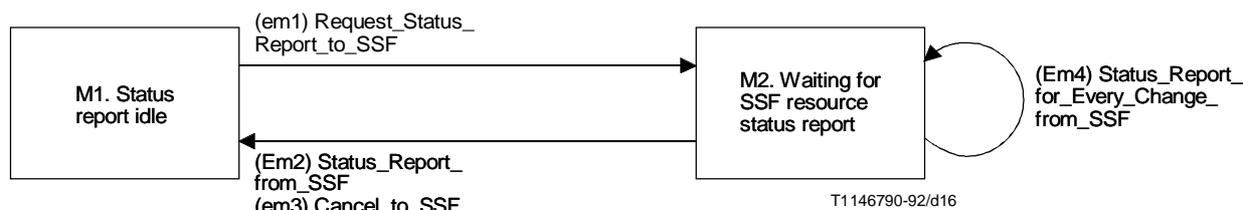


FIGURE 16/Q.1218

Machine FSM de rapport d'état (*status report*) de l'entité SCME

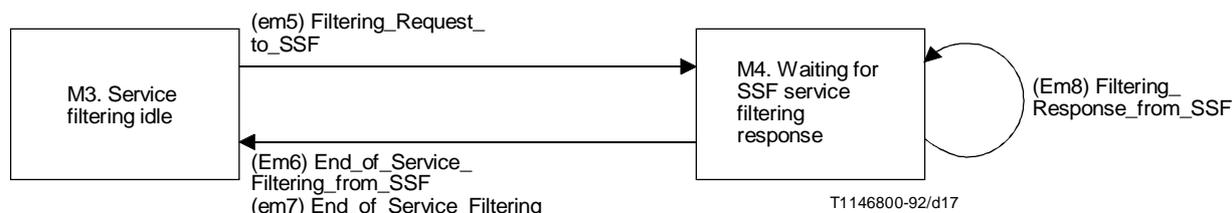


FIGURE 17/Q.1218

Machine FSM de filtrage du service (*service filtering*) de l'entité SCME

Le modèle SCME exécute les opérations suivantes:

- **RequestCurrentStatusReport** (demande de rapport d'état courant);
- **RequestEveryStatusChangeReport** (demande de rapport de tout changement d'état);
- **RequestFirstStatusMatchReport** (demande de rapport de la première coïncidence d'état);
- **CancelStatusReport** (y compris l'identificateur Invoke ID précédemment utilisé pour les opérations request FirstStatusMatchReport ou pour les opérations RequestEveryStateChangeReport);
- **StatusReport** (état rapport);
- **ActivateServiceFiltering** (activer le filtrage de service);
- **ServiceFilteringReport** (rapport de filtrage de service);
- **CallGap** (séquencement d'appels);
- **ActivityTest** (test d'activité).

Emettre les opérations **CallGap** et **ActivityTest** ne donne pas lieu à des transitions d'état dans l'entité SCME. Les procédures de la suite des opérations ci-dessus sont décrites dans ce qui suit.

Les opérations qui ne sont pas mentionnées ci-dessus ne modifient pas l'état de l'entité SCME; ces opérations sont transmises au modèle SCSM concerné.

3.1.2.4.1 Etat M1 – "Status report idle" (Rapport d'état repos)

Événement suivant¹⁾ est pris en compte dans cet état:

- (em1) Request_Status_Report_to_SSF – Événement interne produit par une décision de transmettre l'une des opérations ci-après:
 - request current status report;
 - request first status match report;
 - request every status change report.

Cet événement provoque le passage à l'état M2, waiting for SSF response status report.

3.1.2.4.2 Etat M2 – "Waiting for SSF response status report"

On considère que les événements ci-après se trouvent dans cet état:

- (Em2) Status_Report_from_SSF – Événement interne produit par la réception d'une réponse à une opération de demande de compte rendu sur l'état actuel ou demande de premier compte rendu d'état de correspondance précédemment émise vers l'entité SSF. Cet événement provoque le passage à l'état M1 compte rendu d'état Idle;

¹⁾ Tous les événements sont énumérés, et le numéro d'un événement est précédé d'un préfixe, qui est soit la lettre "E" (pour des événements externes) ou "e" (pour des événements internes) et mis entre parenthèses au début de la désignation de l'événement. Le champ d'application des noms d'événement et des numéros est défini par la machine à états finis dans laquelle ces événements se produisent: la même chose s'applique aux noms d'état.

- (em3) *Cancel_to_SSF* – Événement interne dû à la nécessité pour la logique de service de cesser la surveillance d'état des ressources dans l'entité SSF, et dû à la transmission d'une opération de compte rendu d'état de correspondance à l'intention de l'entité SSF. Cet événement a lieu seulement pour les opérations suivantes précédemment émises: demande de premier compte rendu concernant toutes les opérations relatives aux correspondances (*request first status match report*) ou demande de compte rendu concernant toutes les opérations relatives aux changements d'état (*request every status change report*). Cet événement provoque le passage à l'état M1 compte rendu d'état Idle;
- (Em4) *Status_Report_for_Every_Change_from_SSF* – Événement externe produit par la réception de la réponse à l'opération demande de compte rendu concernant toutes les opérations relatives aux changements d'état (*request every status change report*) précédemment émise à l'intention de l'entité SSF. Cet événement ne provoque pas la sortie hors de cet état, ainsi le modèle SCSM reste toujours dans l'état M2, Attente de compte rendu d'état de ressources de l'entité SSF (*waiting for SSF resource status report*).

3.1.2.4.3 Etat M3 – "Service filtering idle"

On considère que l'événement ci-après se trouve dans cet état:

- (em5) *Filtering_Request_to_SSF* – Événement interne dû à la nécessité pour la logique de service de filtrer les demandes de service vers l'entité SSF, et par la transmission de l'opération Activer le filtrage de services (*activate service filtering*). Cet événement provoque le passage à l'état M4, Attendre la réponse de filtrage de services de l'entité SSF (*waiting for SSF service filtering response*).

3.1.2.4.4 Etat M4 – "Waiting for SSF service filtering response"

Dans cet état, l'entité SCF attend la réponse de filtrage de services en provenance de l'entité SSF. On considère que les événements ci-après se trouvent dans cet état:

- (Em6) *End_of_Service_Filtering_Response_from_SSF* – Événement externe causé par la réception de la réponse à l'opération *request service filtering* précédemment émise vers l'entité SSF à la fin de la durée de filtrage de services. Cet événement provoque le passage à l'état M3, *service filtering Idle*;
- (em7) *End_of_Service_Filtering* – Événement interne, déclenché par la fin de temporisation de durée de filtrage de services dans l'entité SCF. Cet événement provoque le passage à l'état M3, *service filtering Idle*;
- (Em8) *Filtering_Response_from_SSF* – Événement externe, déclenché par la réception de la réponse à l'opération *request service filtering* précédemment émise vers l'entité SSF. Cet événement ne provoque pas de sortie de cet état, et le modèle SCSM reste à l'état M4, Attente de la réponse de filtrage de services *waiting for SSF service filtering response*.

Quand l'opération de filtrage de services est active, une autre opération de filtrage de services peut être adressée à l'entité SSF utilisant les mêmes critères de filtrage; ce second "filtre" se substituant au précédent.

3.1.2.4.5 L'objet commande de ressource

L'objet commande de ressource (RCO) (*resource control object*) est une partie de l'entité de gestion SCF qui gère les données relatives aux renseignements sur les ressources.

L'objet RCO comprend:

- 1) une structure de données qui (par définition) se trouve dans l'entité SDF et à laquelle on ne peut accéder qu'au moyen des méthodes RCO;
- 2) les méthodes RCO.

Pour les besoins de la présente Recommandation, aucune contrainte en matière de réalisation n'a été imposée sur la structure. Les seules conditions requises pour la structure sont que, pour chaque ressource prise en charge, elle:

- 1) mémorise l'état de la ressource (par exemple occupée ou repos);
- 2) gère la file d'attente des modèles SCSM qui attendent cette ressource. Pour la surveillance permanente, l'objet RCO conserve trace de l'état des ressources en utilisant l'opération *request every status change report*.

Les trois opérations ci-après sont définies pour l'objet RCO:

- 1) **Get_Resource** – Cette opération est utilisée pour obtenir l'adresse d'une ligne au repos pour un modèle SCSM. Lorsque la ressource est occupée, le modèle SCSM est mis dans la file d'attente de la ressource;
- 2) **Free_Resource** – Cette opération est utilisée lorsqu'une notification de déconnexion émise par l'entité SSF est reçue. Cette méthode provoque l'avance dans la file d'attente (si elle n'est pas vide) ou marque la ressource libre;
- 3) **Cancel** – Cette opération est utilisée à la fin de la temporisation de la file d'attente ou lorsqu'il y a eu abandon de l'appel.

3.1.2.5 Le modèle SCSM

Le diagramme général du modèle SCSM est indiqué à la Figure 18 pour ce qui est des procédures relatives à la partie machine SCF-FSM de l'entité physique SCP/AD/SN au cours du traitement d'un appel RI. Chaque état dispose de sous-machines FSM composées de sous-états.

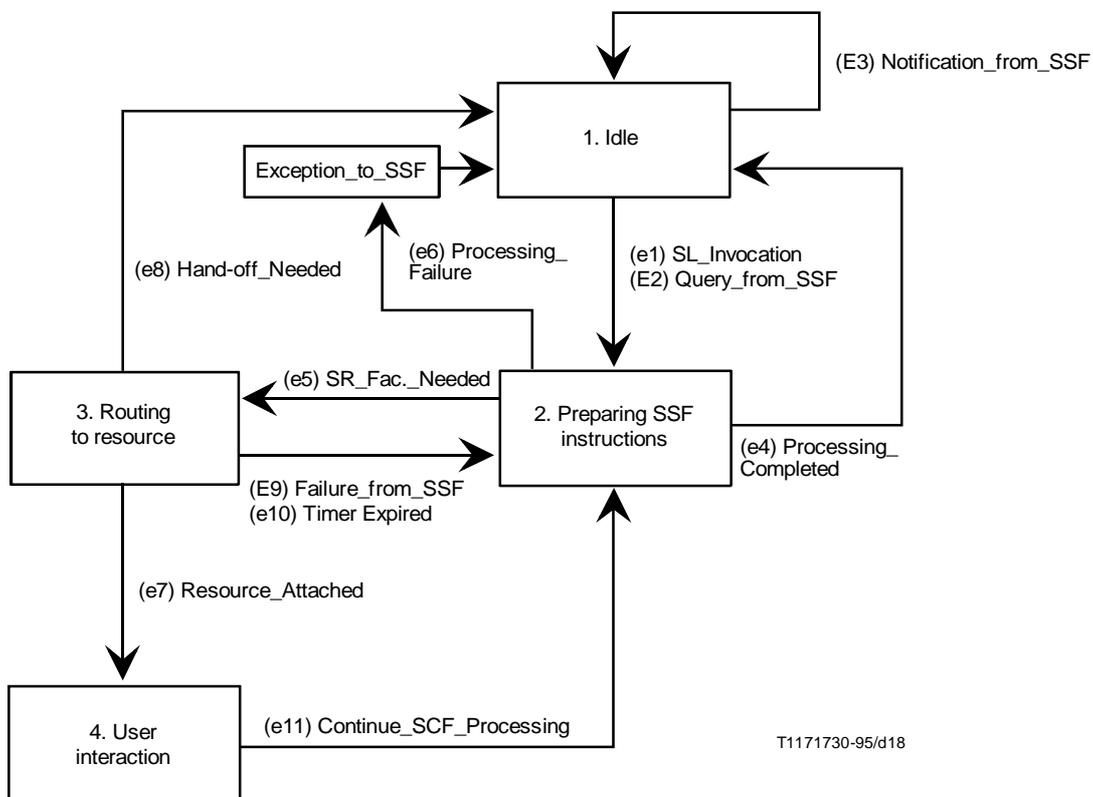


FIGURE 18/Q.1218

Machine à états finis du modèle SCSM

Les règles générales applicables à plusieurs états sont les suivantes:

Dans chaque état, s'il y a une erreur dans une opération reçue, le programme de logique de service (SLP) et les fonctions de maintenance en sont informés. Généralement, le modèle SCSM reste dans le même état; cependant, un traitement différent des erreurs est possible dans des cas spécifiques comme cela est décrit au 3.3. Selon la classe d'opération, l'erreur peut être signalée aux entités SSF, SRF ou SDF (voir Recommandation Q.774).

Il en est également ainsi si, dans chaque état, le modèle SCSM est informé que le dialogue avec l'entité SSF a pris fin, il informe alors le programme de logique de service et revient à l'état Idle. Dans ce cas, toutes les ressources attribuées à cet appel, y compris celles qui sont nécessaires aux dialogues applicables à d'autres fonctions, doivent être libérées. Pour simplifier le diagramme, ces transitions ne sont pas représentées dans les figures.

Lorsque le programme de logique de service (SLP) demande des informations sur l'appel, le modèle SCSM transmet l'opération **CallInformationRequest** à l'entité SSF, et puis le **CallInformationReport** est mis en attente.

Dans tout état (sauf l'état Idle), le modèle SCSM peut recevoir l'opération **CallInformationReport** provenant de l'entité SSF, lorsque l'opération **CallInformationRequest** est mise en attente. Parmi les autres demandes en instance qui devraient être traitées de la même façon que le **CallInformationRequest**, il y a l'opération **ApplyCharging**.

A partir de tout état (sauf l'état Idle), si l'opération **CallInformationReport** est mise en attente et que le programme de logique de service indique que le processus a été terminé, le modèle SCSM reste dans le même état jusqu'à la réception de l'opération **CallInformationReport**.

Les règles générales pour une composante ou une série de composantes envoyée dans un ou plusieurs messages du sous-système TCAP, qui peut inclure une seule opération ou plusieurs opérations, sont spécifiées dans 3.1.1.5 et ne sont pas décrites ici.

Le modèle SCSM dispose d'un temporisateur d'application, le $T_{SCF-SSF}$, qui a pour objet de réinitialiser le temporisateur T_{SSF} utilisé pour éviter d'avoir des durées de suspension d'appel trop importantes et pour conserver l'association entre l'entité SSF et l'entité SCF.

Le temporisateur $T_{SCF-SSF}$ est initialisé dans les cas suivants:

- quand l'entité SCF reçoit une opération de type **InitialDP**, **TAnswer**, **TDisconnect**, **TermAttemptAuthorized**, **TMidCall**, **TNoAnswer**, **AnalysedInformation**, **TBusy**, **CollectedInformation**, **OAnswer**, **OCalledPartyBusy**, **ODisconnect**, **OMidCall**, **ONoAnswer**, **OriginationAttemptAuthorized**, **RouteSelectFailure** ou **AssistRequestInstructions** (voir 3.1.2.5.2.1 état 2.1: "**Préparer les instructions SSF**", et 3.1.2.5.2.2.1 état 2.2.1: "**Préparer les instructions SSF**"). Dans ce cas, ce temporisateur est réinitialisé lorsque la première demande, autre que l'opération **ResetTimer** (Remise à zéro de la temporisation), est envoyée à l'entité SSF. Postérieurement à l'envoi de la première réponse, l'entité SCF peut réinitialiser le temporisateur T_{SSF} un nombre quelconque de fois en utilisant l'opération **ResetTimer** et également réinitialiser le temporisateur $T_{SCF-SSF}$. Avant l'envoi de la première réponse, à la fin de la temporisation de $T_{SCF-SSF}$, l'entité SCF peut réinitialiser une fois le T_{SSF} , en utilisant l'opération **ResetTimer**, et également réinitialiser $T_{SCF-SSF}$. A l'expiration de la deuxième temporisation de $T_{SCF-SSF}$, le modèle SCSM informe le programme de logique de service (SLP) et les fonctions de maintenance, et le modèle SCSM passe à l'état **Idle**;
- quand l'entité SCF entre dans l'état Préparer les Instructions SSF après avoir envoyé l'opération **InitiateCallAttempt** ou l'opération **DisconnectForwardConnection** ou après avoir reçu un compte rendu sur EDP-R provenant de l'entité SSF. Dans ce cas, l'entité SCF peut réinitialiser le temporisateur T_{SSF} en utilisant l'opération **ResetTimer** un nombre quelconque de fois;
- quand l'entité SCF passe au sous-état Queueing (Mise en attente) (voir 3.1.2.5.2.2, état 2.2.2: "**Queueing**"). Dans ce cas, à la fin de la temporisation de $T_{SCF-SSF}$, l'entité SCF peut réinitialiser le temporisateur T_{SSF} en utilisant l'opération **ResetTimer** (remise à zéro de la temporisation) un nombre quelconque de fois;
- quand l'entité SCF passe à l'état "**Attendre les instructions de demande d'assistance**" ou à l'état "**Interaction avec l'utilisateur**" (voir 3.1.2.5.3.2 et 3.1.2.5.4). Dans ces cas, à l'expiration de la temporisation de $T_{SCF-SSF}$, l'entité SCF peut réinitialiser T_{SSF} en utilisant l'opération **ResetTimer** un nombre quelconque de fois (FACULTATIF).

NOTE – Le terme "FACULTATIF" fait référence à l'utilisation du temporisateur d'application $T_{SCF-SSF}$. Qu'il soit ou non utilisé dépend de la réalisation, mais, s'il est utilisé, il doit être synchronisé avec le temporisateur T_{SSF} dans l'automate SSF-FSM.

Dans l'ensemble des quatre cas, la temporisation $T_{SCF-SSF}$ peut avoir quatre valeurs différentes définies par l'application. La valeur de la temporisation $T_{SCF-SSF}$ est inférieure à la valeur correspondante de la temporisation T_{SSF} .

Dans tous les autres cas, en recevant ou en envoyant n'importe quelle opération, l'entité SCF devrait réinitialiser le $T_{SCF-SSF}$. Dans l'état "**Attente de la notification ou de la demande**" (voir 3.1.2.5.2.3), le temporisateur $T_{SCF-SSF}$ n'est pas utilisé.

Le modèle SCSM dispose également d'un temporisateur d'application, le temporisateur $T_{ASSIST/HANDOFF}$, dont l'objet est d'empêcher un temps de suspension assistance/non-assistance excessif. Le modèle SCSM arme le temporisateur $T_{ASSIST/HANDOFF}$ quand le modèle SCSM envoie une opération **EstablishTemporaryConnection** ou **Select Route/Connect** avec une corrélation ID. Ce temporisateur est arrêté quand le modèle SCSM reçoit une opération **AssistRequestInstructions** à partir de l'entité SSF d'assistance/de non-assistance. A la fin de la temporisation $T_{ASSIST/HANDOFF}$, le modèle SCSM informe les fonctions de logique de service et de maintenance. Ensuite, dans le cas de l'assistance, le modèle SCSM passe à l'état "**Préparer les instructions SSF**", et l'instruction SLPI sera libérée dans le cas de la non-assistance.

Les opérations relatives à la commande d'appel associées à l'interface SCF-SSF (à l'exception des opérations relatives à l'entité SCME) sont classées dans les catégories suivantes:

- 1) opérations relatives au traitement d'appel; et
- 2) opérations relatives au traitement des non-appels.

Les opérations de traitement des appels sont regroupées dans les deux séries suivantes:

- **CollectInformation** (collecter les informations);
- **AnalyseInformation** (analyser les informations);
- **SelectFacility** (sélectionner un service complémentaire);
- **SelectRoute** (sélectionner le trajet);
- **Connect** (connecter);
- **Continue** (continuer).

et

- **InitiateCallAttempt** (déclencher une tentative d'appel);
- **ConnectToResource** (connecter à la ressource);
- **DisconnectForwardConnection** (déconnecter la connexion vers l'avant);
- **ReleaseCall** (libérer l'appel);
- **EstablishTemporaryConnection** (établir une connexion temporaire).

En ce qui concerne le premier ensemble d'opérations associées au traitement des appels, l'entité SCF peut ne pas envoyer deux opérations du même ensemble dans une série de messages TCAP ou dans une séquence de composantes à l'entité SSF, mais les envoyer seulement une par une. Deux opérations du même ensemble seront séparées par au moins un message EDP-R reçu par l'événement SCSM. Cela s'applique à toute opération du premier ensemble suivie par **ConnectToResource** ou **EstablishTemporaryConnection**.

Les opérations de traitement des non-appels comprennent le reste des opérations à l'interface SCF-SSF mais ne comprennent pas les opérations associées à l'entité SCME. Lorsque la logique de service doit envoyer des opérations en parallèle, ces opérations sont envoyées dans la séquence composante.

Dans ce qui suit, chaque état, ainsi que les événements qui provoquent la sortie de cet état, fait l'objet d'une description dans un paragraphe distinct. Les sorties sont présentées dans des rectangles plus petits que les rectangles représentant les états; à l'inverse des états et des événements, les sorties ne sont pas numérotées.

3.1.2.5.1 Etat 1 – "Idle"

On considère que les événements ci-après sont dans cet état:

- (e1) **SL_Invocation** – Événement interne dû à la nécessité pour la logique de service de commencer un appel. Le modèle SCSM transmet l'opération **InitiateCallAttempt** à l'entité SSF;
- (E2) **Query_from_SSF** – Événement externe, déclenché par la réception d'une des opérations suivantes:
 - **InitialDP**;
 - **AssistRequestInstructions** (pour le cas du service avec passation);
 - **TAnswer**;
 - **TDisconnect**;
 - **TermAttemptAuthorized**;
 - **TMidCall**;
 - **TNoAnswer**;
 - **AnalysedInformation**;
 - **TBusy**;
 - **CollectedInformation**;
 - **OAnswer**;
 - **OCalledPartyBusy**;
 - **ODisconnect**;
 - **OMidCall**;
 - **ONoAnswer**;
 - **OriginationAttemptAuthorized**;
 - **RouteSelectFailure**.

Dans le cas de la passation, le temporisateur $T_{ASSIST/HANDOFF}$ (assistance, passation) est arrêté.

Les deux événements provoquent le passage à l'état 2, "**Préparer les instructions SSF**".

- (E3) Notification_from_SSF – Ceci est un événement externe déclenché par la réception d'une opération de type **InitialDP** ou **TAnswer**, **TDisconnect**, **TermAttemptAuthorized**, **TMidCall**, **TNoAnswer**, **AnalysedInformation**, **TBusy**, **CollectedInformation**, **OAnswer**, **OCalledPartyBusy**, **ODisconnect**, **OMidCall**, **ONoAnswer**, **OriginationAttemptAuthorized**, **RouteSelectFailure** notifiant la détection d'un point TDP_N dans l'entité SSF. Cet événement provoque le retour au même état.

3.1.2.5.2 Etat 2 – "Préparer les instructions SSF"

Dans cet état, l'entité SCF détermine la façon de poursuivre le traitement.

On considère que les événements ci-après sont dans cet état:

- (e4) Processing_Completed – Événement interne. Dans ce cas, l'entité SCF a achevé le traitement des instructions vers l'entité SSF. Cet événement provoque une réponse à envoyer à l'entité SSF et le passage à l'état 1, **Idle**;
- (e5) SR_Facilities_Needed – Événement (interne), dû à la nécessité de la logique de service de disposer d'autres informations en provenance du participant; il est donc nécessaire d'établir une connexion entre le participant et l'entité SRF. Cet événement provoque le passage à l'état 3, **Routing to Resource**;
- (e6) Processing_Failure – Cet événement (interne) déclenche le traitement d'exception approprié et un retour à l'état 1, **Idle**.

NOTE – Ici et ultérieurement dans la présente Recommandation, le traitement d'exception n'est pas défini. On suppose cependant qu'il doit inclure la libération de toutes les ressources mobilisées et l'envoi d'un message de réponse approprié à l'intention de l'entité SSF.

Pour décrire plus avant les procédures associées à cet état, celui-ci est divisé en trois sous-états qui sont décrits dans les trois paragraphes suivants: (cette subdivision est illustrée à la Figure 19).

3.1.2.5.2.1 Etat 2.1 – "Préparation des instructions SSF"

Dans l'état 2.1, "**Préparer les instructions SSF**", les premières décisions sont prises pour savoir si l'information SDF ou une ressource spécialisée est nécessaire, si la mise en file d'attente est prise en charge, etc. En outre, le traitement associé au point EDP-R est également assuré dans cet état.

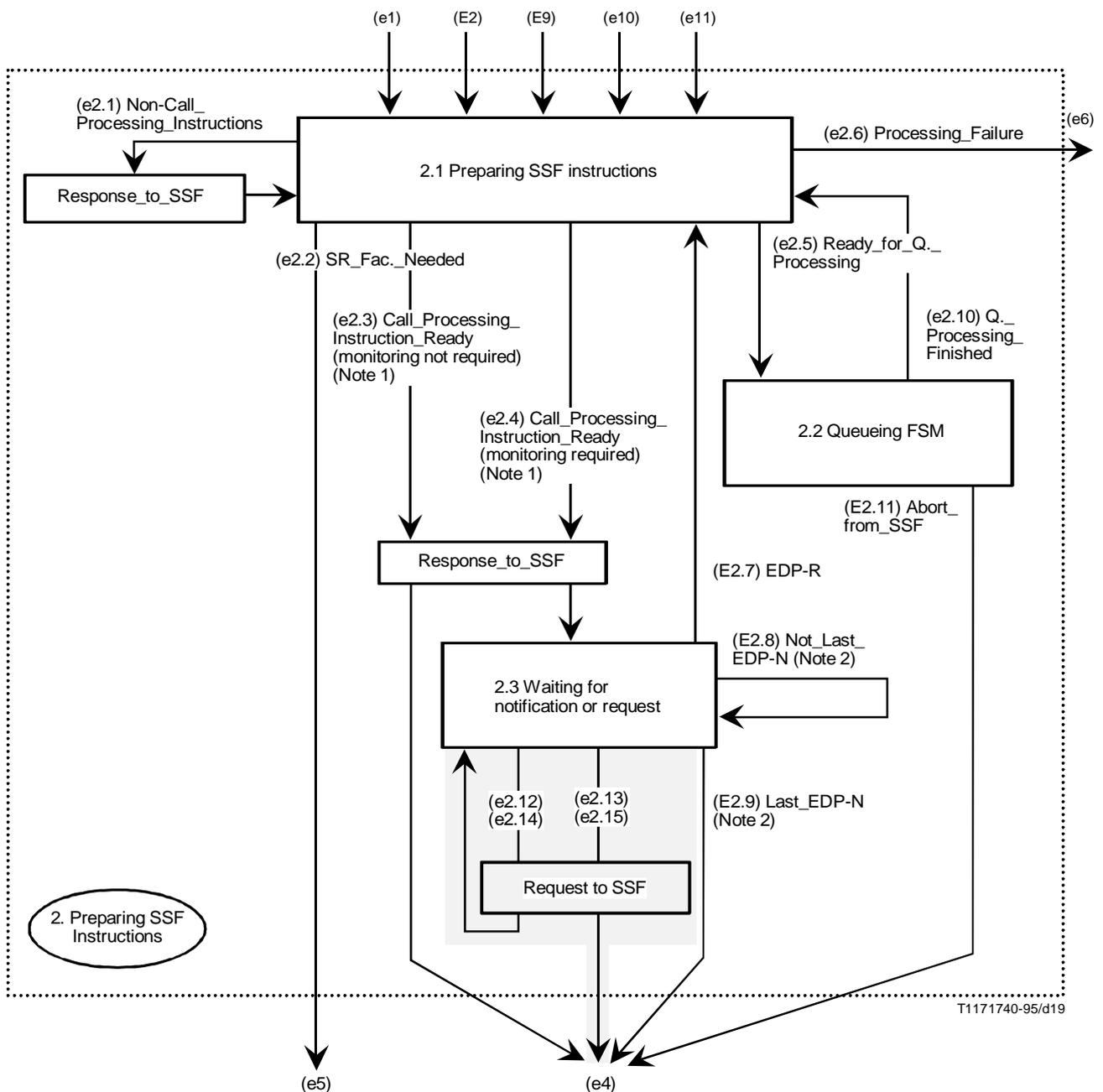
Lorsqu'il passe à cet état, le modèle SCSM arme ou réinitialise le temporisateur T_{SCF-SSF}.

On considère les événements ci-après dans cet état:

- (e2.1) Non-Call_Processing_Instructions – Ceci est un événement interne provoqué par la logique de traitement de service quand il est nécessaire d'envoyer une opération vers une entité SSF. Il entraîne la présentation d'une ou plusieurs des opérations suivantes à l'entité SSF:
 - **ApplyCharging** (applique la taxation);
 - **CallInformationRequest** (demande d'information relative à l'appel);
 - **Cancel** (allRequests) (annulation de toutes les demandes d'informations d'appel);
 - **FurnishChargingInformation** (fournir les informations relatives à la taxation);
 - **RequestReportBCSMEvent** (demande de rapport d'événement BCSM);
 - **RequestNotificationChargingEvent** (demande de notification d'événement de taxation);
 - **ResetTimer** (remise à zéro du temporisateur);
 - **SendChargingInformation** (envoyer l'information de taxation).

Cet événement provoque le retour à l'état 2.1, "**Préparer les instructions SSF**".

- (e2.2) SR_Facilities_Needed – Ceci est un événement interne, causé par la logique de service lorsqu'il est nécessaire d'utiliser l'entité SRF. Cet événement correspond à l'événement SCSM (e5).
- (e2.3) Call_Processing_Instruction_Ready (Monitoring not required) (surveillance non nécessaire) – Ceci est un événement interne produit par la logique de service quand une opération relative au traitement d'appel final est prête et qu'aucun point EDP n'est armé et qu'il n'y a pas d'opérations particulières **CallInformationReport**, **EventNotificationCharging** ou **ApplyChargingReport** en attente. Il provoque l'émission d'une des opérations suivantes à l'attention de l'entité SSF:
 - **AnalyseInformation**;
 - **Connect**;
 - **Continue**;
 - **ReleaseCall**;
 - **SelectFacility**;
 - **SelectRoute**.



NOTES

1 Comprenant les opérations Call Information Request, Apply Charging avec les opérations report request et Request Notification Charging Event.

2 Comprenant les opérations Call Information Report, Apply Charging Report et Event Notification Charging.

(e2.12) Notification_or_Request_Continuing_Instruction
 (e2.13) Monitoring_Cancel_Instruction
 (e2.14) Release_Call_Instruction (Call Information Report or Apply Charging Report a été demandé)
 (e2.15) Release_Call_Instruction (ni Call Information Report ni Apply Charging Report n'a été demandé)

FIGURE 19/Q.1218
 Développement partiel de l'état 2 de la machine FSM

En outre, une ou plusieurs des opérations ci-après peuvent être émises à l'entité SSF avant les opérations précitées:

- **Cancel (allRequests);**
- **RequestReportBCSMEEvent** (pour désarmer tous les points EDP armés);
- **FurnishChargingInformation;**
- **SendChargingInformation.**

Cet événement correspond à l'événement SCSM (e4).

- (e2.4) **Call_Processing_Instruction_Ready** (surveillance nécessaire) – Ceci est un événement interne produit par la logique de service lorsqu'une opération relative au traitement d'appel est prête et que la surveillance de l'appel est nécessaire (par exemple, un point EDP est armé ou il y a une opération spéciale **CallInformationReport**, ou une opération spéciale **ApplyChargingReport**, ou bien il est nécessaire d'émettre une telle demande). Il provoque l'émission des opérations suivantes à l'intention de l'entité SSF:

- **AnalyseInformation;**
- **CollectInformation;**
- **Connect;**
- **Continue;**
- **ReleaseCall;**
- **SelectFacility;**
- **SelectRoute.**

En outre, une ou plusieurs des opérations ci-après peuvent être émises à l'entité SSF avant une des opérations précitées:

- **ApplyCharging;**
- **CallInformationRequest;**
- **FurnishChargingInformation;**
- **RequestReportBCSMEEvent;**
- **RequestNotificationChargingEvent;**
- **SendChargingInformation.**

Cet événement provoque le passage à l'état 2.3, "**Attente de la notification ou de la demande**".

- (e2.5) **Ready for Queueing Processing** – Ceci est un événement interne produit par la logique de service lorsque l'appel doit être mis en file d'attente. Cet événement provoque le passage à l'état 2.2 **Queueing FSM**.
- (e2.6) **Processing_Failure** – Ceci est un événement interne, correspondant à l'événement SCSM (e6) **Processing_Failure**.

3.1.2.5.2.2 Etat 2.2 – "Queueing FSM"

Lorsque l'entité SCF traite la demande émise par l'entité SSF/CCF, elle peut constater que la ressource vers laquelle l'appel doit être acheminé n'est pas disponible. Par exemple, la ressource en question peut être déjà "occupée".

NOTE – La façon dont l'état des ressources est entretenu est décrite en 3.1.2.4.5.

Une telle ressource peut être une ligne individuelle ou un faisceau ou un groupe de lignes ou de jonctions défini par le client. Dans ce dernier cas, le terme "occupé" signifie que toutes les lignes ou tout le faisceau est occupé, et le mot "repos" signifie qu'au moins une ligne ou une jonction du groupe est en repos.

Si la ressource est occupée, l'entité SCF peut mettre l'appel dans une file d'attente et le reprendre plus tard lorsque la ressource en question est au repos. Les opérations ci-après peuvent être envoyées dans cet état:

- **HoldCallInNetwork (mise en garde de l'appel dans le réseau);**
- **ApplyCharging;**
- **CallInformationRequest;**
- **FurnishChargingInformation;**
- **RequestReportBCSMEEvent;**
- **RequestNotificationChargingEvent;**
- **ResetTimer;**
- **SendChargingInformation.**

Les événements suivants sont considérés dans cet état:

- (e2.10) *Queueing_Processing_Finished* – Ceci est un événement interne produit par le programme de logique de service lorsqu'il est prêt à préparer une opération relative à l'appel à l'intention de l'entité SSF. Cet événement provoque le passage à l'état 2.1, "**Préparer les instructions SSF**".
- (E2.11) *Abort_from_SSF* – Ceci est un événement externe déclenché par la réception du message **Abort** provenant de l'entité SSF (sur abandon d'appel), et qui provoque une transition qui correspond à l'événement SCSM (e4).

Cet état se développe ensuite dans une machine FSM, qui est décrite à la Figure 20.

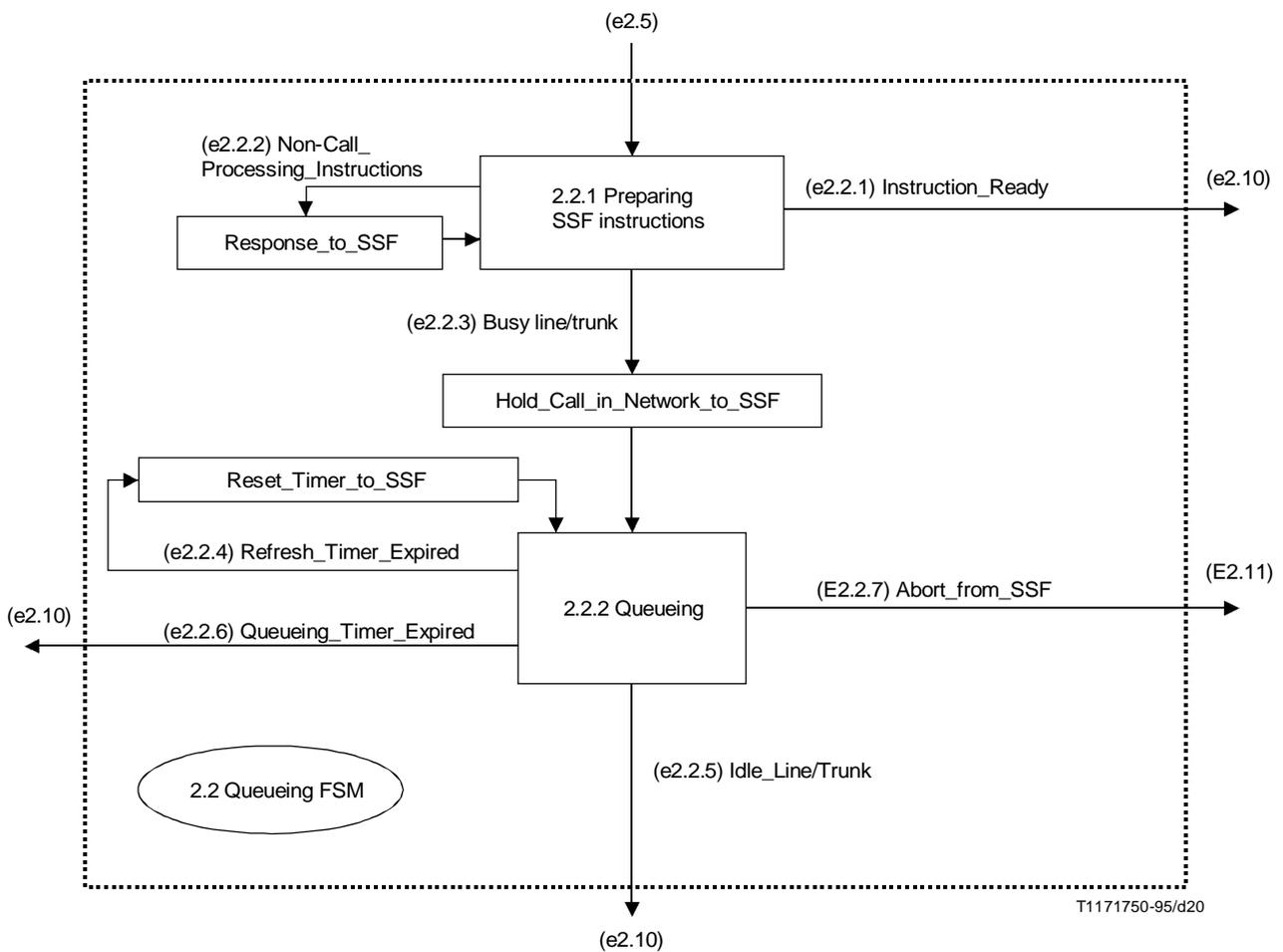


FIGURE 20/Q.1218

Développement partiel de l'état 2 de la machine FSM concernant la mise en file d'attente

Cette machine FSM ne décrit pas explicitement toutes les combinaisons possibles de fonctions de surveillance des ressources utilisées pour la mise en file d'attente. Les possibilités suivantes peuvent être utilisées dans les réalisations:

- **RequestFirstStatusMatchReport** au moyen de l'entité SCME;
- **RequestCurrentStatusChangeReport** au moyen de l'entité SCME;

- **RequestEveryStatusChangeReport** au moyen de l'entité SCME;
- surveillance fondée sur l'émission par le modèle SCSM de l'opération **RequestEventReportBCSM** et réception consécutive de l'opération **EventReportBCSM** pour signaler la disponibilité de la ressource. Les opérations Request et Report apparaissent dans un unique contexte d'appel différent. Dans ces cas, les opérations à l'intention de l'entité SDF ou la fonctionnalité SCF équivalente peuvent être utilisées pour scruter l'état des ressources.

Dans la suite du présent paragraphe, après avoir donné une description état par état de la machine FSM, on donnera une description des mécanismes de prise en charge de l'entité SCME.

3.1.2.5.2.2.1 Etat 2.2.1 – "Préparer les instructions SSF"

Dans cet état, l'entité SCSM prépare les instructions à l'intention de l'entité SSF pour l'exécution de l'appel. On considère que les événements ci-après sont dans cet état:

- (e2.2.1) Instruction_Ready – Événement interne se produisant seulement lorsque la ressource demandée est disponible. Dans ce cas, le modèle SCSM a obtenu l'adresse de la ressource libre au moyen de l'opération Get_Resource de l'objet Resource Control Object (voir 3.1.2.4.5). Cet événement provoque le passage à l'état 2.1 "**Préparer les instructions SSF**" [passage à (e2.10)];
- (e2.2.2) Non-Call_Processing_Instructions – Événement interne produit par la logique de service lorsqu'il est nécessaire d'envoyer une telle opération vers l'entité SSF. Il entraîne l'envoi d'une ou de plusieurs des opérations ci-après à l'entité SSF:
 - **ApplyCharging;**
 - **CallInformationRequest;**
 - **FurnishChargingInformation;**
 - **RequestReportBCSMEvent;**
 - **RequestNotificationChargingEvent;**
 - **ResetTimer;**
 - **SendChargingInformation.**

Cet événement provoque le retour à l'état 2.2.1 "**Préparer les instructions SSF**".

- (e2.2.3) Busy_Line/Trunk (Ligne/Jonction occupée) – Événement interne causé par l'objet Resource Control Object lorsque aucune ligne/jonction n'est disponible. Cet événement produit l'envoi de l'opération à l'entité SSF avec une valeur adéquate du temporisateur de mise en file d'attente, et une transition à l'état 2.2.2, **Queueing**.

3.1.2.5.2.2.2 Etat 2.2.2 – "Queueing" (Mise en file d'attente)

Dans cet état, le modèle SCSM attend une indication de la part de l'objet Resource control object pour acheminer un appel vers une jonction/ligne au repos. Lorsque le modèle SCSM se trouve dans cet état, diverses annonces sont diffusées. Dans la présente Recommandation, on n'a pas développé plus avant cet état; il n'est toutefois pas différent de celui des états 3 et 4 du modèle SCSM. Néanmoins, si les annonces sont achevées avant que l'appel ne sorte de sa file d'attente et si la machine SSF-FSM passe à l'état "**Attente des instructions**", l'opération devrait être envoyée pour fixer une valeur appropriée au temporisateur T_{SSF} . Une fois que le modèle SCSM passe à cet état, le temporisateur de mise en attente est déclenché et le temporisateur $T_{SCF-SSF}$ est réinitialisé. Les rôles respectifs de ces temporisateurs sont les suivants:

- 1) le temporisateur de mise en file d'attente limite le temps pendant lequel un appel peut se trouver dans la file d'attente, et sa valeur peut dépendre du client;
- 2) le temporisateur $T_{SCF-SSF}$ signale le moment où l'opération **ResetTimer** doit être envoyée à l'entité SSF-CCF pour que celle-ci n'abandonne pas l'appel. La valeur de cette temporisation est définie en coordination avec celle du temporisateur T_{SSF} concerné dans l'entité SSF-CCF.

L'opération **FurnishChargingInformation** peut être envoyée à l'entité SSF à ce moment pour indiquer le déclenchement de la mise en file d'attente à des fins d'enregistrement des appels.

On considère que les événements ci-après sont dans cet état:

- (e2.2.4) Refresh_Timer_Expired – Événement interne qui donne lieu à l'envoi de l'opération **ResetTimer** à l'entité SSF-CCF et un retour au même état;
- (e2.2.5) Idle_Line/Trunk – Événement interne qui correspond à l'événement état 2 (e2.10);

- (e2.2.6) `Queueing_Timer_Expired` – Événement interne qui donne lieu au traitement de l'opération `Cancel` de l'objet `Resource control object` et provoque le passage à l'état 2.1 "**Préparer les instructions SSF**" [Transition (e2.10)] (la procédure suivante dépend de la décision de la logique de service qui diffuse ou non l'annonce vocale terminale);
- (E2.2.7) `Abort_from_SSF` – Événement externe déclenché par la réception du message `Abort` provenant de l'entité `SSF` (sur abandon d'appel); il provoque une transition qui correspond à l'événement état 2 (E2.11). L'entité de gestion de commande de service (`SCME`) veille à la mise à jour des données de mises en file d'attente au moyen de l'opération `Cancel` de l'objet `Resource object`.

3.1.2.5.2.3 Etat 2.3 – "Attente de la notification ou de la demande"

Dans cet état, le modèle `SCSM` attend une notification ou une demande de l'entité `SSF`.

Dès le passage à cet état, le modèle `SCSM` arrête le temporisateur `TSCF-SSF`.

On considère que les événements ci-après sont dans cet état:

- (E2.7) `EDP-R` – Événement externe, déclenché par la réception des opérations suivantes:
 - `EventReportBCSM` (pour un point `EDP_R`);
 - `TAnswer`;
 - `TDisconnect`;
 - `TermAttemptAuthorized`;
 - `TMidCall`;
 - `TNoAnswer`;
 - `AnalysedInformation`;
 - `TBusy`;
 - `CollectedInformation`;
 - `OAnswer`;
 - `OCalledPartyBusy`;
 - `ODisconnect`;
 - `OMidCall`;
 - `ONoAnswer`;
 - `OriginationAttemptAuthorized`;
 - `RouteSelectFailure`.

Cet événement provoque le passage à l'état 2.1 "**Préparer les instructions SSF**".

- (E2.8) `Not_Last_EDP-N` – Événement externe causé par la réception de l'une des opérations suivantes:
 - `ApplyChargingReport`;
 - `CallInformationReport`;
 - `EventReportBCSM` (pour un point `EDP_N`);
 - `EventNotificationCharging`;
 - `TAnswer`;
 - `TDisconnect`;
 - `TermAttemptAuthorized`;
 - `TMidCall`;
 - `TNoAnswer`;
 - `AnalysedInformation`;
 - `TBusy`;
 - `CollectedInformation`;
 - `OAnswer`;
 - `OCalledPartyBusy`;
 - `ODisconnect`;
 - `OMidCall`;
 - `ONoAnswer`;
 - `OriginationAttemptAuthorized`;
 - `RouteSelectFailure`.

Dans ce cas, il y a encore un point `EDP` particulier armé ou une opération particulière `CallInformationReport` ou `ApplyChargingReport` en attente. Cet événement provoque le retour à l'état 2.3 **Attente de la notification ou de la demande**.

- (E2.9) Last_EDP-N – Événement externe, déclenché par la réception d'une des opérations suivantes:
 - **ApplyChargingReport;**
 - **CallInformationReport;**
 - **EventReportBCSM** (pour le point EDP_N);
 - **TAnswer;**
 - **TDisconnect;**
 - **TermAttemptAuthorized;**
 - **TMidCall;**
 - **TNoAnswer;**
 - **AnalysedInformation;**
 - **TBusy;**
 - **CollectedInformation;**
 - **OAnswer;**
 - **OCalledPartyBusy;**
 - **ODisconnect;**
 - **OMidCall;**
 - **ONoAnswer;**
 - **OriginationAttemptAuthorized;**
 - **RouteSelectFailure.**

Dans ce cas, il n'y a pas de point EDP armé particulier et pas d'opération **CallInformationReport** ou **ApplyChargingReport** en attente. Cet événement correspond à l'événement SCSM (e4).

- (e2.12) Notification_or_Request_Continuing_Instruction – Événement interne, déclenché par la logique de service lorsqu'il est nécessaire d'envoyer une telle opération à l'entité SSF. L'une des opérations suivantes est alors transmise à l'entité SSF:
 - **RequestReportBCSMEvent** (pour lequel:
 - i) un ou plusieurs points EDP seront armés,
 - ii) certains points EDP armés seront désarmés,
 - iii) tous les points EDP armés seront désarmés quand il y a d'autres demandes en cours);
 - **FurnishChargingInformation;**
 - **ApplyCharging;**
 - **RequestNotificationChargingEvent;**
 - **SendChargingInformation.**

Cet événement provoque le passage à l'état 2.3 "**Attente de la notification ou de la demande**".

- (e2.13) Monitoring_Cancel_Instruction – Événement interne, déclenché par la logique de service lorsqu'il est nécessaire d'envoyer cette opération à l'entité SSF. L'une des opérations suivantes est alors transmise à l'entité SSF:
 - **Cancel** (all Requests) (toutes les demandes);
 - **RequestReportBCSMEvent** (pour lequel tous les points EDP armés seront désarmés lorsqu'il n'y aura plus d'autres demandes en cours).

Cet événement correspond à l'événement SCSM (e4).

- (e2.14) Release_Call_Instruction (**CallInformationReport** ou **ApplyChargingReport** a été demandé) – Événement interne, déclenché par la logique de service lorsqu'il est nécessaire d'envoyer une telle opération à l'entité SSF. L'une des opérations suivantes est alors transmise à l'entité SSF:
 - **ReleaseCall** (lorsqu'il y a un message **CallInformationReport** particulier ou un message **ApplyChargingReport**).

Cet événement provoque le passage à l'état 2.3 "**Attente de la notification ou de la demande**".

- (e2.15) Release_Call_Instruction (aucun des messages **CallInformationReport** ou **ApplyChargingReport** n'a été demandé) – Ceci est un événement interne provoqué par la logique de service quand il est nécessaire d'envoyer une telle opération à l'entité SSF; l'opération suivante est alors transmise à l'entité SSF:
 - **ReleaseCall** (quand il n'y a pas de message **CallInformationReport** particulier ou de message **ApplyChargingReport**).

Cet événement correspond à l'événement SCSM (e4).

Fin de la description état 2 "**Préparer les instructions SSF**".

3.1.2.5.3 Etat 3 – "Routing to Resource" (Acheminement vers la ressource)

La ressource est l'un des services complémentaires SRF (par exemple, périphérique intelligent).

Dans cet état, il est nécessaire d'avoir des interactions avec l'entité SSF. En conséquence, les événements ci-après provoquent la sortie de cet état:

- (e7) **Resource_Attached** – L'entité SRF est disponible. Cet événement provoque le passage à l'état 4, **User Interaction**;
- (e8) **Handoff_Needed** – Lorsque la procédure de passation est déclenchée, le modèle SCSM met un terme à l'interaction avec l'entité SSF qui est à l'origine de la demande. Cet événement provoque le passage à l'état 1, **Idle**. Lorsque l'opération **AssistRequestInstructions** est reçue par l'organe de commande du modèle SCME en provenance de l'entité SSF vers laquelle la passation est effectuée, l'organe de commande du modèle SCME crée un nouveau modèle SCSM. L'entité SCF conservera suffisamment d'informations pour effectuer la corrélation entre l'opération subséquente **AssistRequestInstructions** (de l'entité SSF de passation) et l'occurrence de l'utilisation du script de la logique de service;
- (E9) **Failure_from_SSF** – L'entité SSF n'est pas en mesure d'établir une connexion avec les ressources demandées, ce qui provoque le passage à l'état 2, "**Préparer les instructions SSF**";
- (e10) **Timer_Expired** – Cet événement correspond à la fin de la temporisation $T_{ASSIST/HANDOFF}$. Cet événement provoque le passage à l'état 2, "**Préparer les instructions SSF**".

Afin de continuer à décrire les procédures afférentes à cet état, l'état est subdivisé en trois sous-états qui sont décrits dans les deux paragraphes suivants (cette sous-division est illustrée à la Figure 21).

3.1.2.5.3.1 Etat 3.1 – "Déterminer le mode"

Dans cet état, le modèle SCSM détermine le mode d'interaction avec l'utilisateur pour connecter l'appel à l'entité SRF. On considère que les événements ci-après sont dans cet état:

- (e3.1) **Instruction_Ready** – Événement interne qui a lieu seulement dans l'entité SSF déclencheuse relais. Dans ce cas, le modèle SCSM envoie à l'entité déclencheuse SSF l'opération **ConnectToResource** accompagnée de l'opération **PlayAnnouncement** ou **PromptAndCollectUserInformation** et passe à l'état 4 **User Interaction**. Ce passage correspond à l'événement (e7);
- (e3.2) **Assist_Needed** – Événement interne qui se produit lorsque l'entité SSF assistante ou lorsque la relation directe SCF-SRF est nécessaire. Dans ce cas, le modèle SCSM envoie l'opération **EstablishTemporaryConnection** vers l'entité SSF déclencheuse avec l'adresse de l'entité SSF assistante ou l'adresse de l'entité SRF, et passe à l'état 3.2 **WaitingForAssistRequestInstructions**;
- (e3.3) **Handoff_Needed** – Cet événement interne ne se produit que dans le cas du relais. Dans ce cas, le modèle SCSM envoie l'opération **Connect** ou **SelectRoute** avec l'adresse de l'entité SSF vers laquelle la passation est effectuée, vers l'entité SSF déclencheuse, initialise le temporisateur $T_{ASSIST/HANDOFF}$ et passe à l'état 1, **Idle** (repos). Cette transition correspond à l'événement (e8). L'entité SCF conservera suffisamment d'informations pour établir une corrélation entre l'opération subséquente **AssistRequestInstructions** (à partir de l'entité SSF relais) et l'occurrence de l'utilisation du script de la logique de service.

3.1.2.5.3.2 Etat 3.2 – "Attendre les instructions de demande d'assistance"

Dans cet état, le modèle SCSM attend l'opération **AssistRequestInstructions** provenant de l'entité SSF assistante (relais SSF) ou de l'entité SRF (relation directe SCF-SRF). Après passage à cet état, le modèle SCSM déclenche le temporisateur $T_{ASSIST/HANDOFF}$, et réinitialise le temporisateur $T_{SCF-SSF}$ (s'il est utilisé). On considère que les événements ci-après sont dans cet état:

- (E3.4) **Assist_Request_Instructions_from_SSF** (assisting SSF relay case) – Événement externe causé par la réception de l'opération **AssistRequestInstructions** provenant de l'entité SSF assistante. Dans ce cas, le modèle SCSM transmet l'opération **ConnectToResource** accompagnée de l'opération **PlayAnnouncement** ou **PromptAndCollectUserInformation** à l'entité SSF assistante, et passe à l'état 4, **User Interaction**. Cette transition correspond à l'événement (e7);
- (E3.5) **Assist_Request_Instructions_from_SRF** (cas SCF-SRF) – Événement externe déclenché par la réception de l'opération **AssistRequestInstructions** provenant de l'entité SRF. Dans ce cas, le modèle SCSM transmet l'opération **PlayAnnouncement** ou l'information **PromptAndCollectUserInformation** vers l'entité SRF, et passe à l'état 4, **User Interaction**. Cette transition correspond à l'événement (e7);

3.1.2.5.4 Etat 4 – "User Interaction" (Interaction avec l'utilisateur)

Dans cet état, l'entité SCF demande à l'entité SRF d'effectuer une interaction avec un utilisateur (par exemple, recueillir des informations supplémentaires et/ou diffuser des messages vocaux). Lorsqu'une interaction a eu lieu, l'entité SCF peut demander à l'entité SSF de déconnecter le support entre SSF et SRF. Elle peut aussi envoyer une opération user interaction (Interaction avec l'utilisateur) contenant une réponse, ce qui permettra à l'entité SRF déclencheuse de libérer.

En passant à cet état, le modèle SCSM réinitialise le temporisateur $T_{SCF-SSF}$ (s'il est utilisé).

Un événement ci-après provoque la sortie de cet état:

- (e11) Continue_SCF_Processing – Dans ce cas, l'entité SCF a obtenu de la part de l'entité SRF toutes les informations nécessaires pour charger l'entité SSF de faire aboutir l'appel. Cet événement provoque le passage à l'état 2, "Préparer les instructions SSF".

Afin d'examiner plus précisément le traitement de cet état, une machine FSM distincte est détaillée à la Figure 22A.

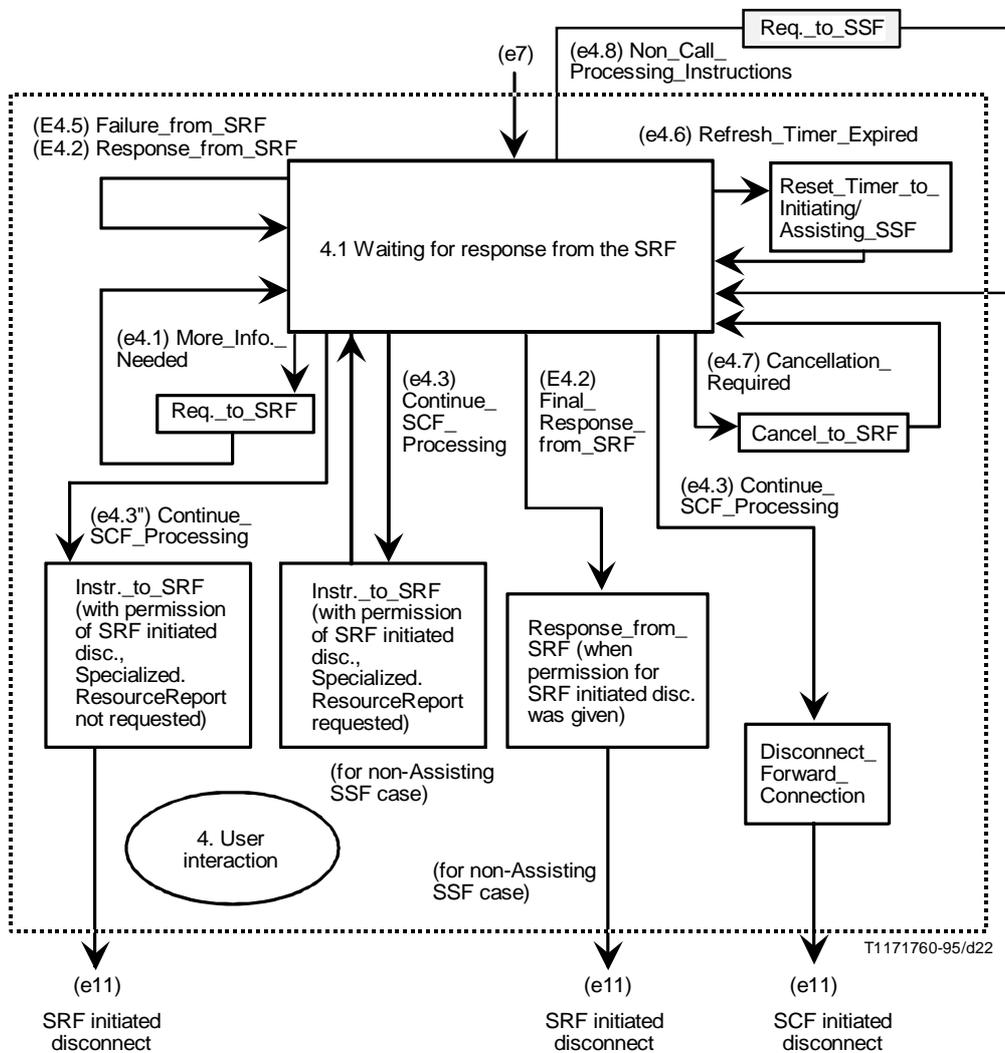


FIGURE 22A/Q.1218

Machine FSM représentative de l'état 4

3.1.2.5.4.1 Etat 4.1 – "Attente d'une réponse en provenance de l'entité SRF"

Dans cet état, l'entité SCF attend la réponse précédemment envoyée et analyse cette réponse. On considère que les événements ci-après sont dans cet état:

- (e4.1) *More_Information_Needed* – Se traduit par l'émission d'une autre opération vers l'entité SRF; il provoque le retour à l'état 4.1;
- (E4.2) *Response_from_SRF* – Événement externe déclenché par la réception de l'opération **SpecializedResourceReport** ou d'un renvoi de résultat par l'opération **PromptAndCollectUserInformation**. A la réception de cette opération, le modèle SCSM revient au même état;
- (E4.2) *Final_Response_from_SRF* (Réponse finale en provenance de l'entité SRF) – Événement externe provoqué par la réception de l'opération **SpecializedResourceReport**, comme le demande l'entité SCF, en réponse à l'opération préalable **PlayAnnouncement** ou Renvoi de résultat suivant l'opération **PromptAndCollectUserInformation** avec autorisation de la déconnexion déclenchée par l'entité SRF. Dans le cas du relais de l'entité SSF déclencheuse et dans celui de l'association directe SCF-SRF, à la réception de cet événement, le modèle SCSM transite vers l'état 2 "Préparer les instructions SSF". Cet événement correspond à l'événement (e11);
- (e4.3) *Continue_SCF_Processing* (Continuer le traitement de l'entité SCF) – Événement interne qui se produit quand le modèle SCSM termine l'interaction avec l'utilisateur et demande la déconnexion de la connexion support entre l'entité déclencheuse SSF et l'entité SRF, au moyen de la déconnexion déclenchée vers l'entité SCF. Dans ce cas, le modèle SCSM envoie l'opération **DisconnectForwardConnection** à l'entité SSF déclencheuse et transite vers l'état 2 "Préparer les instructions SSF". Cet événement correspond à l'événement (e11);
- (e4.3') *Continue_SCF_Processing* (Continuer le traitement de l'entité SCF) – Événement interne qui se produit quand le modèle SCSM termine l'interaction avec l'utilisateur et demande la déconnexion de la connexion support entre l'entité SSF déclencheuse et l'entité SRF au moyen de la déconnexion déclenchée par l'entité SRF, tandis qu'un retour de l'opération **disconnectSpecializedResourceReport** vers l'entité SCF est demandé au cas où une annonce est achevée. Dans ce cas, le modèle SCSM envoie l'opération **PlayAnnouncement** (contenant une demande de renvoi d'une opération **SpecializedResourceReport** comme indication de l'achèvement de l'opération) ou une opération **PromptAndCollectUserInformation** vers l'entité SRF avec autorisation de la déconnexion déclenchée par l'entité SRF. Dans le cas d'une entité SSF assistante, la déconnexion déclenchée par l'entité SRF ne peut pas être utilisée. Dans ce cas, le modèle SCSM retourne vers le même état;
- (e4.3'') *Continue_SCF_Processing* (Continuer le traitement de l'entité SCF) – Événement interne qui se produit quand le modèle SCSM achève l'interaction avec l'utilisateur et demande la déconnexion de la connexion support entre l'entité SRF et l'entité SSF déclencheuse au moyen de la déconnexion déclenchée par l'entité SRF, lorsqu'il n'est pas demandé de retourner vers l'entité SCF d'opération **SpecializedResourceReport** quand une annonce est achevée. Dans ce cas, le modèle SCSM envoie l'opération **PlayAnnouncement** (ne contenant pas de demande de renvoi de l'opération **SpecializedResourceReport** comme indication de l'achèvement de l'opération) en autorisant la déconnexion déclenchée par l'entité SRF vers l'entité SRF. Dans le cas de l'entité SSF assistante, la déconnexion déclenchée par l'entité SRF ne peut pas être utilisée. Dans ce cas, le modèle SCSM retourne vers l'état 2, "Préparer les instructions SSF". Cet événement correspond à l'événement (e11);
- (E.4.5) *Failure_from_SRF* (Echec à partir de l'entité SRF) – Événement externe provoqué par la réception du renvoi d'erreur pour les opérations **PlayAnnouncement** ou **PromptAndCollectUserInformation**. Dans ce cas, le modèle SCSM ne change pas d'état;
- (e4.6) *Refresh_Timer_Expired* (Relance de la temporisation après son expiration) (FACULTATIF: voir 3.1.2.5) – Événement interne qui se produit à la fin de la temporisation d'application $T_{SCF-SSF}$. Dans ce cas, le modèle SCSM transmet l'opération **ResetTimer** vers l'entité SSF déclencheuse/assistante et retourne vers le même état;
- (e4.7) *Cancellation_Required* – Événement interne qui se produit quand le modèle SCSM annule l'opération antérieure **PlayAnnouncement** ou **PromptAndCollectUserInformation**. Dans ce cas, le modèle SCSM envoie l'opération **Cancel** à l'entité SSF assistante (cas du relais SSF) ou l'entité SRF (liaison directe SCF-SRF), et revient au même état.

Il convient de noter que la connexion support entre l'entité SSF et l'entité SRF est rompue quand le modèle SCSM sort de cet état.

- (e4.8) **Non_Call_Processing_Instructions** – Événement interne provoqué par la logique de service quand il est nécessaire d'envoyer cette opération à l'entité SSF. Dans ce cas, une des opérations suivantes est présentée à l'entité SSF:
 - **ApplyCharging;**
 - **FurnishChargingInformation;**
 - **RequestNotificationChargingEvent;**
 - **SendChargingInformation.**

Dans ce cas, le modèle SCSM reste dans le même état.

3.1.2.5.5 Etats associés à l'état SDF

L'interaction avec l'entité SDF est possible à partir de tout état de l'entité SCF. Dans les paragraphes suivants, les états associés à l'entité SDF sont spécifiés. Le modèle décrit la relation entre une entité SCF et une entité SDF. Si une entité SCF a besoin d'une autre entité SDF, une nouvelle entité FSM devrait être créée.

Dans ce qui suit, les états et les événements sont énumérés indépendamment du reste du modèle SCSM; la Figure 22B illustre le propos.

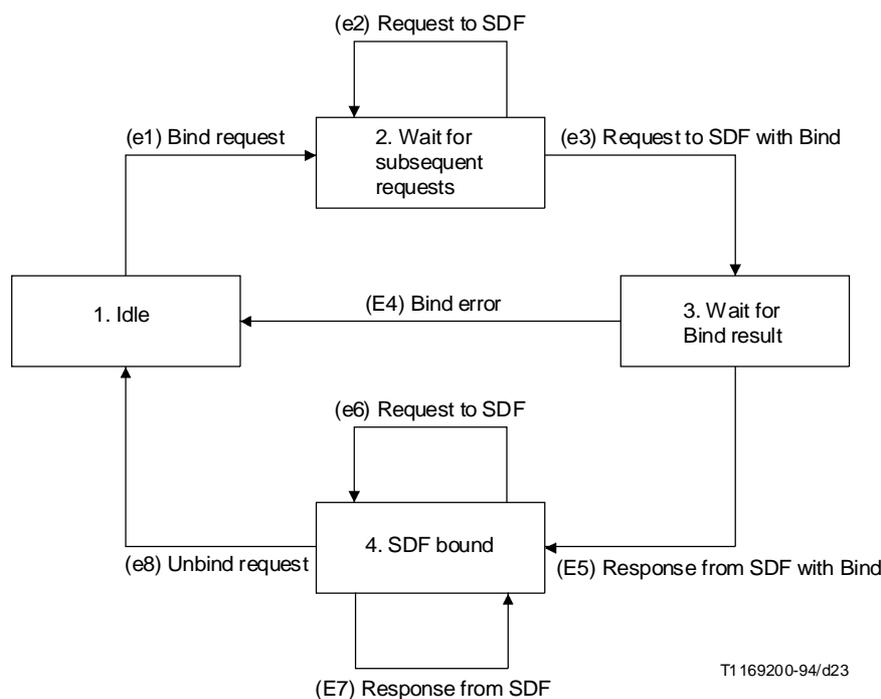


FIGURE 22B/Q.1218
Les états associés à l'entité SDF

3.1.2.5.5.1 Etat 1 – "Idle" (repos)

Ceci est un sous-état de n'importe quel état avant qu'une demande ait été présentée à l'entité SDF. On considère que l'événement ci-après appartient à cet état:

- (e1) **Bind_Request** (demande de rattachement) – Événement interne déclenché par le besoin de logique de service afin de créer une association avec une entité SDF pour commencer à accéder à des données. Cet événement provoque le passage à l'état 2, **Wait for subsequent requests**.

3.1.2.5.5.2 Etat 2 – "Wait for subsequent requests" (Attente des demandes subséquentes)

Dans cet état, il est prévu que des opérations subséquentes soient envoyées (dans le même message) que l'opération **Bind** de rattachement (dans le même message) à l'entité SDF. On considère que les deux événements ci-après appartiennent à cet état:

- (e2) Request_to_SDF – Événement interne provoqué par la réception d'une opération. Cette opération est mise dans un tampon jusqu'à la réception d'un délimiteur (ou jusqu'à ce que la temporisation prenne fin). Le modèle SCSM reste dans le même état;
- (e3) Request_to_SDF_with_Bind (Demande de rattachement à l'entité SDF) – Événement interne provoqué par la réception d'un délimiteur, qui indique la réception de la dernière opération émise. A la réception du délimiteur, un message contenant l'argument de l'opération **Bind** de rattachement et, le cas échéant, les arguments des autres opérations, est envoyé à l'entité SDF. Cet événement provoque la sortie de cet état et le passage à l'état 3, **Wait for Bind result**.

3.1.2.5.5.3 Etat 3 – "Wait for Bind Result" (Attente du résultat du rattachement)

Dans cet état, l'entité SCF attend la réponse de l'entité SDF. On considère que deux événements appartiennent à cet état:

- (E4) Bind Error – Événement externe provoqué par la réception d'une erreur par l'opération de **rattachement** préalablement présentée à l'entité SDF. Cet événement provoque une transition de cet état vers l'état 1, **Idle**;
- (E5) Response_from_SDF_with_Bind – (Réponse de l'entité SDF à la demande de rattachement) – Événement externe provoqué par la réception d'un résultat de **rattachement** complété par les éventuelles réponses apportées à d'autres opérations précédemment présentées à l'entité SDF (le cas échéant). Cet événement provoque le passage à l'état 4, **rattachement SDF**.

3.1.2.5.5.4 Etat 4 – "SDF Bound" (Rattachement SDF)

Dans cet état, l'entité SCF a créé un accès authentifié vers l'entité SDF, et attend que des demandes à l'entité SDF émanent de la logique de service ou que des réponses soient apportées aux opérations précédemment présentées à l'entité SDF. On considère que trois événements appartiennent à cet état:

- (e6) Request_to_SDF – Événement interne provoqué par le besoin de la logique de service d'accéder aux données de l'entité SDF. Le modèle SCSM reste dans le même état;
- (E7) Response_from_SDF (Réponse de l'entité SDF) – Événement externe, provoqué par la réception des réponses apportées aux opérations précédemment présentées à l'entité SDF. Le modèle SCSM reste dans le même état;
- (e8) Unbind_request (Demande de détachement) – Événement interne, provoqué par le besoin de la logique de service de mettre fin à l'accès authentifié vers l'entité SDF. Cet événement provoque une transition vers l'état 1, **Idle** (repos).

Outre le modèle ci-dessus, une description SDL du modèle SCSM est indiquée à l'Annexe B.

3.1.3 Procédures relatives à l'entité application SRF

3.1.3.1 Considérations générales

Ce paragraphe contient la définition des procédures relatives à l'entité application de l'entité SRF se rapportant à l'interface SRF-SCF. Ces procédures font appel au système de signalisation n° 7; on peut aussi utiliser d'autres systèmes de signalisation.

D'autres capacités peuvent être prises en charge de manière spécifique à la réalisation dans les entités physiques IP, SSP ou SN.

L'entité application, conformément à l'architecture définie dans les Recommandations Q.700, Q.771 et Q.1400, comprend le sous-système TCAP et un ou plusieurs éléments ASE appelés utilisateurs-TC. Les paragraphes ci-après définissent les règles applicables à l'élément de service d'application utilisateur-TC, aux fonctions SACF et MACF, qui assurent l'interface avec le sous-système TCAP en utilisant les primitives spécifiées dans la Recommandation Q.771.

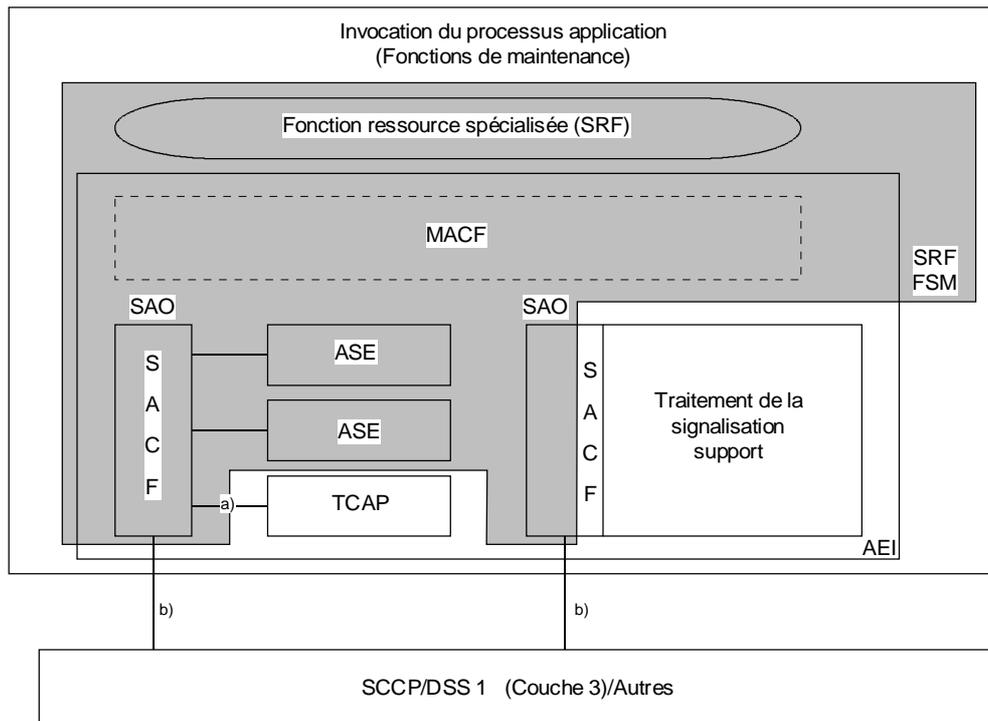
La procédure peut également être utilisée avec d'autres systèmes de signalisation prenant en charge les structures définies de la couche application.

Au cas où les interprétations pour les procédures relatives à l'entité d'application définies dans ce qui suit divergeraient des procédures détaillées et des règles d'utilisation des services TCAP, il conviendrait de se référer aux dispositions et aux règles dont les détails figurent en 3.3 et 3.4.

3.1.3.2 Modèle et interfaces

Le modèle fonctionnel de la fonction AE-SRF est représenté à la Figure 23; les éléments ASE assurent l'interface avec le sous-système TCAP (pour communiquer avec l'entité SCF) et également l'interface avec les fonctions de maintenance. Le domaine d'application de la présente Recommandation est limité à la zone ombrée de la Figure 23.

Les interfaces représentées à la Figure 23 utilisent les primitives ASE utilisateur-TC spécifiées dans la Recommandation Q.771 [interface (1)] et les primitives de couche réseau (N-) spécifiées dans la Recommandation Q.711 [interface (2)]. Les opérations et les paramètres associés au protocole d'application du réseau intelligent (INAP) sont définis à l'article 2.



T1146860-92/d24

- AEI Invocation de l'entité application (*application entity invocation*)
- SRF Fonction ressource spécialisée (*specialized resource function*)
- FSM Machine à états finis (*finite state machine*)
- MACF Fonction de commande d'association multiple (*multiple association control function*)
- SACF Fonction de commande d'association unique (*single association control function*)
- SAO Objet d'association unique (*single association object*)

a) Primitives TC ou primitives Q.932.

b) Primitives N.

NOTE – La machine SRF-FSM peut comporter plusieurs machines à états finis.

FIGURE 23/Q.1218

Modèle fonctionnel de l'entité d'application SRF

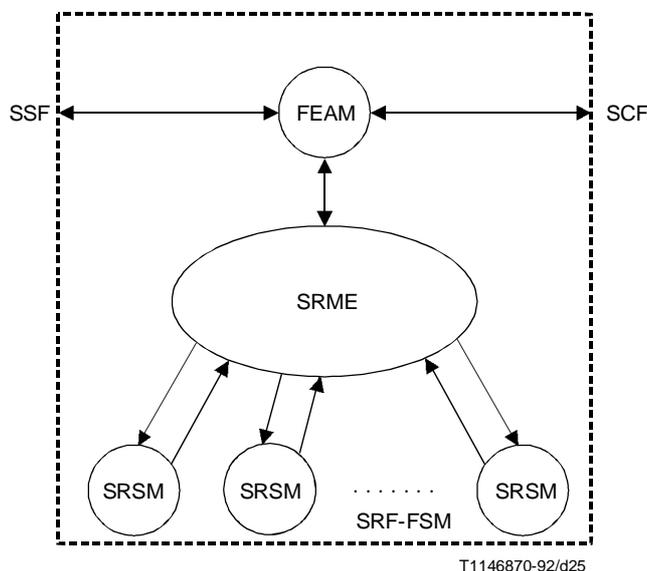
3.1.3.3 Relation entre la machine SRF-FSM et les fonctions de maintenance/gestion de connexion support

L'interface primitive entre le SRF-FSM et les fonctions de maintenance est une interface interne et ne fait pas l'objet de la normalisation dans l'ensemble de capacités CS-1 du réseau intelligent.

La relation entre la gestion de connexion support et la machine SRF peut être décrite comme suit pour le cas d'un appel déclenché par l'entité SSF. Lorsqu'une tentative d'appel est déclenchée par l'entité SSF, une instance de machine SRF-FSM est créée.

La machine SRF-FSM gère l'interaction avec la machine SCF-FSM et SSF-FSM.

Les fonctions de gestion relatives à l'exécution de l'opération provenant de l'entité SCF sont assurées par l'entité de gestion SRF (SRME) (*SRF management entity*). L'entité SRME assure l'interface des différents modèles d'état appel SRF (SRSM) (*SRF call state models*) et le gestionnaire d'accès d'entité fonctionnelle (FEAM) (*functional entity access manager*). La Figure 24 représente la structure de la machine SRF-FSM.



- | | |
|------|--|
| FEAM | Gestionnaire d'accès aux entités fonctionnelles
(<i>functional entity access manager</i>) |
| SRME | Entité de gestion, SRF (<i>SRF management entity</i>) |
| SRSM | Modèle d'état d'appel SRF (<i>SRF call state model</i>) |

FIGURE 24/Q.1218

Structure de la machine SRF-FSM

Le modèle associe une machine à états finis (FSM) à chaque demande d'interaction initiale provenant de l'entité SCF. Ainsi, plusieurs demandes initiales peuvent être exécutées de manière parallèle et asynchrone par l'entité SRF, ce qui explique pourquoi il faut disposer d'une seule entité assurant les tâches de création, de demande et de maintenance des objets du modèle SRSM. Cette entité est appelée entité de gestion SRF (SRME) (*SRF management entity*). Outre les tâches ci-dessus, l'entité SRME maintient les dialogues avec les entités SCF et SSF pour toutes les instances du modèle SCSM. En particulier l'entité SRME:

- 1) interprète les messages reçus en provenance des autres entités fonctionnelles et les traduit en événements SRSM correspondants;
- 2) traduit les sorties du modèle SRSM en messages correspondants à l'intention des autres entités fonctionnelles.

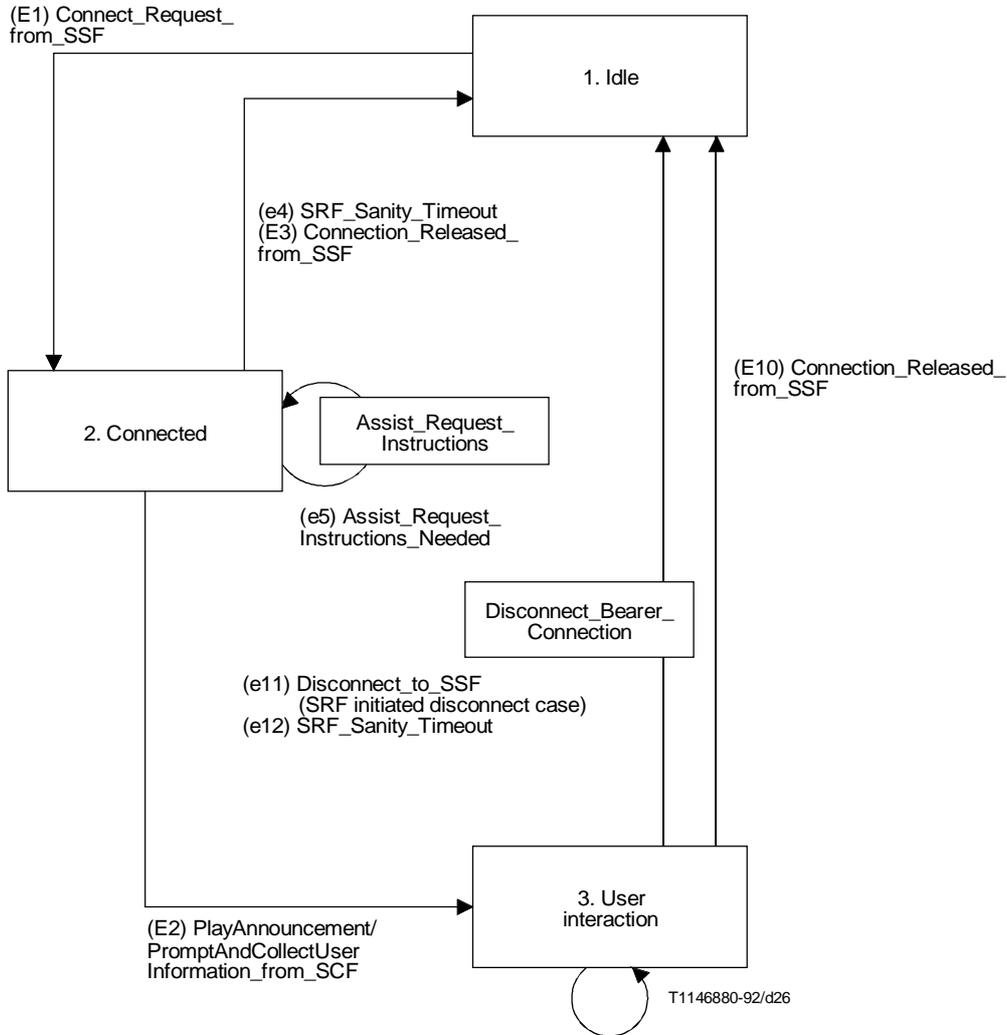
NOTE – Une telle demande de l'entité SCF est exécutée par l'entité SCSM lorsqu'elle se trouve dans son état 4.

Enfin, le gestionnaire d'accès aux entités fonctionnelles (FEAM) libère l'entité SRME des fonctions d'interface de niveau inférieur. Le gestionnaire d'accès aux entités fonctionnelles (FEAM) inclut:

- 1) l'établissement et le maintien des interfaces vers les entités SSF et SCF;
- 2) la retransmission (et au besoin la mise en file d'attente) des messages provenant des entités SSF et SCF vers l'entité SRME;
- 3) le formatage, la mise en file d'attente (au besoin), et l'envoi des messages provenant de l'entité SRME vers les entités SSF et SCF.

3.1.3.4 Le modèle SRSM

Le modèle SRSM est représenté à la Figure 25. Dans ce qui suit, chaque état sera décrit dans un paragraphe distinct ainsi que les événements qui provoquent la sortie de cet état. Enfin, les sorties seront représentées dans des rectangles plus petits que les états; à l'inverse des états et des événements, les sorties ne sont pas numérotées.



- (E5) PlayAnnouncement/PromptAndCollectUserInformation_from_SCF
- (E6) Cancel_from_SCF
- (e7) SRF_Report_to_SCF
- (e8) PlayAnnouncement/PromptAndCollectUserInformation_Cancelled_to_SCF
- (e9) Cancel_Error_to_SCF

FIGURE 25/Q.1218

Le modèle SRSM

Chaque état est analysé dans les paragraphes suivants. Les règles générales applicables à plusieurs états sont analysées ici.

Une composante ou une séquence de composantes reçues à l'intérieur d'un ou plusieurs messages du sous-système TCAP peuvent comprendre une ou plusieurs opérations, et le traitement se déroule comme suit:

- le modèle SRSM traite les opérations dans l'ordre où elles sont reçues;

- le modèle SRSM examine les opérations successives de la séquence. Lorsqu'on rencontre une opération Cancel (Annulation d'annonce) (pour une opération PlayAnnouncement ou une opération PromptAndCollectUserInformation) dans la séquence dans l'état Interaction avec l'utilisateur, le modèle SRSM les exécute immédiatement. Dans tous les autres cas, le modèle SRSM met les opérations en file d'attente et attend un événement (un tel événement pourrait être l'exécution d'une opération en cours, ou la réception d'un événement externe);
- en cas d'erreur de traitement de l'une des opérations dans la séquence, la machine SRF-FSM traite l'erreur (voir ci-après) et annule toutes les opérations restantes de la séquence;
- si une opération n'est pas comprise ou se trouve hors contexte (c'est-à-dire qu'elle viole les règles applicables à la fonction SACF définies par le modèle SRSM) tel que décrit ci-dessus, la machine SRF-FSM traite l'erreur selon les règles indiquées au 3.4.2 (en utilisant TC-U-REJECT ou l'erreur d'exécution UnexpectedComponentSequence).

Dans tout état, s'il y a une erreur dans une opération reçue, les fonctions d'exploitation maintenance sont informées. Généralement, le modèle SRSM reste dans le même état dans lequel il se trouvait lorsque les opérations erronées auront été reçues, mais cependant des relevés différents de défauts sont possibles dans des cas particuliers comme le prévoit 3.2; selon la classe de l'opération, l'erreur pourrait être signalée par l'entité SRF à l'entité SCF en utilisant la composante appropriée (voir la Recommandation Q.774).

Dans un état quelconque, si le dialogue avec l'entité SCF (liaison directe SCF-SRF) a pris fin, le SRSM retourne à l'état Idle (repos) après s'être assuré que toute les ressources affectées aux dialogues ont été libérées. L'entité SRF restera connectée à l'entité SSF tant que ses opérations PlayAnnouncement seront actives ou mises en tampon. Les ressources affectées à l'appel seront libérées quand toutes les annonces auront été effectuées ou quand l'entité SSF traitera la déconnexion du support (c'est-à-dire libération de l'appel).

Dans un état quelconque (sauf l'état Idle), si l'entité SSF déconnecte le support avec l'entité SRF avant que l'entité SRF ait terminé l'interaction d'utilisateur, le modèle SRSM libère l'appel et s'assure que toutes les ressources affectées à l'appel ont été retirées. Il passe ensuite à l'état Idle.

Le modèle SRSM a un temporisateur d'application T_{SRF} dont l'objet est d'éviter d'avoir une durée de mise en attente d'appel trop importante. Ce temporisateur est armé lorsque l'entité SRF envoie un message support Setup Response à l'entité SSF (cas relais SSF) ou l'opération AssistRequestInstructions (cas SCF-SRF direct). Ce temporisateur est arrêté quand une demande est reçue de l'entité SCF. L'entité SRF peut réinitialiser le temporisateur T_{SRF} sur transmission de l'opération SpecializedResourceReport ou du retour d'erreur pour l'opération PromptAndCollectUserInformation lorsqu'il n'y a pas d'opération du type user interaction (interaction avec l'utilisateur) mise en file d'attente. A la fin de la temporisation T_{SRF} , le modèle SRSM passe à l'état Idle en s'assurant que toutes les ressources affectées à l'appel ont été libérées.

3.1.3.4.1 Etat 1 – "Idle"

L'état Idle est l'état dans lequel on se trouve avant ou à la fin d'une instance d'interaction d'utilisateur. Le passage à cet état s'effectue suite aux événements E3, e4, E10, e11 et e12. La sortie est consécutive à l'événement E1.

- (E1) Connect_Request_from SSF – Cet événement correspond au message de demande de connexion sémaphore support provenant de l'entité SSF. Les détails de la machine à l'état de signalisation support associés à l'établissement de la connexion n'affectent pas la machine FSM. Le modèle SRSM passe à l'état "Connecté";
- (E3) Connection_Released_from_SSF – Cet événement a lieu lorsque le modèle SRSM reçoit un message de libération émis par l'entité SSF dans l'état "connecté". Le modèle SRSM passe à l'état "Idle";
- (e4) SRF_Sanity_Timeout – Cet événement se produit lorsque le modèle SRSM s'est trouvé dans l'état "connecté" pendant une durée définie par l'exploitant de réseau (temporisateur T_{SRF}) sans avoir d'opération PlayAnnouncement/PromptAndCollectUserInformation à exécuter. L'entité SRF déclenche une séquence de déconnexion du canal support vers l'entité SSF en utilisant le système de signalisation de canal support vers l'entité SSF en utilisant le système de signalisation sur le canal support applicable. Le modèle SRSM passe à l'état "Idle";
- (E10) Connection_Released_from_SSF – Cet événement a lieu, lorsque le modèle SRSM reçoit un message de libération émis par l'entité SSF dans l'état d'interaction. Le modèle SRSM passe à l'état "Idle";

- (e11) Disconnect_to_SSF – Cet événement a lieu lorsque l'entité SCF a validé une déconnexion déclenchée par l'entité SRF au moyen de la dernière opération PlayAnnouncement/PromptAndCollectUserInformation à partir de l'entité SCF (E2) ou (E5) avec le paramètre disconnectFromIPForbidden. Le modèle SRSM déclenche une séquence de déconnexion de canal support vers l'entité SSF en utilisant le système de signalisation de canal support applicable après avoir envoyé la dernière opération SpecializedResourceReport vers l'entité SCF (e7). Le modèle SRSM passe à l'état "Idle";
- (e12) SRF_Sanity_Timeout – Cet événement se produit lorsque le modèle SRSM a été dans l'état User interaction pendant une durée définie par l'opérateur de réseau (temporisateur T_{SRF}) sans avoir d'opération PlayAnnouncement/PromptAndCollectUserInformation à exécuter. L'entité SRF déclenche une séquence de déconnexion de canal support vers l'entité SSF en utilisant le système de signalisation de canal support applicable. Le modèle SRSM passe à l'état "Idle".

3.1.3.4.2 Etat 2 – "Connected" (connecté)

Dans cet état correspond le modèle SRSM qui se trouve dans la situation suivante: un canal support a été établi entre un utilisateur et l'entité SRF mais l'opération initiale PlayAnnouncement/ PromptAndCollectUserInformation n'a pas encore été reçue (par exemple, lorsque l'on utilise les procédures EstablishTemporaryConnection). La méthode utilisée pour offrir ce canal support n'intéresse pas la machine FSM.

- (E1) Connect_Request_from_SSF – Cet événement correspond à un message de demande de connexion de signalisation support émise par l'entité SSF dans l'état Idle. Les détails de la machine à état de signalisation support associée à l'établissement de la connexion n'intéressent pas la machine SRF-FSM. Le modèle SRSM passe à l'état "Connecté";
- (E2) PlayAnnouncement/PromptAndCollectUserInformation_from_SCF – Cet événement a lieu lorsque la première opération (ou les premières opérations) PlayAnnouncement ou PromptAndCollectUserInformation émise(s) par l'entité SCF est (ou sont) reçue(s). Le modèle SRSM passe à l'état "User interaction" (Interaction avec l'utilisateur);
- (E3) Connection_Released_from_SSF – Cet événement a lieu lorsque l'entité SRF reçoit un message de libération provenant de l'entité SSF. Le modèle SRSM passe à l'état "Idle";
- (e4) SRF_Sanity_Timeout – Cet événement a lieu lorsque le modèle SRSM a été connecté pendant une durée définie par l'exploitant de réseau (temporisateur T_{SRF}) sans avoir d'opération PlayAnnouncement/PromptAndCollectUserInformation à exécuter. Le modèle SRSM déclenche une séquence de déconnexion du canal support vers l'entité SSF en utilisant le système de signalisation de canal support applicable. Le modèle SRSM passe à l'état "Idle";
- (e5) Assist_Request_Instructions_Needed – Cet événement se produit lorsque l'opération AssistRequestInstructions est envoyée du modèle SRSM vers l'entité SCF en l'absence d'événement (E2) PlayAnnouncement/PromptAndCollectUserInformation déclenché par la présence d'une opération PlayAnnouncement/PromptAndCollectUserInformation concaténée avec la demande d'établissement émise par l'entité SSF (E1) (cas d'une relation directe SCF-SRF). Cet événement ne produit pas de modification d'état.

3.1.3.4.3 Etat 3 – "User interaction" (Interaction avec l'utilisateur)

L'état User interaction indique qu'une communication a lieu entre l'utilisateur et l'entité SRF par l'intermédiaire du canal support établi dans l'état "Connecté". Le passage à cet état résulte de l'événement E2. La sortie de cet état résulte des événements E10, e11 et e12. Les événements E5, E6, e7, e8 et e9 ne provoquent pas de modification d'état. L'événement E5 représente également des opérations supplémentaires PlayAnnouncement/PromptAndCollectUserInformation qui sont mises en tampon comme indiqué dans les procédures.

- (E2) et (E5) PlayAnnouncement/PromptAndCollectUserInformation_from_SCF – Cet événement est déclenché par la réception d'une ou plusieurs opérations initiales ou suivantes PlayAnnouncement ou PromptAndCollectUserInformation en provenance de l'entité SCF. Le modèle SRSM passe à l'état "User interaction" sur le premier événement (E2). Il reste dans l'état "User interaction" pour les états (E5) suivants;
- (E6) Cancel_from_SCF (for PlayAnnouncement/PromptAndCollectUserInformation) – Cet événement est déclenché par la réception de l'opération correspondante PlayAnnouncement ou PromptAndCollectUserInformation en provenance de l'entité SCF. L'interaction indiquée prend fin si elle est en cours d'exécution, sinon elle est supprimée dans le tampon. Le modèle SRSM reste dans l'état "User interaction";

- (e7) SRF_Report_to_SCF – Cet événement a lieu lorsqu'une opération SpecializedResourceReport ou une information de réponse PromptAndCollectUser Information est envoyée vers l'entité SCF. Le modèle SRSM reste à l'état "User interaction";
- (e8) PlayAnnouncement/PromptAndCollectUserInformation_Cancelled_to_SCF – Cet événement a lieu quand une erreur PlayAnnouncement/PromptAndCollectUserInformation causée par l'opération Cancel (pour PlayAnnouncement ou PromptAndCollect UserInformation) est envoyée vers l'entité SCF. Cet événement représente l'annulation effective d'une opération PlayAnnouncement/-PromptAndCollectUserInformation active ou mise en tampon. Le modèle SRSM reste dans l'état "User interaction";
- (e9) Cancel_Error_to_SCF – Cet événement a lieu lorsque l'erreur PlayAnnouncement/PromptAndCollectUserInformation causée par l'opération Cancel (pour l'opération PlayAnnouncement ou PromptAndCollectUserInformation) est envoyée vers l'entité SCF. Cet événement représente l'échec de l'annulation de l'opération PlayAnnouncement/PromptAndCollectUserInformation. Le modèle SRSM reste dans l'état "User interaction";
- (E10) Connection_Released_from_SSF – Cet événement a lieu quand le modèle SRSM reçoit un message de libération en provenance de l'entité SSF. Le modèle SRSM passe à l'état "Idle";
- (e11) Disconnect_to_SSF – Cet événement se produit quand l'entité SCF a autorisé une déconnexion déclenchée par l'entité SRF avec la dernière opération PlayAnnouncement/-PromptAndCollectUserInformation provenant de l'entité SCF (E2) ou (E5). Le modèle SRSM déclenche une séquence de déconnexion de canal support vers l'entité SSF en utilisant le système de signalisation de canal support applicable après envoi de la dernière opération SpecializedResourceReport ou un message retour de l'opération PromptAndCollectUserInformation à l'entité SCF. Le modèle SRSM passe à l'état "Idle";
- (e12) SRF_Sanity_Timeout – Cet événement se produit quand le modèle SRSM s'est trouvé dans un état user interaction pendant une période de temps définie par l'exploitant de réseau (temporisateur T_{SRF}) sans avoir à exécuter une opération PlayAnnouncement/PromptAndCollectUserInformation. Le modèle SRSM déclenche une séquence de déconnexion de canal support vers l'entité SSF en utilisant le système de signalisation de canal support applicable. Le modèle SRSM passe à l'état "Idle";

Outre ces transitions explicitement indiquées, l'échec d'une connexion support utilisateur-entité SRF provoquera le passage du modèle SRSM d'un état quelconque à l'état Idle. Pour des raisons de clarté, ces transitions ne sont pas représentées à la Figure 25.

3.1.3.5 Exemples de procédures de commande SRF

Le présent paragraphe contient une description détaillée des procédures SRF. Les diagrammes fléchés sont utilisés pour décrire les états connexion, interaction avec l'utilisateur et déconnexion.

Les procédures de commande SRF reposent sur divers systèmes d'attribution physiques des entités SRF. Ces procédures sont décrites dans le présent paragraphe conformément aux exemples de scénarios physiques d'architecture de protocole décrite en 0.2.

Les procédures service assist (assistance au service) et service handoff (service sans assistance) fondées sur les scénarios physiques sont également décrites à titre d'exemple dans le présent paragraphe.

Il convient de noter que dans le présent paragraphe, les messages de signalisation de commande de connexion support sont utilisés aux fins d'explication mais ne font pas l'objet de normalisation dans la présente Recommandation. Les termes utilisés pour les messages de signalisation de commande de connexion support n'ont qu'une signification fonctionnelle.

3.1.3.5.1 Procédures de connexion SRF

3.1.3.5.1.1 Procédures physiques de connexion SRF

Pour couvrir les différents scénarios il est nécessaire de disposer de plusieurs procédures. Les cas couverts sont décrits ci-après et illustrés à la Figure 26.

- i) le périphérique intelligent (IP) est intégré au (*service switching point*) point de commutation de service (SSP), ou directement associé à ce point, qui interagit avec le (*service control point*) point de commande de service (SCP), mais les opérations effectuées par ce point SCP en direction du périphérique intelligent sont relayées par l'intermédiaire du point SSP qui assure toutes les conversions de protocole nécessaires;
- ii) le périphérique intelligent est directement associé au point SSP qui interagit avec le point SCP, mais les opérations effectuées par le point SCP en direction du périphérique intelligent sont envoyées directement à ce périphérique sans être relayées par le point SSP;

- iii) le périphérique intelligent est intégré à un autre point SSP, ou directement rattaché à un autre point SSP autre que celui qui interagit avec le point SCP mais les opérations effectuées par le SCP à l'intention du périphérique intelligent sont relayées par l'intermédiaire d'un deuxième point SSP (appelé méthode "Assist"), et à la fin de l'interaction d'utilisateur, la commande est transférée au premier point SSP;
- iv) le périphérique intelligent est directement attaché à un nœud différent du point SSP qui interagit avec le point SCP mais les opérations effectuées par le point SCP en direction du périphérique intelligent sont envoyées directement à ce périphérique sans faire intervenir le point SSP relais (appelé la méthode "Assist", mais avec une variante en ce qui concerne la connectivité physique des entités intervenantes) vers la fin de l'interaction d'utilisateur, la commande transférée au premier point SSP;
- v) le périphérique intelligent est attaché à un autre point SSP et à la fin de l'interaction d'utilisateur, la commande de l'appel est maintenue en ce point SSP (appelé méthode "Handoff" ou méthode à passation).

Dans chacun des cas ci-dessus, les opérations entre les points SCP et SSP peuvent s'appuyer sur le sous-système TCAP du système de signalisation n° 7; la messagerie entre le point SSP et le périphérique intelligent lorsque le point SSP assure le relais peut faire appel au système DSS 1 en utilisant l'élément d'information Facility (dans ce cas, le point SSP devra effectuer la conversion de protocole du sous-système TCAP du système de signalisation n° 7 vers l'élément d'information Facility du système DSS 1 pour les opérations et les réponses qu'il relaie entre le point SCP et le périphérique intelligent); la messagerie directe entre le point SCP et le périphérique intelligent peut être basée sur le sous-système TCAP du système de signalisation n° 7; la signalisation de commande support peut être n'importe quel système.

Chaque scénario va être maintenant examiné en utilisant des diagrammes à flèche.

Le cas i) est illustré à la Figure 27. Il convient de noter que lorsque le périphérique intelligent est intégré au point SSP, les activités internes du nœud peuvent toujours être modélisées de la sorte, mais les détails sur la façon de procéder sont laissés au choix du réalisateur. Pour cette méthode, il n'est pas nécessaire que le point SCP fasse la distinction entre périphériques intelligents intégré ou externe, mais directement connectés. Voir également une remarque sur la possibilité de concaténation de la première opération d'interaction d'utilisateur avec l'opération ConnectToResource examinée dans le paragraphe traitant de l'interaction d'utilisateur ci-dessus. L'établissement de la relation SCF-SRF dans ce cas est implicite.

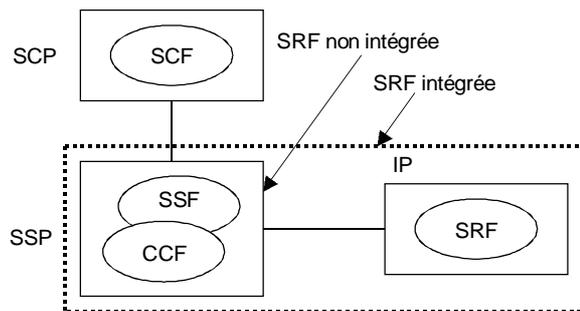
Dans le cas ii), il faut que le périphérique intelligent indique au point SCP qu'il est prêt à recevoir les opérations (voir la Figure 28). L'établissement de la relation SCF-SRF est explicite. Il convient de noter qu'il est nécessaire d'acheminer un identificateur de corrélation (Correlation ID) pour vérifier que la transaction établie entre le point SCP et le périphérique intelligent peut être corrélée avec l'établissement de la connexion support suite à l'opération précédente du SCP à l'intention du point SSP.

Le cas iii) nécessite l'ouverture d'une transaction avec le point SSP assistant afin qu'il puisse relayer les opérations du point SCP vers le périphérique intelligent (intégré ou externe). Après que la signalisation de commande du support ait atteint le point SSP assistant, il y a déclenchement sur l'identité de la facilité appelée et début d'une interaction avec le point SCP qui a demandé l'assistance. Il serait également possible d'effectuer un déclenchement sur d'autres éléments d'information tels l'adresse entrante. La signalisation de commande du support doit contenir les informations permettant d'identifier le point SCP demandant l'assistance, et un identificateur de corrélation. Cette information peut être dissimulée dans l'information d'adresse de manière telle que les systèmes de signalisation de type non-message puissent être utilisés pour établir la connexion avec le point SSP assistant. Après la réception de l'opération AssistRequestInstructions par le protocole SCP, les procédures sont les mêmes que dans le cas i). La Figure 29 indique le préambule de l'échange.

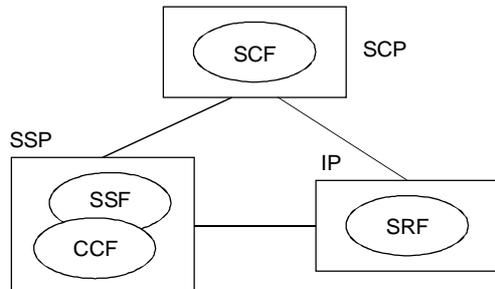
Le cas iv) ne nécessite pas l'établissement d'une deuxième transaction à partir du commutateur assistant, et il n'est donc pas nécessaire que ce soit un point SSP. Cela devient ensuite les préliminaires à la procédure représentée à la Figure 28 comme montré à la Figure 30.

Le cas v) nécessite simplement l'envoi d'une opération vers le premier point SSP pour acheminer l'appel au point SSP de passation, et donc la Figure 27 s'applique au point SSP auquel l'opération est passée (*handed-off*). Cela est représenté à la Figure 31. Il convient de noter que l'activité au point SSP auquel l'opération est passée représente une nouvelle interaction avec le point SCP et que l'opération «AssistRequestInstructions» est utilisée. Lorsque la signalisation de commande du support a atteint le point SSP assistant, il se déclenche sur l'identité de la facilité appelée, et déclenche une interaction avec le point SCP qui a demandé l'assistance. (Il sera également possible de se déclencher sur d'autres éléments d'information telle l'adresse entrante.) La signalisation de commande du support doit contenir l'information permettant d'identifier le point SCP qui demande l'assistance et un identificateur de corrélation. Cette information peut être dissimulée dans l'information d'adresse de manière telle que les systèmes de signalisation de type non-message puissent être utilisés pour établir la connexion support avec le point SSP assistant.

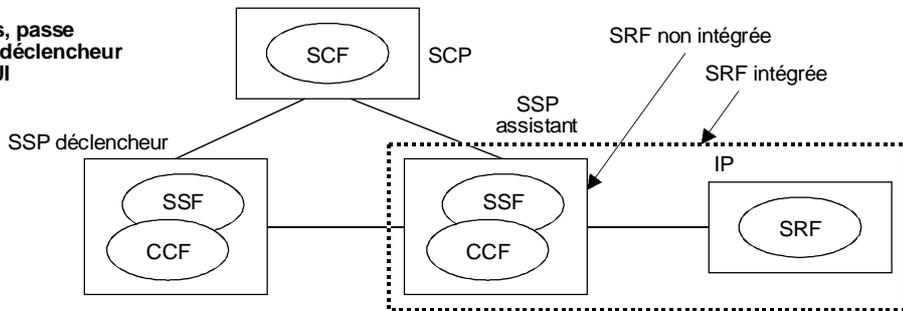
Cas i) SSF relais



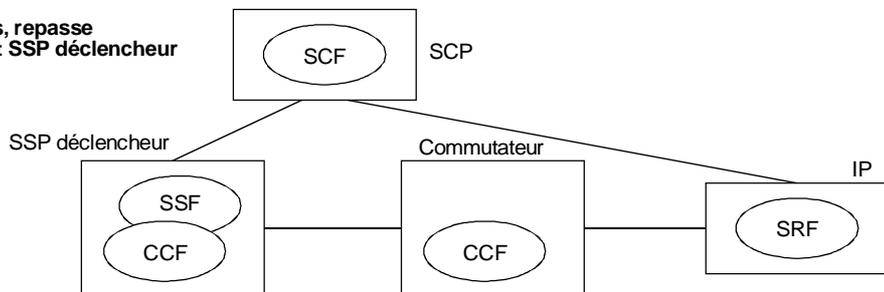
Cas ii) Trajet direct SCP-IP



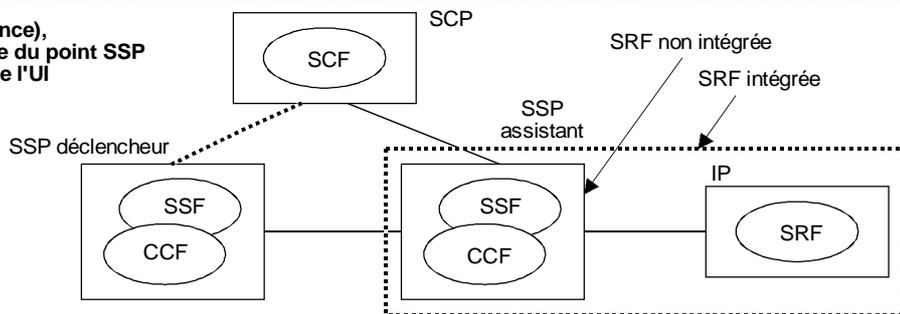
Cas iii) Assistance avec relais, passe la commande au SSP déclencheur après exécution de l'UI



Cas iv) Assistance sans relais, repasse la commande au point SSP déclencheur à la fin de l'UI



Cas v) Hand-off (sans assistance), maintient la commande du point SSP assistant après la fin de l'UI



T1146890-92/d27

FIGURE 26/Q.1218
Scénarios physiques

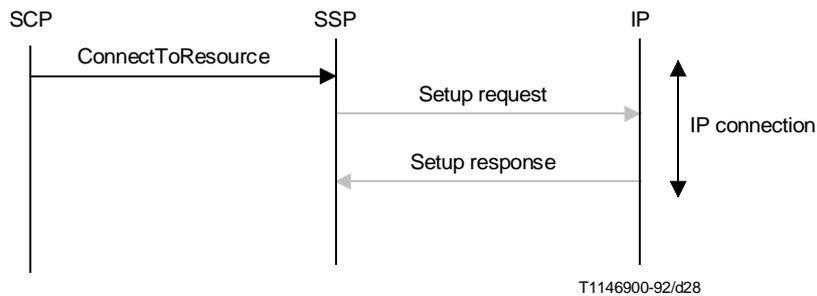


FIGURE 27/Q.1218
Connexion avec un périphérique intelligent (IP) intégré ou externe avec relais par le point SSP des opérations IP

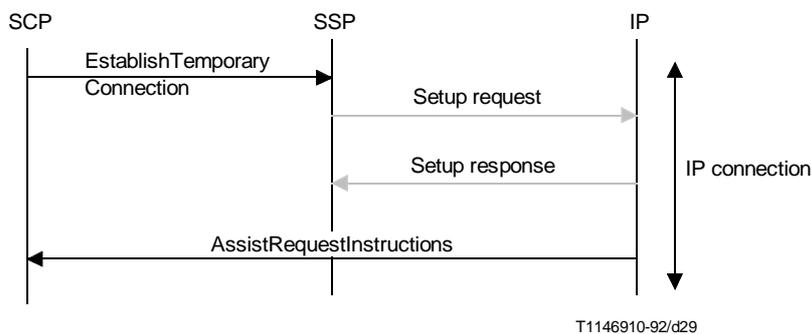


FIGURE 28/Q.1218
Connexion avec un périphérique intelligent (IP) avec une liaison directe avec le point SCP, le périphérique déclenche l'interaction avec le point SCP

NOTE – Les lignes noires indiquent des opérations INAP, les lignes grises indiquent des opérations DSS 1.

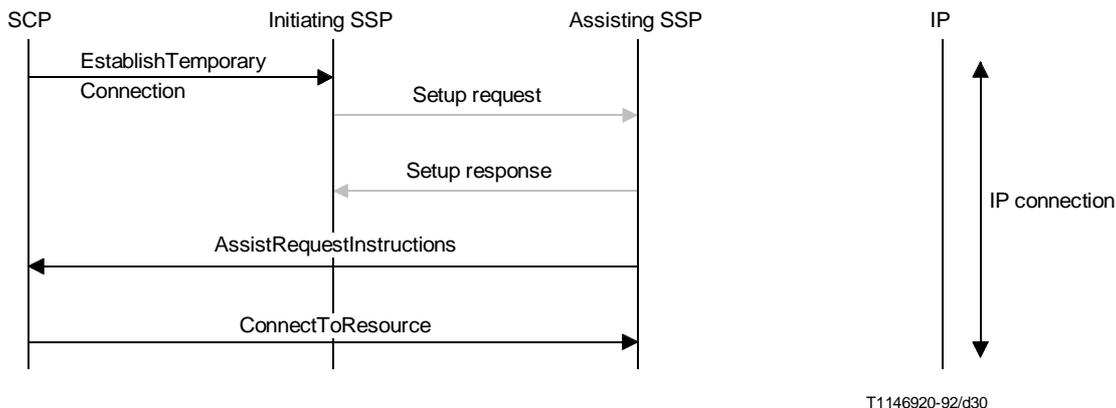


FIGURE 29/Q.1218
Préliminaires pour le cas assisté avec périphérique IP intégré ou externe et relais par le SSP des messages SCP-IP

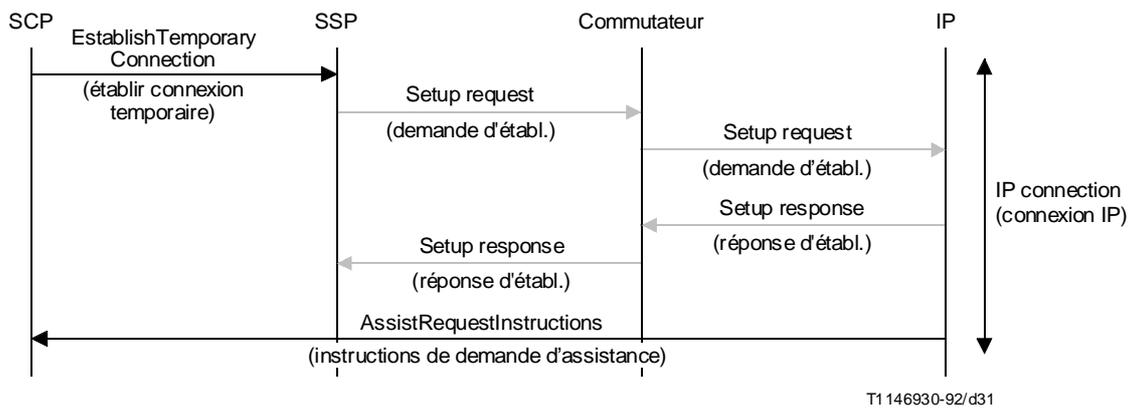


FIGURE 30/Q.1218

Préliminaires pour le cas assist avec IP externe et messagerie directe SCP-IP

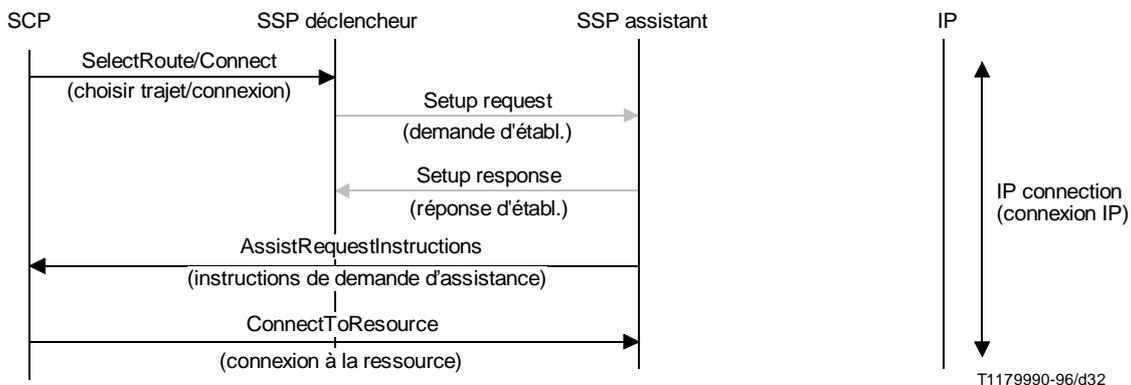


FIGURE 31/Q.1218

Préliminaires pour le cas «sans assistance»

3.1.3.5.2 Procédures d'interaction de l'utilisateur associées à l'entité SRF

Ces procédures permettent:

- l'envoi d'un ou plusieurs messages à l'utilisateur en utilisant les opérations PlayAnnouncement (diffusion d'annonce);
- d'établir un dialogue avec l'utilisateur final en utilisant une ou plusieurs séquences d'opérations PromptAndCollectUserInformation (incitation et collecte d'information d'utilisateur);
- une combinaison des deux éléments ci-dessus;
- l'annulation d'opérations PlayAnnouncement ou PromptAndCollectUserInformation en utilisant une opération générique cancel.

3.1.3.5.2.1 Play Announcement/Prompt & Collect User Information (PA/P&C)

Il n'existe que deux scénarios physiques pour l'interaction d'utilisateur:

- i) le point SSP relaie les opérations du point SCP au périphérique intelligent et les réponses du périphérique intelligent à l'intention du point SCP (cas relais SSF);
- ii) les opérations à partir du point SCP vers le périphérique intelligent et les réponses du périphérique intelligent sont envoyées entre le point SCP et le périphérique intelligent sans faire intervenir le point SSP (cas direct SCF-SRF).

Le cas i) est illustré à la Figure 32.

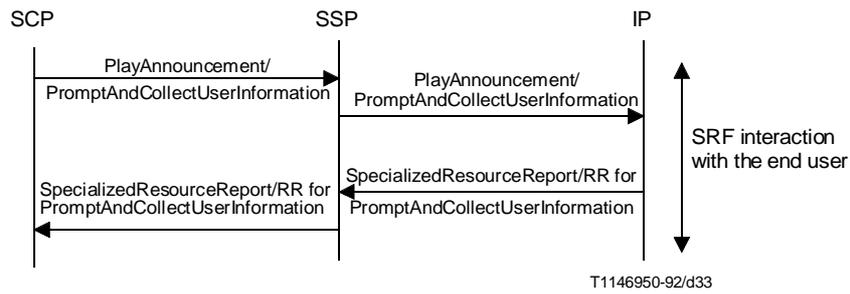


FIGURE 32/Q.1218

**Relais par le point SSP des opérations
et des réponses d'interaction d'utilisateur**

Le cas ii) est illustré à la Figure 33.

Il est également nécessaire de considérer la capacité qu'offre le sous-système TCAP du système de signalisation n° 7 pour concaténer plusieurs unités PDU en un message. Cette capacité permet, pour le premier scénario de la Figure 27, d'acheminer dans un message l'opération ConnectToResource et la première opération PlayAnnouncement/PromptAndCollectUserInformation. Ceci présente plusieurs avantages dans le scénario physique et notamment le nombre réduit de messages et la possibilité d'une meilleure qualité de fonctionnement perçue par l'utilisateur.

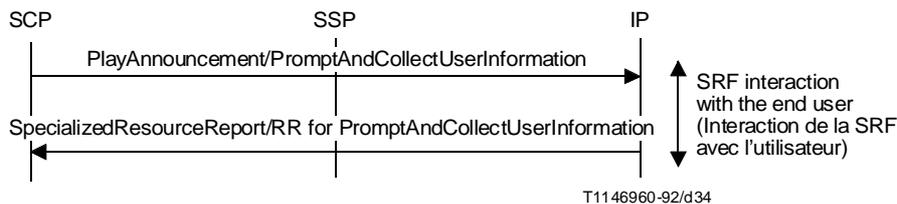


FIGURE 33/Q.1218

**Cas direct SCF-SRF des opérations
et réponses d'interaction d'utilisateur**

3.1.3.5.3 Procédures de déconnexion SRF

Les procédures de déconnexion SRF sont gérées par l'entité SCF et la procédure utilisée est choisie sur la base des besoins du service exécuté. La procédure de déconnexion du support sélectionnée par l'entité SCF doit soit permettre à l'entité SRF d'effectuer une déconnexion à la fin de l'interaction d'utilisateur, ou à l'entité SCF d'ordonner explicitement à l'entité SSF d'effectuer la déconnexion.

La déconnexion SRF ne provoque pas la déconnexion par l'entité SSF/CCF en retour vers le terminal d'utilisateur d'extrémité à moins que la transaction avec l'entité SCF ait pris fin, indiquant que l'interaction d'utilisateur a terminé l'appel. L'entité SSF/CCF reconnaît qu'une connexion vers une entité SRF a lieu car les opérations à partir de l'entité SRF à cette fin sont distinctes des opérations qui seraient utilisées pour acheminer un appel vers une destination. Cela n'a pas de conséquence sur les machines à état de la signalisation du support, car les événements de signalisation support entrants et sortants ne sont pas simplement transférés les uns vers les autres, mais sont plutôt absorbés dans le traitement de l'appel, et régénérés selon les besoins du même traitement d'appel. Par conséquent, pour obtenir la fonctionnalité souhaitée, le traitement d'appel doit simplement choisir de ne pas régénérer la déconnexion dans le sens arrière. Ce concept est illustré à la Figure 34.

De même que pour les procédures de connexion SRF, la déconnexion SRF dépend de la configuration physique du réseau.

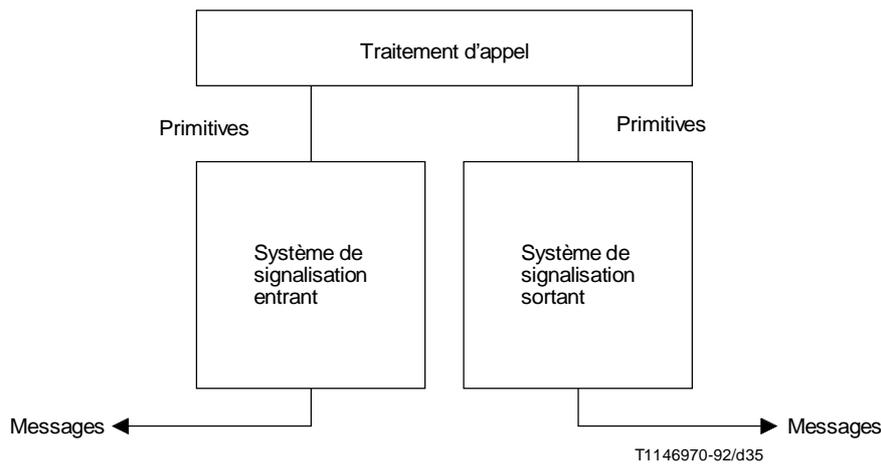


FIGURE 34/Q.1218

**Relation entre le traitement d'appel
et le système de signalisation entrant et sortant**

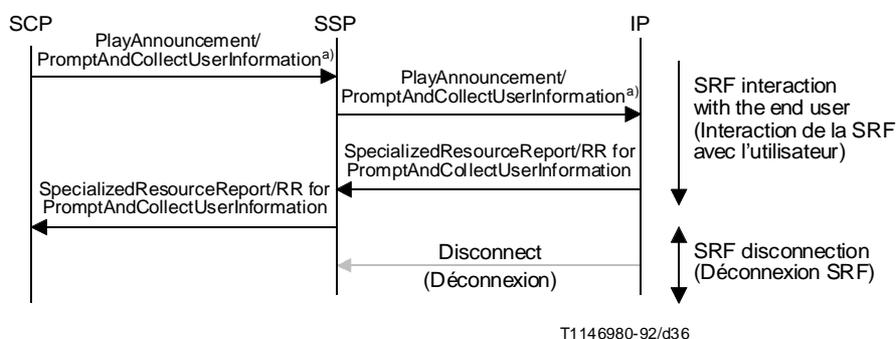
Afin de simplifier l'interface entre les entités SCF et SRF, il est nécessaire de formuler un certain nombre d'hypothèses. Ces hypothèses et les règles qui en résultent, se traduisent par des procédures ambiguës du point de vue des entités SCF et SRF. Les règles, énoncées ci-après concernent la déconnexion déclenchée par l'entité SRF ou «SRF Initiated Disconnect», et la déconnexion déclenchée par l'entité SCF ou «SCF Initiated Disconnect». D'autres scénarios sont possibles, mais ils ne sont pas présentés ici car ils sont redondants avec la fonctionnalité présentée ci-dessous ou n'apportent pas d'élément supplémentaire du point de vue du service.

- 1) Si une série d'opérations PlayAnnouncement/PromptAndCollectUserInformation doit être exécutée par la même entité SRF, la déconnexion par cette entité est bloquée pour toutes les opérations sauf la dernière et peut être bloquée sur la dernière opération PlayAnnouncement/PromptAndCollectUserInformation. Lorsqu'une opération PlayAnnouncement/PromptAndCollectUserInformation suivante est reçue, elle est mise en tampon jusqu'à la fin de toute opération PlayAnnouncement/PromptAndCollectUserInformation antérieure.
- 2) Une opération générique cancel termine l'opération PlayAnnouncement/PromptAndCollectUserInformation indiquée si elle est en cours d'exécution par l'entité SRF, mais ne déconnecte pas l'entité SRF. Si l'opération cancel s'applique à une opération PlayAnnouncement/PromptAndCollectUserInformation mise en tampon, cette opération PlayAnnouncement/PromptAndCollectUserInformation est mise au rebut, mais les opérations en cours PlayAnnouncement/PromptAndCollectUserInformation et toutes opérations PlayAnnouncement/PromptAndCollectUserInformation sont exécutées. Une entité SRF interagit avec un utilisateur seulement et, par conséquent, l'annulation d'une opération PlayAnnouncement/PromptAndCollectUserInformation n'affecte que l'utilisateur auquel cette entité SRF est connectée.
- 3) L'entité SCF doit explicitement ordonner l'opération «Disconnect» (déconnexion) ou permettre une déconnexion déclenchée par SRF à la fin de l'opération PlayAnnouncement/PromptAndCollectUserInformation. Une entité SRF maintenue en connexion sans opération PlayAnnouncement/PromptAndCollectUserInformation à exécuter peut de manière autonome se déconnecter si elle n'a pas reçu d'opération PlayAnnouncement/PromptAndCollectUserInformation au bout d'un certain temps. Tel est le cas par exemple après qu'une opération EstablishTemporaryConnection qui n'est pas suivie au bout d'un temps défini d'une opération PlayAnnouncement/PromptAndCollectUserInformation. Cette valeur de temporisation d'intégrité dépendra de la nature de l'interaction que l'entité SRF prend en charge et doit être choisie en conséquence par l'opérateur de réseau.
- 4) Lorsqu'une déconnexion déclenchée par l'entité SRF est validée dans une opération PlayAnnouncement/PromptAndCollectUserInformation, l'entité SRF doit effectuer la déconnexion à la fin de l'interaction d'utilisateur.
- 5) Lorsqu'une déconnexion déclenchée par l'entité SRF n'est pas validée, l'entité SCF doit demander à l'entité SRF de l'informer de la fin de l'interaction d'utilisateur en utilisant l'opération SpecializedResourceReport pour l'annonce effectuée ou utiliser le return result pour l'opération PromptAndCollectUserInformation.

- 6) Si l'utilisateur se déconnecte, l'entité SRF est déconnectée et l'entité SSF libère les ressources et gère la transaction entre les entités SSF et SCF tel que spécifié dans la Recommandation Q.1214 et la présente Recommandation. L'entité SRF met au rebut toute opération mise en tampon et repositionne ses ressources à l'état repos. La relation avec l'entité SCF a pris fin.
- 7) Lorsque l'entité SCF ordonne explicitement à l'entité SSF d'effectuer une déconnexion par l'opération «DisconnectForwardConnection», l'entité SSF libère la connexion support vers l'entité SRF et revient à l'état «waiting for instructions». Aucune opération signalant la déconnexion SRF de la part de l'entité SSF à l'intention de l'entité SCF n'est nécessaire.

3.1.3.5.3.1 Déconnexion déclenchée par l'entité SRF

La procédure de déconnexion par l'entité SRF est illustrée à la Figure 35. La déconnexion SRF est validée par l'entité SCF dans une opération PlayAnnouncement/PromptAndCollectUserInformation. Lorsque l'entité SRF reçoit une déconnexion par l'opération PlayAnnouncement/PromptAndCollectUserInformation autorisante, elle termine le dialogue comme indiqué par l'opération PlayAnnouncement/PromptAndCollectUserInformation, et lance la déconnexion déclenchée par l'entité SRF en utilisant la signalisation de commande du support applicable. L'entité SSF/CCF sait qu'il y a une entité SRF au cours de déconnexion mais ne poursuit pas la libéralisation de l'appel en direction de l'utilisateur. L'entité SSF revient à l'état «waiting for instructions» et exécute d'éventuelles opérations se trouvant dans le tampon. Dans le cas sans assistance, le point représenté dans la Figure 35 est le point SSP auquel l'opération est passée.



a) Déconnexion à partir de la SRF non interdite.

FIGURE 35/Q.1218

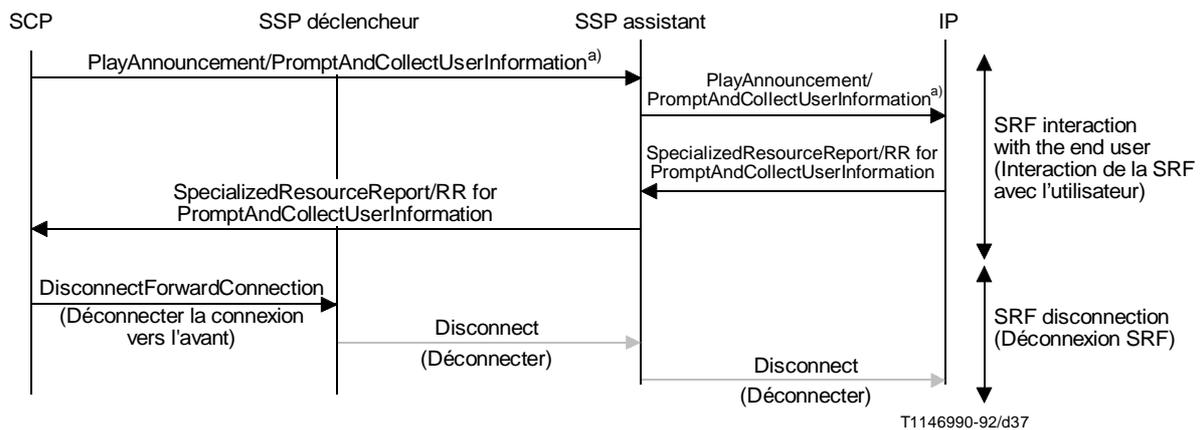
Déconnexion SCF pour les scénarios «local», «intégré» et «sans assistance»

Dans le cas d'une entité SSF assistante, les procédures de déconnexion déclenchées par l'entité SRF ne sont pas utilisées car l'entité SSF assistante reste dans l'état «waiting for instructions» et ne propage pas la déconnexion de la connexion support vers l'entité SSF déclencheuse. Les procédures de déconnexion déclenchées par l'entité SCF décrites dans les paragraphes ci-après sont utilisées pour le cas d'une entité SSF assistante.

Pour le cas direct SCF-SRF, les procédures fonctionnent de la même façon. La déconnexion SRF est validée par l'entité SCF dans une opération PlayAnnouncement/PromptAndCollectUserInformation. Lorsque l'entité SRF reçoit la déconnexion par l'opération PlayAnnouncement/PromptAndCollectUserInformation autorisante, elle met fin au dialogue comme demandé dans l'opération PlayAnnouncement/PromptAndCollectUserInformation, et déclenche ensuite une déconnexion déclenchée par l'entité SRF au moyen de la signalisation de commande du support applicable. L'entité SSF/CCF déclencheuse sait qu'il s'agit d'une déconnexion SRF et ne poursuit pas la libération de l'appel en direction de l'utilisateur. L'entité SSF déclencheuse revient à l'état «waiting for instructions» et exécute les éventuelles opérations mises en tampon.

3.1.3.5.3.2 Déconnexion déclenchée par l'entité SCF

La procédure de déconnexion déclenchée par l'entité SCF est illustrée à la Figure 36. Les messages supports sont représentés en gris. La figure ne montre que le cas d'une entité SSF assistante et pas le cas SCF-SRF direct. Pour déclencher la déconnexion de l'entité SRF lancée par l'entité SCF, l'entité SCF doit demander et recevoir une réponse à la dernière opération de PlayAnnouncement/PromptAndCollectUserInformation demandée. L'opération SpecializedResourceReport contient une information «annonce terminée» et le retour du résultat (return result) pour l'opération P&C contient l'élément «information collectée».



^{a)} Déconnexion à partir de la SRF interdite.

FIGURE 36/Q.1218

Déconnexion déclenchée par la SCF pour le scénario avec assistance

La déconnexion déclenchée par l'entité SCF fait appel à une opération appelée `DisconnectForwardConnection`. Une fois que cette opération a été reçue par l'entité SSF, elle déclenchera une action «libération de la connexion du canal support» entre les entités physiques contenant les entités SSF et SRF, en utilisant la signalisation de commande du support applicable. Etant donné que les entités SCF (qui déclenche la déconnexion), SSF (qui demande à la signalisation du support d'effectuer une déconnexion) et SRF (qui reçoit la notification de déconnexion par l'intermédiaire de la signalisation du support) savent qu'une déconnexion est en cours, ces entités sont synchronisées. On peut donc utiliser une fin préétablie pour clore la transaction. Cela n'empêche pas d'utiliser pour cela des messages finaux explicites.

Dans le cas d'une entité SSF assistante, le point SSP, sur réception de l'opération `DisconnectForwardConnection` provenant du point SCP, effectue la déconnexion vers l'avant en direction du point SSP assistant, et cette déconnexion se propage vers le périphérique intelligent. Le point SSP déclenchant, sachant que la connexion vers l'avant a été déclenchée suite à une opération `EstablishTemporaryConnection` ne se déconnecte pas vers l'utilisateur mais revient à l'état «waiting for instructions».

3.1.3.5.4 Exemples de séquences complètes d'interaction d'utilisateur

Les figures ci-après et les tableaux qui les accompagnent contiennent des exemples de séquences complètes d'opérations d'interaction d'utilisateur couvrant les trois étapes suivantes:

- connecter l'entité SRF et l'utilisateur d'extrémité (connexion support) et établir la relation SCF-SRF;
- interagir avec l'utilisateur;
- déconnecter l'entité SRF et l'utilisateur d'extrémité (connexion support) et terminer la relation SCF-SRF.

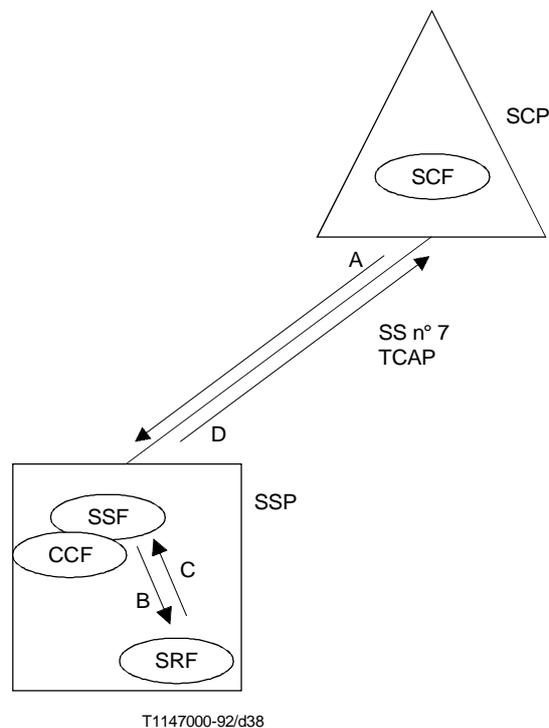


FIGURE 37/Q.1218
Point SSP avec SRF intégrée

La Figure 37 représente le cas d'un point SSP avec entité SRF intégrée, dans ce cas, les scénarios de procédure peuvent être mis en correspondance comme suit:

Nom de procédure	Opérations	Flux de protocole
Connecter à la ressource les premières opérations PA/P&C	ConnectToResource; PlayAnnouncement/ PromptAndCollectUserInformation Setup; PlayAnnouncement/ PromptAndCollectUserInformation	A B
Interaction d'utilisateur	PlayAnnouncement/ PromptAndCollectUserInformation SpecializedResourceReport/RR for PromptAndCollectUserInformation	A puis B C puis D
Déconnexion déclenchée par l'entité SRF	SpecializedResourceReport/RR for PromptAndCollectUserInformation Disconnect	C puis D C (commande du support intégrée au point SSP)
Déconnexion déclenchée par l'entité SCF	SpecializedResourceReport/RR for PromptAndCollectUserInformation DisconnectForwardConnection Disconnect	C puis D A B (commande du support intégrée au point SSP)

On peut simplement étendre ce cas à celui où l'entité SRF se trouve dans un périphérique intelligent localement associé au point SSP. Les opérations SCF-IP sont relayées via l'entité SSF dans le point SSP. Cette situation est décrite à la Figure 38.

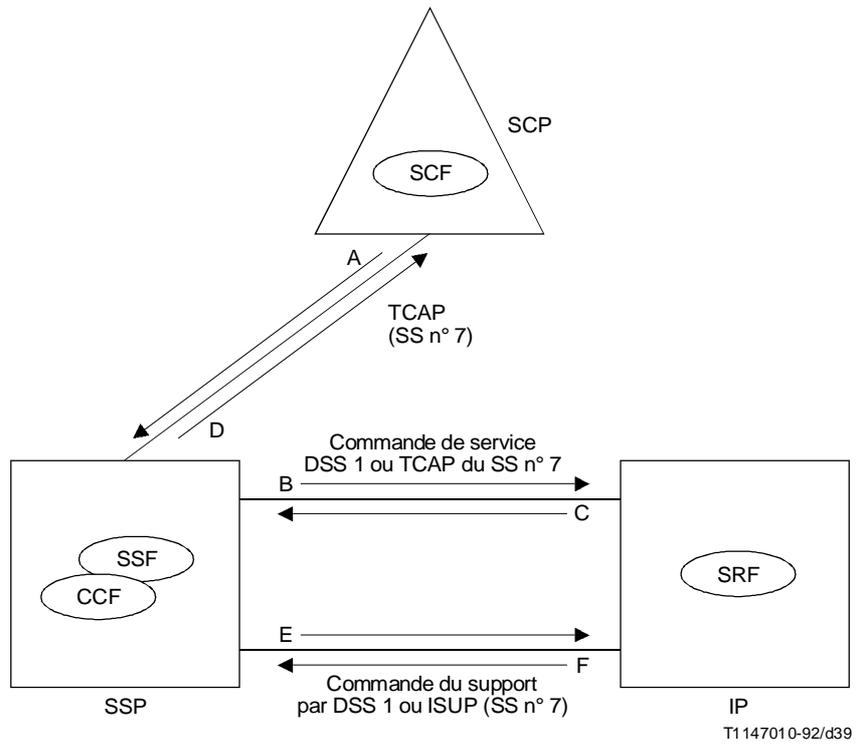


FIGURE 38/Q.1218

Le point SSP assure le relais des messages entre le point SCP et le périphérique intelligent (IP)

Les scénarios de procédure pour cette entité SSF relais avec un périphérique intelligent (voir la Figure 38) peuvent être mis en correspondance comme suit:

Nom de procédure	Opérations	Flux de protocole
Connexion à la ressource et premières opérations PA/P&C	ConnectToResource; PlayAnnouncement/ PromptAndCollectUserInformation <i>Si on utilise la signalisation DSS 1</i> Setup; PlayAnnouncement/ PromptAndCollectUserInformation <i>Si on utilise le système de signalisation n° 7</i> IAM PlayAnnouncement/ PromptAndCollectUserInformation	A E et B (élément d'information facilité) E B
Interaction d'utilisateur	PlayAnnouncement/ PromptAndCollectUserInformation SpecializedResourceReport/RR for PromptAndCollectUserInformation	A puis B C puis D
Déconnexion déclenchée par le SRF	SpecializedResourceReport/RR for PromptAndCollectUserInformation <i>Si DSS 1 est utilisé</i> Disconnect <i>Si on utilise le système de signalisation n° 7</i> Libération	C puis D F F
Déconnexion déclenchée par le SCF	SpecializedResourceReport/RR for PromptAndCollectUserInformation DisconnectForwardConnection <i>Si DSS 1 est utilisé</i> Disconnect <i>Si on utilise le système de signalisation n° 7</i> Libération	C puis D A E E

Dans certains cas, le périphérique intelligent peut disposer d'une interface SS n° 7 ou autre avec le point SCP de commande. Ce cas est représenté à la Figure 39. Il convient de noter que le point SCP doit assurer la corrélation de deux transactions afin de coordonner les activités.

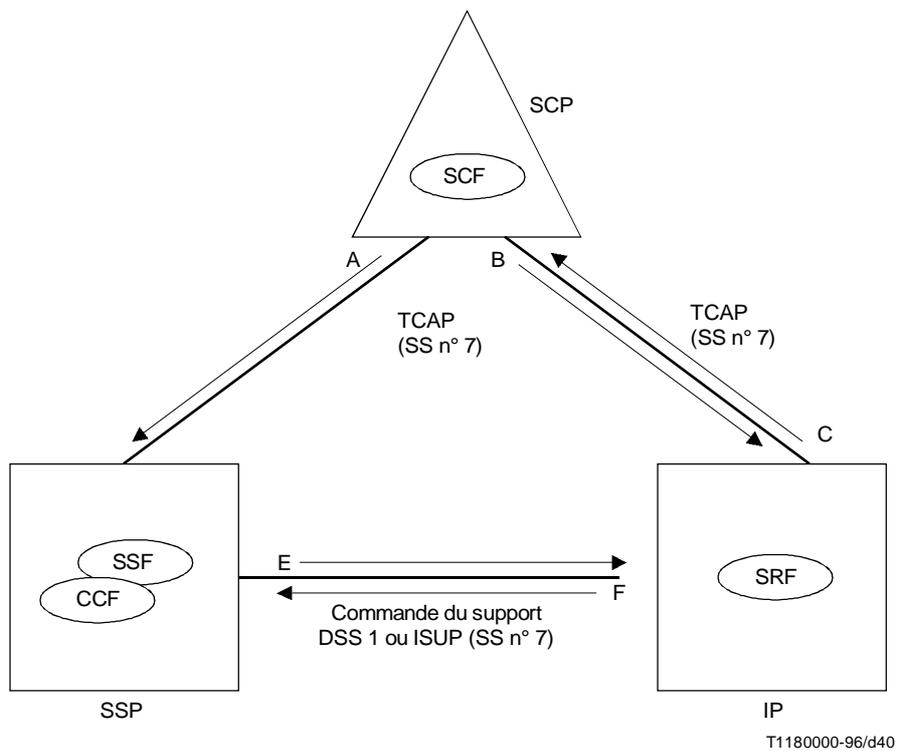


FIGURE 39/Q.1218
Transfert d'information direct SCP-IP

A la Figure 39, les scénarios de procédure peuvent être mis en correspondance comme suit:

Nom de procédure	Opérations	Flux de protocole
Connecter à la ressource	EstablishTemporaryConnection <i>Si DSS 1 est utilisé</i> Setup AssistRequestInstructions <i>Si SS n° 7 est utilisé</i> IAM AssistRequestInstructions	A E C E C
Interaction d'utilisateur	PlayAnnouncement/PromptAndCollectUser Information SpecializedResourceReport/RR for PromptAndCollectUserInformation	B C
Déconnexion déclenchée par le SRF	SpecializedResourceReport/RR for PromptAndCollectUserInformation <i>Si DSS 1 est utilisé</i> Disconnect <i>Si SS n° 7 est utilisé</i> Libération	C F F
Déconnexion déclenchée par le SCF	SpecializedResourceReport/RR for PromptAndCollectUserInformation DisconnectForwardConnection <i>Si DSS 1 est utilisé</i> Disconnect <i>Si SS n° 7 est utilisé</i> Libération	C A E E

Le scénario avec entité SSF assistante fait intervenir des extensions de procédure directe par rapport aux cas représentés ci-dessus. Une mise en correspondance du cas avec entité SSF assistante est illustrée à la Figure 40. Dans ce cas, on ne peut utiliser la déconnexion déclenchée par l'entité SRF. D'autres mises en correspondance physiques peuvent être obtenues et sont décrites dans le texte qui suit la figure et le tableau associé.

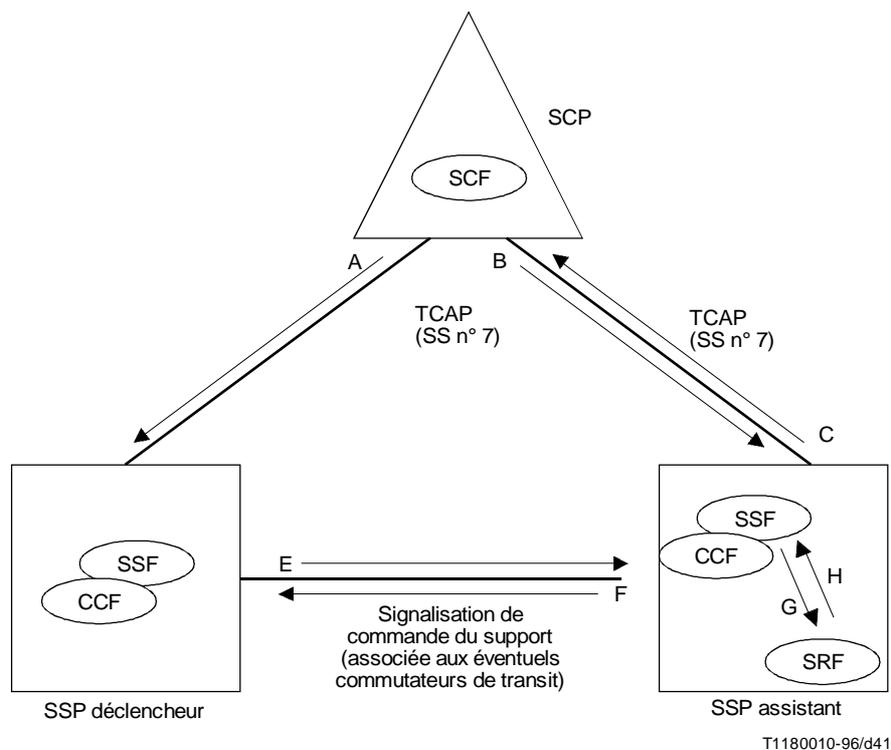


FIGURE 40/Q.1218
Point SSP assistant (SSP relais)

Il convient de noter que le cas SRF intégré et SSF relais requiert une transaction entre le point SCP et le point SSP assistant (voir la Figure 40) mais cela n'est pas valable pour le cas SCP direct puisque la transaction a lieu directement entre le point SCP et le périphérique intelligent relié à l'autocommutateur distant. Dans ce dernier cas, tous les commutateurs de transit, y compris celui auquel le périphérique intelligent se trouve connecté (SRF), sont transparents aux procédures.

Il convient également de noter que le point SCP doit également assurer la corrélation des deux transactions.

A la Figure 40, les scénarios de procédure peuvent être mis en correspondance comme suit:

Nom de procédure	Opérations	Flux de protocole
Préambule Assist	EstablishTemporaryConnection <i>Si DSS 1 est utilisé</i> Setup AssistRequestInstructions ConnectToResource Setup ResetTimer <i>Si SS n° 7 est utilisé</i> IAM AssistRequestInstructions ConnectToResource Setup ResetTimer	A E C B G A E C B G A
Interaction d'utilisateur	PlayAnnouncement/ PromptAndCollectUserInformation SpecializedResourceReport/RR for PromptAndCollectUserInformation	B puis G H puis C
Déconnexion déclenchée par SCF	SpecializedResourceReport/RR for PromptAndCollectUserInformation DisconnectForwardConnection <i>Si DSS 1 est utilisé</i> Disconnect <i>Si SS n° 7</i> Libération	H puis C A E puis G (commande du support intra-SSP) E et G (commande du support intra-SSP)

Il convient de noter que le cas du point SSP assistant représenté à la Figure 40 peut être généralisé pour couvrir à la fois les cas où l'entité SRF est intégrée dans le point SSP assistant (comme indiqué) et le cas où l'entité SRF est localement connectée au point SSP assistant. Dans ce dernier cas, la communication SRF (flux de protocole B, C, G et H) serait conforme au scénario physique représenté à la Figure 38.

Le scénario du service sans assistance peut également être considéré comme composé d'une séquence faisant intervenir un service RI acheminant un appel d'un point SSP à un autre, suivi par l'un quelconque des scénarios d'interaction avec l'utilisateur physique. La Figure 40 peut également être utilisée pour décrire ce scénario.

3.1.3.5.4.1 Séquences de message pour assistance au service

Le sous-alinéa suivant fournit des détails supplémentaires sur les séquences de message dans la procédure d'assistance au service de la Figure 40:

- 1) L'entité SCP, durant le traitement d'une demande d'instruction, détermine que des ressources situées à distance de l'entité SSP déclencheuse sont nécessaires et que le traitement d'appel continuera à partir de l'entité SSP de déclenchement, après que les ressources à distance aient été utilisées (par exemple l'appel sera envoyé vers une adresse de destination après que l'information ait été collectée par le participant). Une opération EstablishTemporaryConnection contenant l'adresse de l'entité SSP assistante (pour l'acheminement de l'appel), l'identificateur ScfID et l'identificateur de corrélation (tous deux utilisés pour que l'entité SSP assistante puisse rétablir la communication vers l'entité SCP) est envoyée à l'unité déclencheuse SSP. L'opération EstablishTemporaryConnection est utilisée à la place d'une opération normale Connect en raison de la nature de la connexion vers l'entité SSP assistante. L'entité SSP déclencheuse doit être informée que l'entité SCP lui demandera de continuer le traitement de l'appel jusqu'à un certain point dans le futur.

NOTE 1 – l'identificateur de corrélation et l'identificateur ScfID peuvent être inclus dans l'adresse d'acheminement de l'entité SSP assistante.

Flux de protocole A

- 2) L'entité SSP déclencheuse achemine l'appel vers l'entité SSP assistante. L'identificateur ScfID et l'identificateur de corrélation sont envoyés vers l'entité SSP assistante. La signalisation existant dans la bande passante et les éléments d'information SS7 (par exemple le numéro d'acheminement) pourraient être utilisés pour transporter ces informations. Le mécanisme de transport qui est utilisé pour envoyer cette information entre les entités SSP est indépendant des procédures de logique de commande de service entre l'entité SCF et l'entité SSF.

Flux de protocole E

- 3) L'entité SSP assistante utilise l'opération AssistRequestInstructions pour établir la communication avec l'entité SCP. L'identificateur de corrélation est envoyé dans le message AssistRequestInstructions pour permettre à l'entité SCP d'effectuer une corrélation entre deux transactions.

Flux de protocole C

- 4) L'entité SCP envoie des instructions vers l'entité SSP assistante sur la base de la commande de logique de service.

Flux de protocole B

- 5) L'entité SCP peut avoir besoin de créer des événements de réinitialisation de temporisation vers l'entité SSP déclencheuse de sorte qu'elle ne temporise pas l'appel.

Flux de protocole A

NOTE 2 – L'opération de réinitialisation de la temporisation est facultative.

- 6) Quand les fonctions de ressources ont été exécutées, une opération DisconnectForwardConnection (Connexion déconnectée en avant) est envoyée à l'entité SSP déclencheuse. Cela indique que la connexion temporaire avec l'entité SSP assistante doit être interrompue.

Flux de protocole A

- 7) L'entité SSP déclencheuse envoie un message par l'intermédiaire d'une signalisation de commande du support vers l'entité SSP assistante afin de mettre fin à la transaction "assist".

Flux de protocole E

- 8) La commande d'appel est renvoyée vers l'entité SSP déclencheuse.

3.1.3.5.4.2 Séquences de message pour la procédure de transfert de tâches

Les paragraphes suivants indiquent les séquences de message dans la procédure de transfert de tâches qui utilise les flux de protocoles indiqués à la Figure 40.

- 1) L'entité SCP, pendant le traitement d'une demande d'instruction, définit que les ressources situées à distance de l'entité SSP déclencheuse sont nécessaires et que le traitement d'appel continuera à partir de l'entité SSP déclencheuse après que les ressources à distance aient été utilisées (par exemple une annonce terminale sera émise). Une opération Connect contenant l'adresse de l'entité SSP assistante (pour l'acheminement de l'appel), l'identificateur ScfID et l'identificateur de corrélation (tous deux utilisés pour aider l'entité SSP assistante à rétablir la communication vers l'entité SCP) est envoyée à l'entité SSP déclencheuse.

NOTE – L'identificateur ScfID et l'identificateur de corrélation correlationID peuvent être inclus dans l'adresse d'acheminement vers l'entité SSP assistante.

Flux de protocole A

- 2) L'entité SSP déclencheuse achemine l'appel vers l'entité SSP assistante. L'identificateur ScfID et l'identificateur de corrélation sont envoyés vers l'entité SSP assistante. La signalisation existant dans la bande passante et les éléments d'information du réseau sémaphore (par exemple numéro d'acheminement) pourraient être utilisés pour transporter cette information. Le mécanisme de transport utilisé pour envoyer cette information entre les entités SSP est indépendant des procédures de commande de logique de service entre l'entité SCF et l'entité SSF.

Flux de protocole E

- 3) L'entité SSP assistante utilise une opération AssistRequestInstructions pour établir la communication avec l'entité SCP. L'identificateur de corrélation CorrelationID est envoyé dans l'opération AssistRequestInstructions pour permettre à l'entité SCP d'effectuer une corrélation entre deux transactions. L'opération AssistRequestInstructions est utilisée à la place d'une instruction normale de demande (point DP initial ou opération spécifique au point DP), étant donné que l'entité SCP doit associer l'opération AssistRequestInstructions provenant du périphérique IP-SSP assistant à un dialogue déjà actif que le point SCP entretient avec une autre entité SSP.

Flux de protocole C

- 4) L'entité SCP envoie des instructions au point SSP assistant sur la base de la commande de logique de service.

Flux de protocole B

- 5) La commande d'appel reste au point SSP assistant.

Les mêmes procédures d'assistance et de transfert de tâches peuvent être réutilisées pour un lien direct avec un périphérique intelligent dans les ensembles de capacités présents et futurs.

3.1.4 Procédures relatives à l'entité application de l'entité SDF

3.1.4.1 Considérations générales

Ce paragraphe contient la définition des procédures de l'entité d'application SDF associées à l'interface SDF-SCF. Ces procédures sont fondées sur l'utilisation du système de signalisation n° 7 (SS n° 7).

D'autres capacités peuvent être prises en charge selon la réalisation dans les entités physiques SCP, SDP, SSP, AD ou SN.

L'entité application, selon l'architecture définie dans les Recommandations Q.700, Q.9771 et Q.1400, comprend le sous-système TCAP (*Transaction capabilities application part*) et un ou plusieurs éléments du service d'application appelés les utilisateurs-TC, qui se réfèrent à l'Annuaire (Recommandations de la série X.500). Les paragraphes suivants définissent les éléments du service application (ASE) des utilisateurs TC et les règles SACF & MACF, qui assurent l'interface avec le sous-système TCAP au moyen de primitives spécifiées dans la Recommandation Q.771.

La procédure peut également être utilisée avec d'autres systèmes d'acheminement de message de signalisation prenant en charge les structures de la couche application définies.

Au cas où les interprétations concernant les procédures d'entités d'application définies ci-dessous divergeraient des procédures détaillées et des règles d'utilisation du service TCAP, les spécifications et les règles contenues en 3.3 et 2.2.2.2 devraient être appliquées.

3.1.4.2 Modèle et interfaces

Le modèle fonctionnel de l'entité application SDF est représenté à la Figure 41; les éléments du service application assurent d'une part, l'interface avec le sous-système TCAP permettant de communiquer avec l'entité SCF, et d'autre part, l'interface avec les fonctions de maintenance. Le domaine d'application de la présente Recommandation est limité à la zone ombrée de la Figure 41.

Les interfaces représentées à la Figure 41 utilisent les primitives d'élément ASE d'utilisateur du gestionnaire TC spécifiées dans la Recommandation Q.771 [interface (1)] et les primitives de couche réseau (N-) spécifiées dans la Recommandation Q.711 [interface (2)]. Les opérations et les paramètres associés au protocole d'application du réseau intelligent (protocole INAP) sont définis à l'article 2.

Une instance de la machine SDF-FSM peut être créée lorsque le traitement d'un appel RI est reçu en provenance de l'entité SCF.

La machine SDF-FSM gère l'interaction avec la machine SCF-FSM.

3.1.4.3 Relation entre la machine SDF-FSM et la fonction de maintenance

L'interface élémentaire entre la machine SDF-FSM et les fonctions de maintenance est une interface interne et ne fait pas l'objet d'une normalisation pour l'ensemble de capacités CS-1.

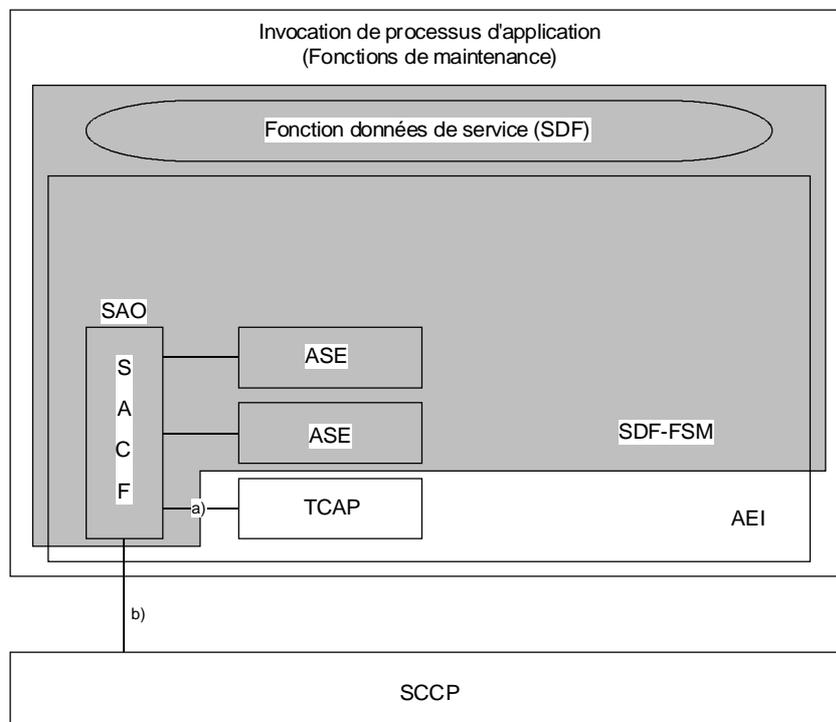
3.1.4.4 Modèle de transition d'état SDF

S'agissant de l'ensemble de capacités CS-1 du réseau intelligent, la tâche de l'entité SDF est de répondre (de façon synchrone) à chaque demande émanant de l'entité SCF après la procédure de Rattachement. La machine à états finis (FSM) (*finite state machine*) applicable est décrite à la Figure 42.

Chaque état est étudié dans l'un des paragraphes suivants.

Les règles générales applicables à plusieurs états sont les suivantes.

Dans un état quelconque, si le dialogue avec l'entité SCF est fini, la machine SDF-FSM revient à l'état "**Idle**" après s'être assurée que toutes les ressources attribuées à l'appel ont été libérées.



T1147040-92/d42

AEI Invocation de l'entité application (*application entity invocation*)
 SDF Fonction données de service (*service data function*)
 FSM Machine à états finis (*finite state machine*)
 SACF Fonction de commande d'association unique (*single association control function*)
 SAO Objet d'association unique (*single association object*)

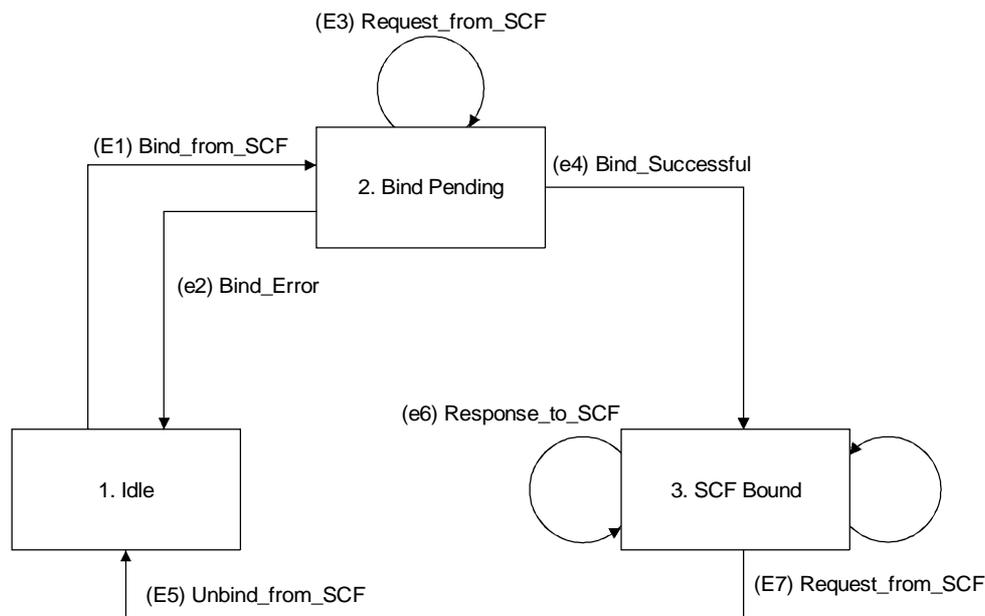
a) Primitives TC ou primitives Q.932.

b) Primitives N.

NOTE – La machine SDF-FSM comporte plusieurs machines FSM.

FIGURE 41/Q.1218

Modèle fonctionnel de l'entité application de l'entité SDF



T1169210-94/d43

FIGURE 42/Q.1218
Machine SDF-FSM

3.1.4.4.1 Etat 1 – "Idle"

Le seul événement accepté dans cet état est:

- (E1) Bind_from_SCF (Détachement de l'entité SCF) – Événement externe causé par la réception de l'opération Bind from the SCF (Détachement de l'unité SCF). Cet événement provoque la sortie de cet état et le passage à l'état 2, **Bind Pending** (Rattachement en cours).

3.1.4.4.2 Etat 2 – "Bind Pending" (Rattachement en cours)

Dans cet état, une demande de rattachement a été reçue de l'entité SCF. L'entité SDF effectue les procédures de commande d'accès après l'opération de rattachement (par exemple authentification d'accès). Il se peut également que l'opération de rattachement soit une opération factice. L'authentification d'accès n'est alors pas nécessaire. Trois événements sont pris en considération dans cet état:

- (e2) Bind_Error (Erreur de Rattachement) – Événement interne causé par l'échec de l'opération de rattachement précédemment présentée à l'entité SDF. Cet événement provoque la sortie de cet état et le passage à l'état 1 "**Idle**" (repos) et une erreur de rattachement est renvoyée à l'entité SCF qui a fait l'appel;
- (E3) Request_from_SCF (Demande émanant de l'entité SCF) – Événement externe provoqué par la réception d'opérations avant que le résultat de l'opération Bind soit connu. Le modèle SDSM reste dans le même état;
- e4) Bind_Successful – Événement interne provoqué par l'accomplissement de l'opération Bind "Rattachement" précédemment présentée à l'entité SDF. Cet événement provoque la sortie de cet état et le passage à l'état 3, "**Rattachement à l'entité SCF**".

3.1.4.4.3 Etat 3 – "SCF Bound" (Rattachement à l'entité SCF)

Dans cet état, l'accès de l'entité SCF à l'entité SDF a été autorisé et les opérations en provenance de l'entité SCF sont acceptées. Outre le fait qu'elle attende des demandes en provenance de l'entité SCF, l'entité SDF peut, dans un tel cas, envoyer les réponses à des opérations préalablement présentées. On considère que trois événements appartiennent à cet état:

- (E5) Unbind_from_SCF (Détacher de l'entité SCF) – Événement externe provoqué par la réception de l'opération Unbind from the SCF (Détachement de l'entité SCF). L'association SCF-SDF est terminée et toutes les ressources associées sont libérées. Cet événement provoque la sortie de cet état et le passage à l'état 1, "Idle";

- (e6) Response_to_SCF (Réponse à l'entité SCF) – Événement interne provoqué par l'exécution des opérations précédemment présentées par l'entité SCF. Les réponses sont envoyées à l'entité SCF. Le modèle SDSM reste dans le même état;
- (E7) Request_from_SCF (Demande à partir de l'entité SCF) – Événement externe provoqué par la réception d'une demande présentée par l'entité SCF à l'entité SDF. Le modèle SDSM reste dans le même état.

En plus de cette machine à états finis, une description en langage SDL est indiquée à l'Annexe B.

3.2 Procédures de traitement des erreurs

Ce paragraphe définit les procédures génériques de traitement des erreurs applicables au protocole INAP dans l'ensemble de capacités CS-1 du réseau intelligent. Les descriptions de procédures d'erreurs ont été divisées en deux paragraphes, le paragraphe 3.2.1 énumérant les erreurs relatives aux opérations INAP et le paragraphe 3.2.2 énumérant les erreurs relatives aux états d'erreurs dans les différentes entités fonctionnelles ne concernant pas directement les opérations INAP.

3.2.1 Procédures d'erreurs liées aux opérations

Les paragraphes suivants définissent le traitement générique d'erreur pour les erreurs liées aux opérations. Les erreurs sont définies comme des erreurs d'opération dans 2.1.2. Les services du sous-système TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits dans 3.4.

Les erreurs correspondant à une procédure spécifique d'une opération sont décrites en 3.3 avec une procédure détaillée relative à cette opération.

Il est possible que toutes les erreurs pouvant être détectées par le décodeur ASN.1 l'aient déjà été pendant le décodage du message de sous-système TCAP et soient indiquées par l'indication d'erreurs de l'utilisateur TC "MistypedParameter" (Paramètre de type erroné) dans l'opération TC-U-Reject.

3.2.1.1 Erreur d'attribut

3.2.1.1.1 Description générale

3.2.1.1.1.1 Description d'erreur

Cette erreur est envoyée par l'entité SDF à l'entité SCF pour rendre compte d'un problème relatif à un attribut. Les conditions dans lesquelles une erreur d'attribut doit être présentée sont définies en 12.4/X.511 (1993).

3.2.1.1.1.2 Description d'argument

Le paramètre d'erreur d'attribut et les codes d'incidents sont spécifiés en 12.4/X.511 (1993).

3.2.1.1.2 Opérations SCF -> SDF

- Search;
- Add Entry;
- Modifie Entry.

Procédure de l'entité d'appel (SCF)

A) Envoi de l'opération

Précondition:	SCSM état 4	Rattaché SDF ou
	SCSM état 2	Attente des demandes subséquentes
Postcondition:	SCSM état 4	Rattaché SDF ou
	SCSM état 2	Attente des demandes subséquentes

B) Renvoi de l'erreur

Précondition:	SCSM état 4	Rattaché SDF
Postcondition:	SCSM état 4	Rattaché SDF

La procédure de traitement des erreurs dépend de la logique de traitement de service. Si l'entité SCF peut changer la demande, elle peut faire une autre interrogation SDF, sinon il devrait être mis fin au traitement du service.

Procédure de l'entité de réponse (SDF)

A) Réception de l'opération		
Précondition:	SDF-FSM état 3	Rattaché SCF
Postcondition:	SDF-FSM état 3	Rattaché SCF
B) Renvoi de l'erreur		
Précondition:	SDF-FSM état 3	Rattaché SCF
Postcondition:	SDF-FSM état 3	Rattaché SCF

L'entité SDF n'a pas pu effectuer l'opération en raison d'un problème d'attribut et, de ce fait, elle envoie une erreur d'attribut à l'entité SCF. Une fois que l'erreur a été envoyée, aucun autre traitement des erreurs n'intervient.

3.2.1.2 Canceled (Annulé)

3.2.1.2.1 Description générale

3.2.1.2.1.1 Description d'erreur

L'erreur "Annulé" indique à l'entité SCF que l'annulation, telle qu'elle a été demandée par l'entité SCF pour une opération spécifique, a bien été effectuée. L'entité SCF ne peut qu'annuler certaines opérations prédéfinies SCF → SRF.

3.2.1.2.2 Opérations SCF → SRF

- PlayAnnouncement (Passage d'annonce);
- PromptAndCollectUserInformation (Recueil d'informations d'utilisateur sur invitation).

Procédures de l'entité d'appel (SCF)

A) Envoyer Annulation		
Précondition:	SCSM état 4.1	Attente de la réponse de l'entité SRF.
Postcondition:	SCSM état 4.1	Attente de la réponse de l'entité SRF.

L'entité SCF envoie une opération Cancel (Annulation) après qu'une des opérations PlayAnnouncement ou PromptAndCollectUserInformation ait été envoyée. L'entité SCF reste dans le même état.

B) Recevoir Erreur annulée		
Précondition:	SCSM n'importe quel état	Selon la logique de traitement de service.
Postcondition:	SCSM n'importe quel état	Selon la logique de traitement de service.

Après avoir envoyé une opération Cancel (Annulation), la logique de traitement de service peut par exemple continuer à émettre d'autres opérations PlayAnnouncement ou PromptAndCollectUserInformation ou une indication DisconnectForwardConnection (Déconnecter la connexion vers avant). Le message Canceled Error peut donc être reçu dans n'importe quel état. Son traitement est dépendant de la logique de traitement de service.

Procédures de l'entité de réponse (SRF)

A) Recevoir Annulation		
Précondition:	SRSF état 3	Interaction avec l'utilisateur.
Postcondition:	SRSF état 3	Interaction avec l'utilisateur.

Il est mis fin à l'opération indiquée de type PlayAnnouncement (passage d'annonce) ou PromptAndCollectUserInformation si cette opération est en cours d'exécution ou qu'elle est effacée de la mémoire tampon. Si l'opération PlayAnnouncement ou PromptAndCollectUserInformation est déjà exécutée, cela entraîne une défaillance "CancelFailed" (Echec de l'annulation).

B) Envoyer Cancel Error (Erreur sur Annulation)		
Précondition:	SRSF état 3	Interaction avec l'utilisateur.
Postcondition:	SRSF état 3	Interaction avec l'utilisateur.

Après avoir renvoyé l'erreur "Annulé", l'entité SRF reste dans le même état. L'exécution de l'opération de type PlayAnnouncement (passage d'annonce) ou PromptAndCollectUserInformation est interrompue, c'est-à-dire que l'entité SRF reste connectée et le passage de l'annonce ou le message PromptAndCollectUserInformation suivant est exécuté, le cas échéant.

3.2.1.3 CancelFailed (Echec de l'Annulation)

3.2.1.3.1 Description générale

3.2.1.3.1.1 Description d'erreur

Cette erreur est renvoyée par Cancel (Annulation) si l'annulation d'une opération, telle qu'elle a été demandée par l'entité SCF, ne s'est pas déroulée normalement. Les raisons possibles de cet échec sont:

- 1) unknownOperation (Opération inconnue), si l'identificateur InvokeID de l'opération à annuler n'est pas connu de l'entité SRF (cela peut également se produire dans le cas où l'opération a déjà été effectuée);
- 2) tooLate (Trop tard), quand l'identificateur InvokeID est connu mais que l'exécution de l'opération se trouve dans un état qui ne peut plus être annulé. Par exemple, l'annonce est terminée mais l'opération SpecializedResourceReport n'a pas encore été envoyée à l'entité SCF. Les conditions d'apparition de la cause d'échec "tooLate" peuvent être dépendantes de la réalisation;
- 3) operationNotCancellable (opération non annulable), quand l'identificateur invokeID pointe sur une opération que l'entité SCF n'est pas autorisée à annuler.

3.2.1.3.1.2 Description d'argument

```
PARAMETER SEQUENCE {
    problem      [0] ENUMERATED {
        unknown operation (0),
        tooLate (1),
        operationNotCancellable (2) },
    operation     [1] InvokeID
}
-- L'opération a échoué en annulation.
```

3.2.1.3.2 Opérations SCF → SSF

- CancelStatusReportRequest (Annulation de demande de compte rendu d'état).
- Cancel (Annuler).

3.2.1.3.3 Opérations SCF → SRF

- Cancel (Annuler).

Procédures de l'entité d'appel (SCF)

A) Envoyer Cancel

Précondition: SCSM état 4.1 Attente de la réponse en provenance de l'entité SRF.

Postcondition: SCSM état 4.1 Attente de la réponse en provenance de l'entité SRF.

L'entité SCF envoie un message Cancel après qu'une opération PlayAnnouncement ou PromptAndCollectUserInformation a été envoyée. L'entité SCF reste dans le même état.

B) Recevoir CancelFailedError (Erreur échec d'annulation)

Précondition: SCSM n'importe quel état Selon la logique de traitement de service.

Postcondition: SCSM n'importe quel état Selon la logique de traitement de service.

Après avoir envoyé une opération Cancel, la logique de traitement de service peut continuer (par exemple à envoyer un autre message PlayAnnouncement ou PromptAndCollectUserInformation ou DisconnectForwardConnection). Le message CancelFailedError peut donc être reçu dans n'importe quel état. Le traitement dépend de la logique de traitement de service.

Procédures de l'entité de réponse (SRF)

A) Recevoir Cancel. Cependant, le message PlayAnnouncement ou PromptAndCollectUserInformation indiqué n'est pas connu, ou a déjà été exécuté. Cela provoque un incident, CancelFailed (Echec d'annulation).

Précondition: SRSM état 3 Interaction avec l'utilisateur.

Postcondition: SRSM état 3 Interaction avec l'utilisateur.

ou SRSM état 1 Repos.

- B) Envoyer CancelFailed Error
- | | | |
|----------------|-------------|---------------------------------|
| Précondition: | SRSM état 3 | Interaction avec l'utilisateur. |
| | ou | SRSM état 1 |
| | | Repos. |
| Postcondition: | SRSM état 3 | Interaction avec l'utilisateur. |
| | ou | SRSM état 1 |
| | | Repos. |

Après avoir renvoyé CancelFailed, l'entité SRF reste dans le même état.

3.2.1.4 ETCFailed (Echec d'établissement de connexion temporaire)

3.2.1.4.1 Description générale

3.2.1.4.1.1 Description d'erreur

ETCFailed (Echec d'établissement de connexion temporaire) est une erreur de l'entité SSF vers l'entité SCF, indiquant le fait que l'établissement d'une connexion temporaire vers une entité SSF ou SRF assistante a échoué [par exemple recevoir une "libération vers l'arrière" après avoir envoyé un message d'adresse initial (IAM)].

3.2.1.4.2 Opérations SCF → SSF

- EstablishTemporaryConnection (Etablissement d'une connexion temporaire).

Procédures de l'entité d'appel (SCF)

- A) L'entité SCF envoie EstablishTemporaryConnection to SSF (Etablissement d'une connexion temporaire avec l'entité SSF)
- | | | |
|----------------|---------------|---|
| Précondition: | SCSM état 3.1 | Determine Mode (Déterminer le mode). |
| Postcondition: | SCSM état 3.2 | Waiting for AssistRequestInstructions (Attente des instructions de demande d'assistance). |
- B) L'entité SCF reçoit ETCFailed Error from SSF (Erreur d'établissement de connexion temporaire en provenance de l'entité SSF)
- | | | |
|----------------|---------------|--|
| Précondition: | SCSM état 3.2 | Waiting for AssistRequestInstructions. |
| Postcondition: | SCSM état 2.1 | Préparer les instructions SSF. |

Le traitement des erreurs dépend de la logique de traitement de service, par exemple en sélectionnant une autre entité SRF ou en poursuivant le traitement de l'appel.

Procédures de l'entité de réponse (SSF)

Une entité SSF reçoit EstablishTemporaryConnection à partir d'une entité SCF, mais l'établissement de la connexion échoue, ce qui aboutit à renvoyer une ETCFailed Error (Erreur due à l'échec d'établissement de connexion temporaire) vers l'entité SCF.

- | | | |
|----------------|----------------|---------------------------|
| Précondition: | SSF-FSM état c | Attente des instructions. |
| Postcondition: | SSF-FSM état c | Attente des instructions. |

Aucun autre traitement des erreurs.

3.2.1.5 ImproperCallerResponse (Réponse inadaptée du demandeur)

3.2.1.5.1 Description générale

3.2.1.5.1.1 Description d'erreur

Le format des informations fournies par l'utilisateur a été vérifié par l'entité SRF et ne correspond pas au format nécessaire tel qu'il a été défini dans le lancement de l'opération PromptAndCollectUserInformation.

3.2.1.5.2 Opérations SCF → SRF

- PromptAndCollectUserInformation (Recueil d'informations d'utilisateur sur invitation).

Procédures de l'entité d'appel (SCF)

- A) L'entité SCF envoie un message PromptAndCollectUserInformation vers l'entité SRF
- | | | |
|---------------|---------------|--|
| Précondition: | SCSM état 3.1 | Déterminer le mode; PromptAndCollectUserInformation accompagnera le message ConnectToResource. |
|---------------|---------------|--|

- | | | | |
|--|----------------|---------------|---|
| | ou | SCSM état 3.2 | Waiting for AssistRequestInstructions; après Establish-TemporaryConnection. |
| | ou | SCSM état 4.1 | Attente de la réponse en provenance de l'entité SRF; si d'autres PlayAnnouncements ou indications PromptAndCollectUser-Information sont à l'état actif. |
| | Postcondition: | SCSM état 4.1 | Attente de la réponse en provenance de l'entité SRF. |
- B) L'entité SCF reçoit une ImproperCallerResponse Error (réponse inappropriée du demandeur) en provenance de l'entité SRF
- | | | | |
|--|----------------|---------------|--|
| | Précondition: | SCSM état 4.1 | Attente de la réponse en provenance de l'entité SRF. |
| | Postcondition: | SCSM état 4.1 | Attente de la réponse en provenance de l'entité SRF. |

Le traitement des erreurs dépend de la logique de traitement de service. Une entité SCF peut déclencher une nouvelle Interaction avec l'utilisateur ou imposer une déconnexion (vers l'entité SSF).

Procédures de l'entité de réponse (SRF)

- A) L'entité SRF reçoit des messages PromptAndCollectUserInformation
- | | | | | |
|--|----------------|-------------|---------------------------------|---------------------------------|
| | Précondition: | SRSM état 2 | Connecté. | |
| | | ou | SRSM état 3 | Interaction avec l'utilisateur. |
| | Postcondition: | SRSM état 3 | Interaction avec l'utilisateur. | |
- B) La réponse du demandeur n'est pas correcte; l'entité SRF renvoie ImproperCallerResponse vers l'entité SCF
- | | | | |
|--|----------------|-------------|---------------------------------|
| | Précondition: | SRSM état 3 | Interaction avec l'utilisateur. |
| | Postcondition: | SRSM état 3 | Interaction avec l'utilisateur. |

L'entité SRF attend une nouvelle opération en provenance de l'entité SCF. Cela peut être une nouvelle opération PromptAndCollectUserInformation ou PlayAnnouncement.

3.2.1.6 MissingCustomerRecord (Absence d'enregistrement du client)

3.2.1.6.1 Description générale

3.2.1.6.1.1 Description d'erreur

Le programme de logique de traitement de service n'a pas pu être trouvé dans l'entité SCF, parce que l'enregistrement nécessaire du client n'existe pas, ou parce que l'instance de programme de logique de traitement de service demandée, indiquée par l'identificateur de corrélation dans "AssistRequestInstructions", n'existe plus. Il convient de distinguer ces deux cas comme deux situations différentes d'erreurs, étant donné que la procédure de traitement des erreurs indique que l'occurrence de l'erreur MissingCustomerRecord est signalée à la fonction de maintenance, mais que le compte rendu à la fonction de maintenance pour l'apparition du premier cas devrait être facultatif puisqu'il intervient non seulement dans des situations exceptionnelles, mais également dans des situations courantes. Par exemple, le premier cas peut se produire lorsque l'utilisateur compose un numéro manquant en "libre appel".

3.2.1.6.2 Opérations SSF → SCF

- AnalysedInformation;
- AssistRequestInstructions;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Procédures de l'entité d'appel (SSF)

A) Envoi de l'opération

Précondition:	SSF-FSM état b	Traitement du déclencheur.	
	ou	SSF-FSM état b'	Attente des instructions; en cas d'assistance/de transfert de tâche.
Postcondition:	SSF-FSM état c	Attente des instructions.	
	ou	SSF-FSM état b'	Attente des instructions; en cas d'assistance/de transfert de tâche.

B) L'entité SSF reçoit l'erreur "MissingCustomerRecord"

Précondition:	SSF-FSM état c	Attente des instructions.	
	ou	SSF-FSM état b'	Attente des instructions; en cas d'assistance/de transfert de tâche.
Postcondition:	SSF-FSM état a	Repos.	
	ou	SSF-FSM état a'	Repos; en cas d'assistance/de transfert de tâche.

L'entité CCF achemine l'appel si nécessaire (par exemple erreur d'acheminement vers une annonce terminale).

Procédures de l'entité de réponse (SCF)

Précondition:	1) SCSM	Etat approprié.
	2) SCSM	Opération reçue, événement approprié survenu.
Postcondition:	1) SCSM état 1	Repos; pour toutes les opérations énumérées ci-dessus, sauf AssistRequestInstructions.
	2) SCSM état 2.1	Préparer les instructions SSF; pour AssistRequest-Instructions.

L'entité SCSM indique que le programme nécessaire de logique de traitement de service (SLP) n'existe pas. L'instance de ce programme peut ne plus exister (par exemple dans le cas de l'opération AssistRequestInstructions), ou le programme de logique de traitement de service (SLP) peut n'avoir jamais existé du tout (c'est-à-dire que l'enregistrement du client dans l'entité SCF n'existe pas et, comme c'est par exemple le cas des points TDP, on essaie d'appeler un programme de logique de traitement de service). Le paramètre d'erreur MissingCustomerRecord est utilisé pour informer l'entité d'appel de cette situation. Les fonctions de maintenance sont averties (cependant, cela est facultatif dans le cas du point TDP).

3.2.1.6.3 Opérations SRF → SCF

- AssistRequestInstructions.

Procédures relatives à l'entité d'appel (SRF)

A) Envoi de l'opération

Précondition:	SRSM état 2	Connecté.
Postcondition:	SRSM état 2	Connecté.

B) L'entité SRF reçoit l'erreur "MissingCustomerRecord"

Précondition:	SRSM état 2	Connecté.
Postcondition:	SRSM état 1	Repos.

L'entité SRF a lancé l'opération Disconnect (Déconnecter).

Procédures de l'entité de réponse (SCF)

L'entité SCSM indique que le programme nécessaire de logique de traitement de service (SLP) n'existe plus. L'établissement d'une connexion entre l'entité SSF et l'entité SRF a duré trop longtemps ou l'identificateur de corrélation était erroné. Dans les deux cas, le programme demandé de logique de traitement de service ne peut pas être trouvé. Le paramètre d'erreur MissingCustomerRecord est utilisé pour informer l'entité d'appel de cette situation. Les fonctions de maintenance sont averties.

3.2.1.7 MissingParameter (Paramètre manquant)

3.2.1.7.1 Description générale

3.2.1.7.1.1 Description d'erreur

Il y a une erreur dans l'argument d'opération qui a été reçu. L'entité de réponse ne peut pas commencer à traiter l'opération demandée, parce que l'argument est incorrect: un paramètre obligatoire (l'application devra toujours renvoyer cette erreur au cas où elle ne serait pas détectée par le décodeur ASN.1) ou un paramètre facultatif prévu qui est essentiel dans l'application n'est pas inclus dans l'argument de l'opération.

3.2.1.7.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- Cancel;
- CancelStatusReportRequest;
- FurnishChargingInformation;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- RequestNotificationChargingEvent;
- RequestReportBCSMEEvent;
- ResetTimer;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- ConnectToResource;
- EstablishTemporaryConnection;
- HoldCallInNetwork;
- InitiateCallAttempt;
- SelectFacility;
- SelectRoute.

Procédures de l'entité d'appel (SCF)

A) Envoi de l'opération

Précondition:	SCSM	N'importe quel état dans lequel les opérations associées à l'appel ci-dessus peuvent être transférées.
	SCME	N'importe quel état dans lequel les opérations non associées à l'appel ci-dessus peuvent être transférées.
Postcondition:	SCSM	N'importe quel état résultant du transfert d'une quelconque des opérations ci-dessus.
Postcondition:	SCME	N'importe quel état résultant du transfert d'une quelconque des opérations ci-dessus non associées à l'appel.

B) L'entité SCF reçoit l'erreur "MissingParameter" (Paramètre manquant)

Précondition:	SCSM	N'importe quel état résultant du transfert d'une quelconque des opérations associées à l'appel ci-dessus.
	SCME	N'importe quel état résultant du transfert d'une quelconque des opérations ci-dessus non associées à l'appel.
Postcondition:	SCSM	Transition vers l'état initial (c'est-à-dire avant d'envoyer l'opération erronée).
	SCME	transition vers l'état initial (c'est-à-dire avant d'envoyer l'opération erronée).

Les fonctions de logique de traitement de service et de maintenance sont averties. Un autre traitement de l'appel dépend de la logique de traitement de service.

Procédures de l'entité de réponse (SSF)

Précondition	1) SSF-FSM	Etat approprié.
	2) SSF-FSM	Opération reçue associée à un appel, événement approprié survenu.
	3) SSME	Etat approprié.
	4) SSME	Opération reçue non associée à l'appel, événement approprié.
Postcondition:	1) SSF-FSM	Transition vers le même état.
	2) SSME	Transition vers l'état initial (c'est-à-dire avant de recevoir l'opération erronée).

La machine SSF-FSM décèle l'erreur contenue dans l'opération reçue. Le paramètre d'erreur est renvoyé pour informer l'entité SCF de la situation.

3.2.1.7.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- AssistRequestInstructions;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Procédures de l'entité d'appel (SSF)

A) Envoi de l'opération

Précondition:	SSF-FSM	N'importe quel état dans lequel les opérations ci-dessus peuvent être transférées.
Postcondition:	SSF-FSM	N'importe quel état résultant du transfert d'une quelconque des opérations ci-dessus.

B) L'entité SSF reçoit l'erreur "MissingParameter" (paramètre manquant)

Précondition:	SSF-FSM	N'importe quel état résultant du transfert d'une quelconque des opérations ci-dessus.
Postcondition:	SSF-FSM état a	Repos.

Après avoir reçu cette erreur, la machine SSF-FSM revient à l'état Repos. L'entité CCF achemine l'appel si nécessaire (erreur d'acheminement vers une annonce terminale). Si l'appel est déjà établi (par exemple déclenchement de demi-appel ou ApplyChargingReport), l'entité CCF maintient l'appel ou le déconnecte. Le choix entre ces deux options est spécifique à l'exploitant de réseau. Dans le cas d'une entité SSF assistante, la connexion temporaire est libérée par l'entité SSF assistante.

Procédures de l'entité de réponse (SCF)

Précondition:	1) SCSM	Etat approprié.
	2) SCSM	Opération reçue associée à un appel, événement approprié survenu.
Postcondition:	1) SCSM état 1	Repos; pour une des opérations citées ci-dessus, sauf AssistRequestInstructions.
	ou	
	2) SCSM état 2.1	Préparer les instructions SSF; dans le cas de l'opération AssistRequestInstructions.

L'entité SCSM décèle la situation erronée. Le paramètre d'erreur est utilisé pour informer l'entité SSF de cette situation. Les fonctions de logique de traitement de service et de maintenance sont averties.

3.2.1.7.4 Opérations SCF → SRF

- PlayAnnouncement;
- PromptAndCollectUserInformation;
- Cancel.

Procédures de l'entité d'appel (SCF)

A) Envoi de l'opération

Précondition:	SCSM état 3.1	Déterminer le mode; PromptAndCollectUserInformation ou l'opération PlayAnnouncement accompagnera l'opération ConnectToResource.
	ou	
	SCSM état 3.2	Waiting for AssistRequestInstructions; après EstablishTemporaryConnection.
	ou	
	SCSM état 4.1	Attente de la réponse en provenance de l'entité SRF; si d'autres PlayAnnouncements ou PromptAndCollectUserInformation sont en cours de traitement.
Postcondition:	SCSM état 4.1	Attente de la réponse en provenance de l'entité SRF.

B) Réception de l'erreur

Précondition:	SCSM état 4.1	Attente de la réponse en provenance de l'entité SRF.
Postcondition:	SCSM état 4.1	Attente de la réponse en provenance de l'entité SRF.

Le traitement des erreurs dépend de la logique de traitement de service. L'entité SCF peut lancer une nouvelle opération "Interaction avec l'utilisateur" ou imposer "Disconnect" (de l'entité SSF).

Procédures de l'entité de réponse (SRF)

Précondition:	SRSMS état 2	Connecté.
	ou	
	SRSMS état 3	Interaction avec l'utilisateur.
Postcondition:	SRSMS état 3	Interaction avec l'utilisateur.

L'entité SRSMS décèle le fait qu'un paramètre demandé n'est pas réalisé dans l'argument de l'opération. Le paramètre d'erreur MissingParameter est utilisé pour informer l'entité SCF de cette situation. L'entité SCF devrait effectuer les actions appropriées pour traiter cette erreur.

3.2.1.7.5 Opérations SRF → SCF

- AssistRequestInstructions.

Procédures de l'entité d'appel (SRF)

A) Envoi de l'opération

Précondition:	SRSMS état 2	Connecté.
Postcondition:	SRSMS état 2	Connecté.

B) Renvoi de l'erreur

Précondition:	SRSMS état 2	Connecté.
Postcondition:	SRSMS état 1	Repos.

Procédures de l'entité de réponse (SCF)

Précondition:	SCSM état 3.2	Waiting for AssistRequestInstructions (Attente des instructions de demande d'assistance).
Postcondition:	SCSM état 2.1	Préparer les instructions SSF.

L'entité SCSM décèle l'erreur dans l'opération reçue. Le paramètre d'erreur est utilisé pour informer l'entité SRF de cette situation. Les fonctions de logique de traitement de service et de maintenance sont averties. L'entité SCF pourrait essayer une autre entité SRF, acheminer ou libérer l'appel (selon la logique de traitement de service).

3.2.1.8 Erreur de nom

3.2.1.8.1 Description générale

3.2.1.8.1.1 Description d'erreur

L'erreur est envoyée par l'entité SDF vers l'entité SCF pour signaler un problème relatif au nom de l'objet. Les conditions dans lesquelles une erreur d'attribut doit être présentée sont définies en 12.5/X.511 (1993).

3.2.1.8.1.2 Description d'argument

Les codes de problèmes et de paramètres d'erreur de nom sont spécifiés en 12.5/X.511 (1993).

3.2.1.8.2 Opérations SCF → SDF

- Search;
- Add Entry;
- Remove Entry;
- Modify Entry.

Procédures de l'entité d'appel (SCF)

A) Envoi de l'opération

Précondition:	SCSM état 4	Rattachement SDF ou
	SCSM état 2	Attendre les demandes subséquentes.
Postcondition:	SCSM état 4	Rattachement SDF ou
	SCSM état 2	Attendre les demandes subséquentes.

B) Réception de l'erreur

Précondition:	SCSM état 4	Rattachement SDF.
Postcondition:	SCSM état 4	Rattachement SDF.

La procédure d'erreur dépend de la logique de traitement de service. Si l'entité SCF est à même de changer la demande, elle peut faire une nouvelle interrogation SDF, sinon il serait mis fin au traitement de service.

Procédures de l'entité de réponse (SDF)

A) Réception de l'opération

Précondition:	SDF-FSM état 3	Rattachement SCF.
Postcondition:	SDF-FSM état 3	Rattachement SCF.

B) Renvoi de l'erreur

Précondition:	SDF-FSM état 3	Rattachement SCF.
Postcondition:	SDF-FSM état 3	Rattachement SCF.

L'entité SDF n'a pas pu exécuter l'opération en raison d'un problème relatif au nom et elle envoie de ce fait une erreur Name (Nom) à l'entité SCF. Une fois que l'erreur a été renvoyée, aucun autre traitement des erreurs n'intervient.

3.2.1.9 ParameterOutOfRange (Paramètre hors gamme)

3.2.1.9.1 Description générale

3.2.1.9.1.1 Description d'erreur

L'entité de réponse ne peut pas commencer le traitement de l'opération demandée, parce qu'une erreur est décelée dans un paramètre de l'argument d'opération: une valeur de paramètre est hors gamme. Cette erreur est appliquée dans les deux cas suivants (quand l'erreur est définie par l'application):

- 1) Pour le paramètre dont le type est défini avec la gamme correspondant à sa taille, comme INTEGER (NOMBRE ENTIER) (x..y), SEQUENCE SIZE (LONGUEUR DE SÉQUENCE) (x..y) OF Type. Cette erreur s'applique quand la valeur du paramètre est z ou quand la taille de paramètre est z, avec $z < x$ ou $z > y$.
- 2) Pour le paramètre dont le type est défini comme liste de valeurs ENUMERATED (ÉNUMÉRÉ), l'erreur ParameterOutOfRange (Paramètre hors gamme) s'applique quand la valeur du paramètre n'est égale à aucune des valeurs ENUMERATED de la liste.

3.2.1.9.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- InitiateCallAttempt;
- RequestNotificationChargingEvent;
- RequestReportBCSMEEvent;
- ResetTimer;
- SelectFacility;
- SelectRoute.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur appropriées.

3.2.1.9.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TCalledPartyBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) en ce qui concerne les procédures d'erreur appropriées.

3.2.1.9.4 Opérations SCF → SRF

- PlayAnnouncement;
- PromptAndCollectUserInformation.

Il convient de se référer au 3.2.1.7 en ce qui concerne les procédures d'erreur applicables.

3.2.1.10 RequestedInfoError (Erreur d'information demandée)

3.2.1.10.1 Description générale

3.2.1.10.1.1 Description d'erreur

L'erreur RequestedInfoError est une réponse immédiate à l'opération CallInformationRequest, indiquant que l'information demandée n'est pas connue de l'entité SSF ou n'est pas disponible. L'erreur RequestedInfoError est utilisée quand une entité SSF/CCF spécifique ne peut pas fournir l'information spécifiée par RequestedInformationType bien qu'il y ait une autre SSF/CCF qui puisse apporter cette information.

3.2.1.10.1.2 Description d'argument

```
PARAMETER ENUMERATED {  
    unknownRequestedInfo (1),  
    requestedInfoNotAvailable (2)  
    -- autres valeurs à l'étude  
}
```

3.2.1.10.2 Opérations SCF → SSF

- CallInformationRequest.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) en ce qui concerne les procédures d'erreur applicables.

3.2.1.11 Erreur dans la prestation de service

3.2.1.11.1 Description générale

3.2.1.11.1.1 Description d'erreur

Cette erreur est envoyée par l'entité SDF vers l'entité SCF pour signaler un problème associé au service fourni. Les conditions dans lesquelles une erreur de service doit être présentée sont définies en 12.8/X.511 (1993).

3.2.1.11.1.2 Description d'argument

Le paramètre d'erreur de service et les codes d'incidents sont spécifiés en 12.8/X.511 (1993).

3.2.1.11.2 Opérations SCF → SDF

- Bind;
- Search;
- Add Entry;
- Remove Entry;
- Modify Entry.

Procédures de l'entité d'appel (SCF)

A) Envoi de l'opération

Précondition:	SCSM état 1	Idle (en cas de rattachement).	
	ou	SCSM état 2	Attente des demandes subséquentes (en cas d'opérations sauf l'opération de rattachement).
	ou	SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).

	Postcondition:	SCSM état 2	Attente des demandes subséquentes (en cas de toutes les opérations possibles).
		ou SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).
B)	Recevoir l'erreur		
	Précondition:	SCSM état 3	Attente du résultat de l'opération de rattachement (en cas de rattachement).
		ou SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).
	Postcondition:	SCSM état 1	Idle (en cas de rattachement).
		ou SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).

La procédure d'erreur dépend de la logique de traitement de service. S'il existe une autre entité SDF, il peut être possible de procéder à une autre interrogation, sinon il convient de mettre un terme au traitement du service.

Procédures de l'entité de réponse (SDF)

A)	Réception de l'opération		
	Précondition:	SDF-FSM état 1	Idle (en cas de rattachement).
		ou SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).
	Postcondition:	SDF-FSM état 2	Rattachement en cours (en cas de rattachement).
		ou SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).
B)	Renvoi de l'erreur		
	Précondition:	SDF-FSM état 2	Rattachement en cours (en cas de rattachement).
		ou SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).
	Postcondition:	SDF-FSM état 1	Idle (en cas de rattachement).
		ou SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).

L'entité SDF n'a pas pu exécuter l'opération en raison d'un problème lié à la logique de service et envoie une erreur de prestation de service à l'entité SCF. Un fois que l'erreur a été renvoyée, aucun autre traitement des erreurs n'est effectué.

3.2.1.12 Erreur de sécurité

3.2.1.12.1 Description générale

3.2.1.12.1.1 Description d'erreur

Cette erreur est envoyée par l'entité SDF vers l'entité SCF pour des raisons de sécurité afin de signaler un problème pour des raisons de sécurité lorsqu'une opération est effectuée. Les conditions dans lesquelles une erreur de sécurité doit être présentée sont définies en 12.7/X.511 (1993).

3.2.1.12.1.2 Description d'argument

Le paramètre d'erreur de sécurité et les codes d'incidents sont spécifiés en 12.7/X.511 (1993).

3.2.1.12.2 Opérations SCF → SDF

- Bind;
- Search;
- Add Entry;
- Remove Entry;
- Modify Entry.

Procédures de l'entité d'appel (SCF)

A) Envoi de l'opération

Précondition:	SCSM état 1	Idle (en cas de rattachement).	
	ou	SCSM état 2	Attente des demandes subséquentes (en cas d'opérations sauf l'opération de rattachement).
	ou	SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).
Postcondition:	SCSM état 2	Attente des demandes subséquentes (en cas de toutes les opérations possibles).	
	ou	SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).

B) Recevoir l'erreur

Précondition:	SCSM état 3	Attente du résultat de l'opération de rattachement (en cas de rattachement).	
	ou	SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).
Postcondition:	SCSM état 1	Idle (en cas de rattachement).	
	ou	SCSM état 4	Rattachement SDF (en cas d'opérations sauf l'opération de rattachement).

La procédure d'erreur est indépendante de la logique de traitement de service. Il convient de mettre un terme au traitement du service.

Procédure de l'entité de réponse (SDF)

A) Réception de l'opération

Précondition:	SDF-FSM état 1	Idle (en cas de rattachement).	
	ou	SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).
Postcondition:	SDF-FSM état 2	Rattachement en cours (en cas de rattachement).	
	ou	SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).

B) Renvoi de l'erreur

Précondition:	SDF-FSM état 2	Rattachement en cours (en cas de rattachement).	
	ou	SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).
Postcondition:	SDF-FSM état 1	Idle (en cas de rattachement).	
	ou	SDF-FSM état 3	Rattachement SCF (en cas d'opérations sauf l'opération de rattachement).

L'entité SDF n'a pas pu exécuter l'opération pour des raisons de sécurité et envoie donc une erreur de sécurité vers l'entité SCF. Un fois que l'erreur a été renvoyée, aucun autre traitement des erreurs n'est effectué.

3.2.1.13 SystemFailure (Echec du système)

3.2.1.13.1 Description générale

3.2.1.13.1.1 Description d'erreur

Cette erreur est renvoyée par une entité physique si elle n'a pas pu effectuer une tâche spécifique comme le demande une opération, et il n'est pas prévu de pouvoir revenir à l'état normal dans l'instance d'appel en cours (ce point nécessite une étude complémentaire).

3.2.1.13.1.2 Description d'argument

PARAMETER

UnavailableNetworkResource

UnavailableNetworkResource ::= ENUMERATED {
 unavailableResources (0),
 componentFailure (1),
 basicCallProcessingException (2),
 resourceStatusFailure (3),
 endUserFailure (4)}

3.2.1.13.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- CancelStatusReportRequest;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- RequestNotificationChargingEvent;
- RequestReportBCSMEvent;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- ConnectToResource;
- DisconnectForwardConnection;
- EstablishTemporaryConnection;
- HoldCallInNetwork;
- InitiateCallAttempt;
- SelectFacility;
- SelectRoute.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.13.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- CollectedInformation;
- InitialDP;
- OAnswer
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.13.4 Opérations SCF → SRF

- Cancel;
- PlayAnnouncement;
- PromptAndCollectUserInformation.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.13.5 Opérations SRF → SCF

- AssistRequestInstructions

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.14 TaskRefused (Tâche refusée)

3.2.1.14.1 Description générale

3.2.1.14.1.1 Description d'erreur

Cette erreur est renvoyée par une entité physique si elle n'a pas pu effectuer une tâche spécifique comme le demande une opération, et il n'est pas prévu de pouvoir revenir à l'état normal dans l'actuelle instance d'appel en cours (la définition précise nécessite une étude complémentaire).

3.2.1.14.1.2 Description d'argument

```
PARAMETER ENUMERATED {  
    generic (0),  
    unobtainable (1),  
    congestion (2)  
    -- autres valeurs à l'étude  
}
```

3.2.1.14.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- Cancel;
- CancelStatusReportRequest;
- FurnishChargingInformation;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- RequestNotificationChargingEvent;
- RequestReportBCSMEvent;
- ResetTimer;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- ConnectToResource;
- DisconnectForwardConnection;
- EstablishTemporaryConnection;
- HoldCallInNetwork;
- InitiateCallAttempt;
- SelectFacility;
- SelectRoute.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.14.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- AssistRequestInstructions;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.14.4 Opérations SCF → SRF

- Cancel;
- PlayAnnouncement;
- PromptAndCollectUserInformation.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.14.5 Opérations SRF → SCF

- AssistRequestInstructions.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.15 UnavailableResource (Ressource indisponible)

3.2.1.15.1 Description générale

3.2.1.15.1.1 Description d'erreur

L'entité SRF n'est pas à même d'accomplir sa fonction (c'est-à-dire passage d'une certaine annonce et/ou recueil d'une information spécifique de l'utilisateur) et ne peut pas être remplacée. Une nouvelle tentative n'est pas possible.

3.2.1.15.2 Opérations SCF → SRF

- PlayAnnouncement;
- PromptAndCollectUserInformation.

Procédures de l'entité d'appel (SCF)

- A) L'entité SCF envoie l'une des opérations PlayAnnouncement ou PromptAndCollectUserInformation vers l'entité SRF

Précondition:	SCSM état 3.1	Déterminer le mode; l'une des opérations PlayAnnouncement ou PromptAndCollectUserInformation accompagnera le message ConnectToResource.	
	ou	SCSM état 3.2	Waiting for AssistRequestInstructions; après Establish-TemporaryConnection.
	ou	SCSM état 4.1	Attente de la réponse en provenance de l'entité SRF; si d'autres opérations PlayAnnouncements ou PromptAndCollectUser-Information sont en cours.
Postcondition:	SCSM état 4.1	Attente de la réponse en provenance de l'entité SRF.	

- B) L'entité SCF reçoit l'erreur UnavailableResource en provenance de l'entité SRF
- | | | |
|----------------|---------------|---|
| Précondition: | SCSM état 4.1 | Attente de la réponse à partir de l'entité SRF. |
| Postcondition: | SCSM état 4.1 | Attente de la réponse à partir de l'entité SRF. |

Si la ressource choisie ne peut pas exécuter sa fonction, un autre traitement dépend de la logique de service.

- Exemples:
- demande SSF de se connecter à une autre entité SRF possible;
 - traitement de service sans l'opération PlayAnnouncement ou PromptAndCollectUserInformation (le cas échéant);
 - mettre fin au traitement du service.

Procédures de l'entité qui répond (SRF)

- A) L'entité SRF reçoit l'opération PlayAnnouncement ou PromptAndCollectUserInformation
- | | | |
|---------------|----------------|--|
| Précondition: | SRSM état 2 | Connecté; s'il y a une opération PlayAnnouncement ou PromptAndCollectUserInformation initiale. |
| | ou SRSM état 3 | Interaction avec l'utilisateur; s'il n'y a pas d'opération PlayAnnouncement ou PromptAndCollectUserInformation initiale. |
- B) L'entité SRF n'est pas à même d'exécuter sa fonction (et ne peut pas être remplacée). L'entité SRF envoie UnavailableResource (ressource indisponible)
- | | | |
|----------------|-------------|---------------------------------|
| Précondition: | SRSM état 3 | Interaction avec l'utilisateur. |
| Postcondition: | SRSM état 3 | Interaction avec l'utilisateur. |

3.2.1.16 UnexpectedComponentSequence (séquence composante imprévue)

3.2.1.16.1 Description générale

3.2.1.16.1.1 Description d'erreur

L'entité de réponse ne peut pas commencer le traitement de l'opération demandée parce que les règles applicables aux entités SACF ou MACF ne sont pas observées, ou bien l'opération n'a pas pu être traitée en l'état actuel de la machine FSM.

3.2.1.16.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- FurnishChargingInformation;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- RequestNotificationChargingEvent;
- RequestReportBCSMEvent;
- ResetTimer;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- ConnectToResource;
- DisconnectForwardConnection;
- EstablishTemporaryConnection;
- HoldCallInNetwork;
- InitiateCallAttempt;
- SelectFacility;
- SelectRoute.

Dans ce cas, l'entité SSF décèle la situation erronée, envoie l'erreur UnexpectedComponentSequence et reste dans le même état. Dans l'entité SCF, les fonctions de logique de traitement de service et de maintenance sont averties et la logique de traitement de service décide du traitement des erreurs.

3.2.1.16.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- AssistRequestInstructions;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Dans le cas d'une entité SSF assistante, si une erreur survient au cas où l'opération AssistRequestInstructions est envoyée, alors qu'une relation a déjà été établie entre une entité SCF et une entité SSF assistante, l'entité SCF renvoie le paramètre d'erreur et en informe la logique de traitement de service et la maintenance. A la réception de l'erreur, l'entité SSF assistante se positionne en position Repos et la connexion temporaire est libérée.

Au cas où l'opération serait envoyée par une entité SSF "déclencheuse" dans le contexte d'une relation existante, l'entité SCF renverrait le paramètre d'erreur. La logique de traitement de service et la maintenance seraient averties. A la réception de l'erreur, l'entité SSF va à l'état Repos.

3.2.1.16.4 Opérations SCF → SRF (seulement applicable dans le cas d'une relation directe SCF-SRF)

- PlayAnnouncement;
- PromptAndCollectUserInformation.

Dans ce cas, l'entité SRF décèle la situation erronée, envoie l'erreur UnexpectedComponentSequence et reste dans le même état. Dans l'entité SCF, les fonctions de logique de traitement de service et de maintenance sont averties et la logique de traitement de service prend des décisions relatives au traitement des erreurs. Un éventuel traitement des erreurs serait d'envoyer l'opération DisconnectForwardConnection (déconnecter la connexion vers l'avant) à l'entité SSF.

3.2.1.16.5 Opérations SRF → SCF

- AssistRequestInstructions.

Dans ce cas, une erreur survient si l'entité SRF a déjà établi une relation avec l'entité SCF et envoie une opération AssistRequestInstructions. L'entité SCF détecte la situation erronée, informe la logique de traitement de service et les fonctions de maintenance et renvoie le paramètre d'erreur. A la réception du paramètre, l'entité SRF se positionne en position Repos et libère la connexion temporaire.

3.2.1.17 UnexpectedDataValue (valeur imprévue d'une donnée)

3.2.1.17.1 Description générale

3.2.1.17.1.1 Description des erreurs

L'entité de réponse ne peut pas achever le traitement de l'opération demandée parce qu'un paramètre a une valeur d'une donnée non prévue.

Il convient de noter que cette erreur est différente du "paramètre hors gamme".

Exemple: **startTime DateAndTime ::= -- valeur indiquée 32 janvier 1993, 12:15:01**

L'entité de réponse ne prévoit pas cette valeur et répond par "UnexpectedDataValue".

3.2.1.17.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- Cancel;
- CancelStatusReportRequest;
- FurnishChargingInformation;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- RequestNotificationChargingEvent;
- RequestReportBCSMEvent;
- ResetTimer;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- ConnectToResource;
- EstablishTemporaryConnection;
- HoldCallInNetwork;
- InitiateCallAttempt;
- SelectFacility;
- SelectRoute.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.17.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- AssistRequestInstructions;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;
- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.17.4 Opérations SCF → SRF

- Cancel;
- PlayAnnouncement;
- PromptAndCollectUserInformation.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.17.5 Opérations SRF → SCF

- AssistRequestInstructions.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.18 UnexpectedParameter (Paramètre non prévu)

3.2.1.18.1 Description générale

3.2.1.18.1.1 Description d'erreur

Il y a une erreur dans l'argument d'opération qui a été reçu. Un paramètre correct mais imprévu était présent dans l'argument d'opération. La présence de ce paramètre n'est pas compatible avec la présence d'autres paramètres. L'entité de réponse ne peut pas commencer à traiter l'opération.

3.2.1.18.2 Opérations SCF → SSF

Mode non associé à un appel

- ActivateServiceFiltering.

Mode associé à un appel/Traitement de non-appel

- ApplyCharging;
- CallInformationRequest;
- FurnishChargingInformation;
- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport;
- RequestNotificationChargingEvent;
- RequestReportBCSMEvent;
- ResetTimer;
- SendChargingInformation.

Mode associé à un appel/Traitement d'appel

- AnalyseInformation;
- CollectInformation;
- Connect;
- ConnectToResource;
- EstablishTemporaryConnection;
- HoldCallInNetwork;
- InitiateCallAttempt;
- SelectFacility;
- SelectRoute.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.18.3 Opérations SSF → SCF

- AnalysedInformation;
- ApplyChargingReport;
- AssistRequestInstructions;
- CollectedInformation;
- InitialDP;
- OAnswer;
- OCalledPartyBusy;
- ODisconnect;
- OMidCall;
- ONoAnswer;
- OriginationAttemptAuthorized;
- RouteSelectFailure;
- TAnswer;
- TBusy;

- TDisconnect;
- TermAttemptAuthorized;
- TMidCall;
- TNoAnswer.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.18.4 Opérations SCF → SRF

- PlayAnnouncement;
- PromptAndCollectUserInformation.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.18.5 Opérations SRF → SCF

- AssistRequestInstructions.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.19 UnknownLegID (Identificateur de demi-appel inconnu)

3.2.1.19.1 Description générale

3.2.1.19.1.1 Description d'erreur

Cette erreur est utilisée pour indiquer à l'entité SCF qu'une branche spécifique, indiquée par la valeur du paramètre LegID dans l'opération, est inconnue de l'entité SSF.

3.2.1.19.2 Opérations SCF → SSF

Mode associé à un appel/Traitement de non-appel

- SendChargingInformation.

Il convient de se référer au 3.2.1.7 MissingParameter (Paramètre manquant) pour les procédures d'erreur applicables.

3.2.1.20 UnknownResource (Ressource inconnue)

3.2.1.20.1 Description générale

3.2.1.20.1.1 Description d'erreur

Cette erreur est utilisée pour indiquer à l'entité SCF qu'une ressource physique particulière qui est indiquée par le paramètre ResourceID n'est pas connue de l'entité SSF.

3.2.1.20.2 Opérations SCF → SSF

Mode associé à un appel/Traitement de non-appel

- RequestCurrentStatusReport;
- RequestEveryStatusChangeReport;
- RequestFirstStatusMatchReport.

Se référer au 3.2.1.7 Paramètre manquant pour les procédures d'erreur appropriées.

3.2.1.21 Erreur de mise à jour

3.2.1.21.1 Description générale

3.2.1.21.1.1 Description d'erreur

Cette erreur est envoyée par l'entité SDF vers l'entité SCF afin de signaler un problème lié à des tentatives d'ajouter, de supprimer ou de modifier des informations vers l'entité SDF. Les conditions dans lesquelles une erreur de mise à jour doit être présentée sont définies en 12.9/X.511 (1993).

3.2.1.21.1.2 Description d'argument

Le paramètre d'erreur de mise à jour et les codes d'incidents sont spécifiés en 12.9/X.511 (1993).

3.2.1.21.2 Opérations SCF → SDF

- Add Entry;
- Remove Entry;
- Modify Entry.

Procédures de l'entité d'appel (SCF)

A) Opération d'envoi

Précondition:	SCSM état 4	Rattachement SDF ou
	SCSM état 2	Attendre les demandes subséquentes.
Postcondition:	SCSM état 4	Rattachement SDF ou
	SCSM état 2	Attendre les demandes subséquentes.

B) Erreur de réception

Précondition:	SCSM état 4	Rattachement SDF.
Postcondition:	SCSM état 4	Rattachement SDF.

La procédure de traitement des erreurs dépend de la logique de service.

Procédures à l'entité de réponse (SDF)

A) Opération de réception

Précondition:	SDF-FSM état 3	Rattachement SCF.
Postcondition:	SDF-FSM état 3	Rattachement SCF.

B) Retour d'une indication d'erreur

Précondition:	SDF-FSM état 3	Rattachement SCF.
Postcondition:	SDF-FSM état 3	Rattachement SCF.

L'entité SDF n'a pas pu exécuter l'opération en raison d'un problème lié à l'ajout, la suppression ou la modification des informations et envoie une indication d'erreur de mise à jour à l'entité SCF. Une fois que l'indication d'erreur a été renvoyée, aucun autre traitement d'erreur n'intervient.

3.2.2 Procédures d'erreur relatives aux entités

Les paragraphes suivants définissent le traitement des erreurs relatives aux entités. Etant donné que les situations d'erreur ne proviennent pas de la réception d'une opération, l'entité d'appel est indiquée ici comme l'entité dans laquelle la situation d'erreur est détectée. L'entité de réponse est celle qui reçoit le compte rendu d'erreurs.

Les services du sous-système TCAP utilisés pour signaler les erreurs sont décrits en 3.4.

3.2.2.1 Fin de temporisation T_{SSF}

3.2.2.1.1 Description générale

3.2.2.1.1.1 Description d'erreur

Une fin de temporisation est intervenue dans l'entité SSF à la réponse provenant de l'entité SCF.

3.2.2.1.2 Procédures SSF → SCF

Procédures de l'entité d'appel (SSF)

La temporisation intervient dans l'entité SSF par le temporisateur T_{SSF} .

Précondition:	SSF-FSM état c	Attente des instructions.	
	ou	SSF-FSM état d	Attente de la fin de l'interaction avec l'utilisateur.
	ou	SSF-FSM état e	Attente de la fin de la connexion temporaire.
Postcondition:	SSF-FSM état a	Repos.	

La machine SSF-FSM interrompt le dialogue et passe à l'état Repos, l'entité CCF acheminant si nécessaire l'appel (par exemple acheminement par défaut vers une annonce de fin d'appel). L'interruption est signalée aux fonctions de maintenance.

Procédures de l'entité qui répond (SCF)

L'entité SCF reçoit une interruption de dialogue.

Précondition:	n'importe quel état.		
Postcondition:	SCSM état 1	Repos; si l'interruption concerne un dialogue SSF.	
	ou	SCSM état 2	Préparer les instructions SSF; si l'interruption concerne un dialogue de l'entité SSF assistante.

L'entité SCF libère toutes les ressources attribuées et signale l'interruption aux fonctions de maintenance, dans le cas où l'interruption est reçue à partir d'un dialogue SSF. L'entité SCF libère toutes les ressources nécessaires au dialogue, rend compte de l'interruption aux fonctions de maintenance et revient à l'état "Préparer les instructions SSF", dans le cas où l'interruption est reçue au cours d'un dialogue de l'entité SSF assistante.

3.2.2.2 Fin de la temporisation d'application T_{SRF}

3.2.2.2.1 Description générale

3.2.2.2.1.1 Description d'erreur

Une fin de temporisation est intervenue dans l'entité SRF à la réponse provenant de l'entité SCF. Cette procédure concerne uniquement le cas d'une relation directe SCF-SRF.

3.2.2.2.2 Procédures SRF → SCF

Procédures de l'entité d'appel (SRF)

La fin de temporisation de T_{SRF} survient dans l'entité SRF.

Précondition:	SRSM état 2	Connecté.	
	ou	SRSM état 3	Interaction avec l'utilisateur.
Postcondition:	SRSM état 1	Repos.	

L'entité SRF interrompt le dialogue et revient à l'état Repos, toutes les ressources attribuées étant libérées. L'interruption est signalée aux fonctions de maintenance.

Procédures de l'entité de réponse (SCF)

L'entité SCF reçoit une interruption de dialogue.

Précondition:	SCSM état 4	Interaction avec l'utilisateur.
Postcondition:	SCSM état 2	Préparer les instructions SSF.

L'entité SCF libère toutes les ressources liées au dialogue, signale l'interruption aux fonctions de maintenance et revient à l'état "Préparer les instructions SSF".

3.3 Procédures détaillées d'opération

L'usage prévu des opérations et des paramètres figurant dans l'article 2 qui ne figure pas dans les descriptions détaillées de procédures qui suivent doit faire l'objet d'une étude complémentaire. Pour obtenir davantage d'informations, il convient de se référer au 6.4/Q.1214.

3.3.1 ActivateServiceFiltering procedure (Procédure d'activation du service de filtrage)

3.3.1.1 Description générale

A la réception de cette opération, l'entité SSF traite les appels vers les différentes destinations de façon spécifique sans demander d'instructions vers l'entité SCF. En cas de filtrage de service, l'entité SSF exécute un algorithme spécial de filtrage de service. Pour le transfert du filtrage de service, les résultats font référence à l'opération "Réponse de filtrage de service".

3.3.1.1.1 Paramètres

a) filteredCallTreatment (Traitement d'appel filtré):

ce paramètre spécifie comment les appels filtrés doivent être traités. Il comprend des informations relatives à l'annonce à transmettre, de même qu'au chargement, au nombre de compteurs utilisés et à la cause de libération qui doit être fournie pour les appels filtrés.

- i) **sFBillingChargingCharacteristics** (Paramètres de taxation et de facturation):
ce paramètre détermine la taxation à appliquer pour le service de filtrage. Sa valeur est spécifique au réseau.
NOTE – Le format et le codage font l'objet d'une étude complémentaire.
- ii) **informationToSend** (Informations à envoyer):
ce paramètre indique une annonce, une tonalité ou une information à visualiser devant être envoyée au demandeur. A la fin de la transmission des informations, l'appel est libéré.
- **inbandInfo** (Informations dans la bande):
ce paramètre spécifie quelles informations dans la bande doivent être envoyées.
 - **messageID** (identificateur de messages):
ce paramètre indique le(s) message(s) devant être envoyé(s) parmi les messages suivants:
 - 1) **elementaryMessageID** (identificateur de message élémentaire):
ce paramètre indique une annonce unique.
 - 2) **texte**:
ce texte indique qu'un texte doit être envoyé. Il sera transformé en informations dans la bande (parole). Ce paramètre comprend deux sous-paramètres, **messageContent** (contenu de message) et les attributs. Les attributs du texte peuvent comprendre des articles tels que le langage.
 - 3) **elementaryMessageIDs** (Identificateurs de message élémentaire):
ce paramètre spécifie une séquence d'annonces.
 - 4) **variableMessage** (Message variable):
ce paramètre spécifie une annonce comprenant une ou plusieurs parties de longueur variable.
 - **numberOfRepetitions** (nombre de répétitions):
ce nombre indique combien de fois au plus le message sera envoyé à l'utilisateur.
 - **durée**:
ce nombre indique pendant quelle durée maximale en secondes le message devra être transmis/répété. Le ZÉRO indique que la répétition est sans fin.
 - **intervalle**:
ce paramètre indique l'intervalle de temps en secondes séparant les répétitions, c'est-à-dire le temps qui s'est écoulé entre la fin de l'annonce et le début de la répétition suivante. Ce paramètre peut uniquement être utilisé si le nombre de répétitions est > 1.
 - **tonalité**:
ce paramètre spécifie une tonalité devant être envoyée à l'utilisateur.
 - **toneID** (identificateur de tonalité):
ce paramètre indique quelle tonalité doit être envoyée.
 - **durée**:
ce paramètre indique la durée en secondes de la tonalité qui doit être envoyée. Le ZÉRO indique une durée infinie.
 - **displayInformation**:
ce paramètre indique une chaîne de texte qui doit être envoyée à l'utilisateur. Ces informations ne peuvent pas être reçues par l'utilisateur du RTPC.
- iii) **maximumNumberOfCounters** (Nombre maximal de compteurs):
ce paramètre indique le nombre de compteurs devant être attribués, de même que le nombre de destinations pour le filtrage de service, c'est-à-dire que le "maximumNumberOfCounters" (le nombre maximal de compteurs) des adresses subséquentes de destination, commençant à partir de l'adresse de destination fournie dans les "filteringCriteria" (critères de filtrage) sont utilisés pour le filtrage de service. Un compteur est attribué à chacune de ces adresses de destination;

le nombre de compteurs peut seulement être > 1, si les critères de filtrage sont du type "addressAndService".

iv) releaseCause (Cause de libération):

ce paramètre fournit la valeur de la cause utilisée pour la libération de l'appel après que "InformationToSend" (l'information devant être envoyée) (par exemple l'annonce) a été envoyée au demandeur. En cas d'absence de la "cause de la libération", la valeur par défaut est la même que la valeur décimale 31 utilisée dans la signalisation ISUP (valeur normale non spécifiée).

b) filteringCharacteristics (Caractéristiques de filtrage):

ce paramètre indique la sévérité du filtrage et le moment précis sur l'échelle de temps où l'opération "ServiceFilteringResponse" (Réponse de filtrage de service) devra être envoyée. Il détermine également si "l'intervalle" ou le nombre d'appels ("numberOfCalls") est utilisé.

i) intervalle:

après la fin de la temporisation, l'appel suivant déclenche les actions suivantes:

- envoi d'un DP initial ou d'une opération spécifique au DP;
- envoi d'une "ServiceFilteringResponse" (réponse du filtrage de service);
- redémarrage de la temporisation.

Le premier intervalle commence dès le début du filtrage.

Un intervalle de 0 indique que tous les appels satisfaisant aux critères de filtrage permettront d'envoyer un DP initial ou une opération spécifique au DP et qu'aucun filtrage ne sera appliqué (c'est-à-dire qu'aucune réponse ServiceFilteringResponse ne sera envoyée).

Un intervalle de -1 indique qu'aucun des appels satisfaisant aux critères de filtrage ne permettra d'envoyer ni un point DP initial ni une opération spécifique au point DP ni une opération "ServiceFilteringResponse".

Les autres valeurs indiquent une durée en secondes.

ii) numberOfCalls:

le *n*ème cas provoque l'envoi d'un DP initial ou d'une opération spécifique au point DP et une opération "ServiceFilteringResponse" est envoyée à l'entité SCF. Cette valeur seuil est atteinte si la somme de tous les compteurs attribués à une entité de filtrage de service est égale au "numberOfCalls" (nombre d'appels);

un nombre d'appels de 0 indique qu'aucun des appels satisfaisant aux critères de filtrage n'aura pour résultat d'envoyer un DP initial ou une opération spécifique au point DP et une opération "ServiceFilteringResponse".

c) filteringTimeOut (Temporisation du filtrage):

ce paramètre indique la durée du filtrage. Au terme de la temporisation, une réponse "ServiceFilteringResponse" est envoyée à l'entité SCF et le filtrage de service est arrêté. Deux méthodes sont préconisées (durée ou stopTime):

i) durée:

si la durée est atteinte, le filtrage de service est arrêté et le compte rendu final est envoyé à l'entité SCF;

une durée de 0 indique que le filtrage de service doit être enlevé;

une durée de -1 indique une durée illimitée;

une durée de -2 indique une durée spécifique à un réseau;

les autres valeurs indiquent une durée en secondes.

ii) stopTime (Temps d'arrêt):

quand la valeur du paramètre stopTime (temps d'arrêt) est atteinte, le filtrage de service est arrêté et le compte rendu final est envoyé à l'entité SCF. Si la valeur du paramètre "stopTime" est déjà atteinte, c'est-à-dire quand la valeur stopTime est inférieure à la valeur de la durée effective, mais la différence ne dépasse pas une valeur équivalant à cinquante années, le filtrage de service est immédiatement interrompu et les valeurs réelles de compteur sont indiquées à l'entité SCF. Cela se produit dans les cas où l'entité SCF essaie explicitement d'arrêter un filtrage de service en cours.

d) **filteringCriteria** (Critères de filtrage):

ce paramètre spécifie quels sont les appels qui doivent être filtrés sur la base de "serviceKey" (clé de service), "callingAddressValue" (valeur d'adresse du demandeur), "calledAddressValue" (valeur d'adresse du demandé) ou "locationNumber" (numéro d'emplacement). Il s'agit d'effectuer un choix entre "serviceKey" (clé de service) ou "addressAndService" (adresse et service).

i) **serviceKey** (Clé de service):

ce paramètre indique sans ambiguïté le service de réseau intelligent demandé auquel le filtrage devrait être appliqué.

ii) **addressAndService** (Adresse et service):

ce paramètre identifie le service de réseau intelligent et le numéro composé auxquels le filtrage devrait être appliqué. La zone géographique peut également être identifiée ("callingAddressValue" et/ou "locationNumber").

iii) **calledAddressValue** (Valeur de l'adresse du demandé):

ce paramètre contient le numéro composé en direction duquel le filtrage devra être appliqué. Le numéro complet du demandé sera spécifié.

iv) **serviceKey** (Clé de service):

ce paramètre identifie sans ambiguïté le service de réseau intelligent demandé auquel le filtrage devrait être appliqué.

v) **callingAddressValue** (Valeur de l'adresse du demandeur):

ce paramètre contient le numéro du demandeur qui identifie l'appelant ou l'origine géographique de l'appel pour lequel le filtrage devra être appliqué.

vi) **locationNumber** (Numéro d'emplacement):

ce paramètre indique la zone géographique d'où provient l'appel devant être filtré. Il est utilisé quand "callingAddressValue" ne contient pas d'informations relatives à l'emplacement géographique du demandeur.

e) **startTime** (Heure de début):

ce paramètre définit le moment où commence le filtrage. Si le paramètre "startTime" n'existe pas encore ou si l'heure est déjà dépassée, l'entité SSF commence immédiatement le filtrage.

3.3.1.2 Entité d'appel (SCF)

3.3.1.2.1 Procédure normale

Précondition SCF:

- l'instance SLPI décèle que le filtrage de service doit être déclenché dans l'entité SSF.

Postconditions SCF:

- 1) l'instance SLPI commence une temporisation d'application pour surveiller la fin prévue du filtrage de service;
- 2) le modèle SCME est dans l'état "Waiting For ServiceFilteringResponse".

L'envoi de l'opération "ActivateServiceFiltering" provoque le passage de l'état "Service Filtering Idle" dans l'état "Waiting For SSF Service Filtering Response". Le modèle SCME reste dans cet état jusqu'à la fin de la temporisation dans l'instance SLPI. Il est prévenu par l'instance SLPI de la fin de la temporisation. Puis, il passe à l'état "Service Filtering Idle".

Si aucune erreur ne s'est produite après la réception d'une opération "ActivateServiceFiltering" dans l'entité SSF, un résultat vide est envoyé en retour vers l'entité SCF. Cela n'entraîne pas de changement d'état dans le modèle SCME.

Pour changer les paramètres d'une entité existante de filtrage de service, l'entité SCF doit envoyer une opération "ActivateServiceFiltering" avec les mêmes "filteringCriteria". Le second paramètre remplace le premier.

3.3.1.2.2 Traitement des erreurs

Le traitement générique d'erreurs pour les erreurs relatives aux opérations est décrit en 3.2 et les services de l'application TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.1.3 Entité de réponse (SSF)

3.3.1.3.1 Procédure normale

Précondition de l'entité SSF:

- aucune.

Postcondition de l'entité SSF:

- l'entité SSME-FSM est dans l'état "Traitement non associé à un appel".

S'il n'y a pas d'entité SSME-FSM déjà existante pour les critères de filtrage "filteringCriteria" donnés, une nouvelle machine SSME-FSM est créée. Cette machine SSME-FSM passe à l'état "Traitement non associé à un appel" et déclenche le filtrage de service pour les appels du RI spécifiés. Les paramètres "filteredCallTreatment", "filteringCharacteristics", "filteringCriteria", "filteringTimeOut" et "startTime" sont initialisés conformément aux valeurs fournies dans l'opération. Un certain nombre de compteurs sera attribué et remis à zéro. Au cas où le paramètre "startTime" ne serait pas atteint, le filtrage de service devra intervenir au moment prévu.

Si l'opération "ActivateServiceFiltering" fait appel à une entité de filtrage déjà existante, les paramètres "filteredCallTreatment", "filteringCharacteristics", "filteringTimeout" et "startTime" sont modifiés comme le prévoit l'opération. Dans le cas où l'entité de filtrage de service appelée est active, l'entité SSF indique les valeurs de comptage à l'entité SCF par l'intermédiaire de l'opération "ServiceFilteringResponse". Le processus de filtrage de service est arrêté si une opération "stopTime" ou "durée" égale à ZÉRO ou un nouveau paramètre "startTime" qui n'a pas encore été atteint est donné. Ensuite, l'entité SSF procède comme cela est décrit dans "ServiceFilteringResponse". Au cas où le paramètre "startTime" ne serait pas encore atteint, le filtrage de service serait poursuivi au moment prévu.

Si le filtrage de service se poursuit, la machine SSME-FSM reste dans l'état "Traitement de non-appel". Autrement, la machine se positionne dans l'état "Idle Management".

Lorsqu'un appel satisfait plusieurs critères "filteringCriteria", il doit être soumis au filtrage sur les critères les plus spécifiques, comme les critères ayant la valeur "callingAddressValue" la plus longue ou le numéro "locationNumber" le plus long, ou bien les critères ayant le plus grand nombre de paramètres spécifiés.

Lors de l'exécution du filtrage de service avec les paramètres "filteringCriteria"- "addressAndService", les premiers paramètres devant être vérifiés seront toujours "serviceKey" et "calledAddressValue".

Si une opération "ActivateServiceFiltering" est transmise à l'entité SSF avec les "filteringCriteria", "addressAndService", et la présence de deux paramètres "callingAddressValue" et "locationNumber", le texte suivant est applicable:

- si l'entité SSF reçoit un appel satisfaisant aux critères "serviceKey" et "calledAddressValue" (dans les critères activés de filtrage), elle recherche si la zone "locationNumber" est présente ou non dans le message initial d'adresse. Si ce nombre est présent et s'il correspond aux "filteringCriteria" activés, l'appel est filtré. Si l'entité SSF découvre que la zone "locationNumber" est absente, elle vérifiera la valeur "callingAddressValue" et exécutera le filtrage en fonction de ce paramètre.

Si aucune erreur ne s'est produite après la réception d'un "ActivateServiceFiltering" sur l'entité SSF, un résultat vide est envoyé en retour vers l'entité SCF. Cela n'entraîne pas de changement d'état dans la machine SSME-FSM.

Les temporisateurs suivants d'application sont utilisés:

- détecter le moment où le filtrage de service commencera (heure de départ);
- durée du filtrage de service;
- intervalle de temps pour le filtrage de service (pour la méthode de commande du temporisateur).

3.3.1.3.2 Traitement des erreurs

Si l'entité SSF détecte une erreur avec l'une quelconque des valeurs d'erreur définies, cette erreur est signalée à l'entité SCF.

L'événement est enregistré dans l'entité SSF et un cas d'erreur est signalé.

Au cas où une nouvelle machine SSME-FSM devrait être créée, il serait mis fin à la relation et toutes les ressources concernées (par exemple compteurs) seraient libérées. La machine SSME-FSM reste dans l'état "Idle Management".

Au cas où il y a déjà une machine SSME-FSM existante, les données de filtrage de service restent inchangées. L'entité SSME-FSM reste dans l'état "Traitement non associé à un appel".

Le traitement générique d'erreurs pour les erreurs relatives aux opérations est décrit au 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.2 Procédure ActivityTest (essai d'activité)

3.3.2.1 Description générale

Cette opération est utilisée pour vérifier l'existence permanente d'une relation entre les entités SCF et SSF. Si la relation est encore existante, l'entité SSF doit répondre. Au cas où aucune réponse n'est reçue dans une période de temps donnée, l'entité SCF supposera que l'entité SSF a échoué d'une manière ou d'une autre et effectuera les actions nécessaires.

3.3.2.1.1 Paramètres

Aucun.

3.3.2.2 Entité d'appel (SCF)

3.3.2.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation existe déjà entre les entités SCF et SSF;
- 2) lorsque la temporisation de contrôle d'activité (Tati) se termine, l'opération "Contrôle d'activité" est envoyée à l'entité SSF.

Postcondition de l'entité SCF:

- si un résultat "ActivityTest" est reçu en retour, l'entité SCME remet à zéro le temporisateur de contrôle d'activité et ne prend pas d'autre mesure.

3.3.2.2.2 Traitement des erreurs

Si une temporisation sur l'opération "ActivityTest" ou si une interruption P est reçue en provenance du sous-système TCAP, cela indique que la relation avec l'entité SSF a été en quelque sorte perdue. Si une fin de temporisation est reçue, l'entité SCF interrompt le dialogue.

L'instance SLPI qui était l'utilisatrice de ce dialogue sera prévenue et l'automate SCSM-FSM correspondant ira à l'état "Idle".

3.3.2.3 Entité de réponse (SSF)

3.3.2.3.1 Procédure normale

Précondition de l'entité SSF:

- une relation existe entre les entités SCF et SSF.

Postconditions de l'entité SSF:

- 1) la machine SSME-FSM reste ou se positionne dans l'état "Traitement non associé à un appel";
- 2) si l'identificateur de dialogue est actif et s'il y a un automate SSF-FSM utilisant le dialogue, l'entité SSME enverra un résultat en retour "ActivityTest" à l'entité SCF. S'il n'y a pas d'autres activités de gestion, l'entité SSME-FSM retournera vers l'état "Idle Management";
si l'identificateur de dialogue n'est pas actif, le sous-système TCAP dans l'entité SSF présentera une interruption P et le modèle SSME ne recevra jamais dans ce cas d'identificateur de demande d'essai "ActivityTest" et ainsi ne sera pas en mesure de répondre.

3.3.2.3.2 Traitement des erreurs

Le traitement des erreurs relatives aux opérations n'est pas applicable, en raison de l'opération de classe 3.

3.3.3 Procédure AddEntry (Ajouter une entrée)

3.3.3.1 Description générale

L'opération "AddEntry" contenue dans la Recommandation X.500 est utilisée pour demander à l'entité SDF d'ajouter une entrée feuille (une entrée objet ou une entrée alias) dans la structure arborescente (DIT). Pour avoir une description complète de l'opération AddEntry, il convient de se référer au 11.1/X.511.

3.3.3.1.1 Paramètres

Voir 11.1.1/X.511 et 11.1.2/X.511.

3.3.3.2 Entité d'appel (SCF)

3.3.3.2.1 Procédure normale

Précondition de l'entité SCF:

- SCSM: "Rattachement SDF" ou "Attente des demandes subséquentes".

Postcondition de l'entité SCF:

- SCSM: "Rattachement SDF" ou "Attente des demandes subséquentes".

Quand l'entité SCSM est dans l'état "Attente des demandes subséquentes" et qu'il existe un besoin de logique de traitement de service d'ajouter une entrée à l'entité SDF, un événement interne [(e2) Request_to_SDF] se produit. L'entité SCSM reste dans l'état "Attente des demandes subséquentes" et l'opération n'est pas envoyée avant que la procédure d'application n'ait indiqué, par un délimiteur (ou une fin de la temporisation), que l'opération devrait être envoyée. L'opération est envoyée vers l'entité SDF dans un message contenant un argument Bind (Rattacher). L'entité SCSM attend la réponse provenant de l'entité SDF. La réception de la réponse [(E5) Response_from_SDF_with_Bind ou (E4) Bind_Error] à l'opération de rattachement préalablement présentée à l'entité SDF provoque le passage de l'entité SCF de l'état "Rattachement SDF" à l'état "Idle". Lorsque l'entité SCSM est passée à l'état "Idle", l'opération AddEntry n'a pas été prise en compte. Dans l'état "Rattachement SDF", la réponse de l'opération AddEntry [(E7) Response_from_SDF] provoque un changement d'état de l'entité SCF suivi d'un retour en même état ("Rattachement SDF"). Cela peut être soit le résultat de l'opération AddEntry, soit une erreur.

Quand l'entité SCSM est dans l'état "Rattachement SDF" et que la logique de traitement de service a besoin d'ajouter une entrée à l'entité SDF, un événement interne se produit. Cet événement appelé (e6) Request_to_SDF provoque un changement d'état suivi d'un retour en même état "Rattachement SDF" et l'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E7) Response_from_SDF] à la requête AddEntry qui a été précédemment présentée à l'entité SDF provoque un changement d'état de l'entité SCF suivi d'un retour au même état "Rattachement SDF". La réponse provenant de l'entité SDF peut être soit le résultat de l'opération AddEntry, soit une erreur.

3.3.3.2.2 Traitement des erreurs

Le traitement générique des erreurs pour les erreurs relatives à l'opération est décrit dans 11.1.4/X.511 et 11.1.5/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 2.2.2.

3.3.3.3 Entité de réponse (SDF)

3.3.3.3.1 Procédure normale

Précondition SDF:

- modèle SDSM: "Rattachement SDF" ou "Rattachement en cours".

Postcondition SDF:

- modèle SDSM: " Rattachement SDF" ou "Rattachement en cours".

L'entité SDF est dans l'état "Rattachement en cours", quand l'événement externe (E3) Request_from_SCF provoqué par la réception d'une opération "AddEntry" en provenance de l'entité SCF survient. L'entité SDF ne procède pas à l'opération avant qu'une opération "Rattachement" ait été exécutée avec succès. Elle reste dans le même état.

L'entité SDF est dans l'état "Rattachement SDF", quand l'événement externe (E7) Request_from_SCF provoqué par la réception d'une opération "AddEntry" provenant de l'entité SCF se produit. L'entité SDF attend la réponse apportée à l'opération.

A la réception de l'événement (E7) et avant d'ajouter l'article de la nouvelle entrée, les actions suivantes sont effectuées par l'entité SDF:

- vérifier que l'objet supérieur auquel il convient d'ajouter l'entrée existe dans l'entité SDF;
- vérifier que l'entrée n'existe pas encore dans l'entité SDF;
- vérifier que les droits d'accès pour ajouter une entrée et chacun de ses composants (attributs et valeurs) sont suffisants;
- vérifier que l'entrée est conforme au schéma d'utilisation de l'Annuaire.

Une fois que les actions spécifiées ci-dessus se sont correctement déroulées, l'entrée est ajoutée dans la base de données SDF. Un résultat zéro est renvoyé vers l'entité SCF. L'envoi du résultat correspond à l'événement (e6) Response_to_SCF.

3.3.3.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 11.1.4/X.511 et 11.1.5/X.511, et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 2.2.2.

3.3.4 AnalysedInformation Procedure (Procédure après l'information analysée)

3.3.4.1 Description générale

Cette opération est envoyée par l'entité SSF à l'entité SCF, après qu'elle eut détecté un état de déclenchement valide au point de détection d'information analysée, ou afin de signaler un événement demandé par RequestReportBCSMEvent.

3.3.4.1.1 Paramètres

- dPSpecificCommonParameters (Eléments communs relevant spécifiquement du point de détection):
- dialledDigits (Chiffres composés):
voir la Recommandation Q.1290.
- callingPartyBusinessGroupID (Identification du groupe d'abonnés du demandeur):
voir la Recommandation Q.1290. L'entité SCF peut utiliser cet élément d'information afin de choisir des programmes de logique de traitement de service pour les groupes et affectés aux besoins d'autorisation. Les exploitants de réseau peuvent spécifier que cet élément d'information devrait être utilisé pour le cas où leur réseau particulier disposerait des informations nécessaires.
- callingPartySubaddress (Sous-adresse du demandeur):
voir la Recommandation Q.931.
- callingFacilityGroup (Faisceau d'équipement du demandeur):
voir la Recommandation Q.1290.
- callingFacilityGroupMember (Elément du faisceau d'équipement du demandeur):
voir la Recommandation Q.1290.
- originalCalledPartyID:
voir l'information de signalisation de numéro demandé initial de la Recommandation Q.762.
- préfixe:
voir la Recommandation Q.1290.
- redirectingPartyID (Identification du réachemineur):
ce paramètre est (le cas échéant) le numéro d'annuaire du dernier réachemineur de l'appel.
- redirectionInformation (Informations de réacheminement):
voir la Recommandation Q.763.
- routeList (Liste d'acheminement):
la liste d'acheminement est la liste des itinéraires qui seront généralement utilisés pour acheminer les appels. Les exploitants de réseau peuvent spécifier si ces éléments d'information doivent être utilisés pour autant que leur réseau particulier dispose de l'information nécessaire.
- travellingClassMark (Marque de classe de transport):
voir la Recommandation Q.1290.
- featureCode (Chiffres composés):
voir la Recommandation Q.1290.
- accessCode (Code d'accès):
voir la Recommandation Q.1290.
- exploitant:
voir la Recommandation Q.1290.

3.3.4.2 Entité d'appel (SSF)

3.3.4.2.1 Procédure normale

Préconditions de la fonction SSF (TDP):

- 1) une tentative d'appel a lieu au départ;
- 2) le numéro du demandé est disponible et le type d'adresse déterminé;
- 3) aucun espacement d'appels ou filtrage de service n'est activé;
- 4) les critères de point de détection ont été satisfaits;
- 5) pour un point de déclenchement de type demande, il n'existe pas de relation de commande qui ait une influence sur le segment d'appel.

Préconditions de la fonction SSF (EDP):

- 1) pour un point de détection d'événement de type demande (EDP-R), il n'y a pas de relation de commande existante et le point EDP AnalysedInformation est armé;
- 2) pour un point de détection d'événement de type notification (EDP-N), il y a une surveillance ou une relation de commande.

Postconditions de la fonction SSF (TDP):

- 1) pour un point de détection de déclenchement de type demande (TDP-R), un traitement d'appel de base a été interrompu au point de détection Analysed_Info et une relation de commande a été établie;
- 2) pour un point de détection de déclenchement de type notification (TDP-N), un traitement d'appel de base s'effectue au point de traitement d'appel (PIC) Select_Route et aucune relation de commande n'est établie.

Postconditions de la fonction SSF (EDP):

- 3) pour un EDP-R, un traitement d'appel de base a été interrompu au point de détection Analysed_Info, et la relation de commande existante reste établie;
- 4) pour un EDP-N, le traitement d'appel de base s'effectue au point de traitement d'appel (PIC) Select_Route et la relation de non-commande existante reste établie, à moins qu'aucun autre EDP ne soit armé et qu'aucune des opérations "CallInformationReport" ou "ApplyChargingReport" ne soit demandée.

L'entité SSF a suffisamment d'informations disponibles associées au segment d'appel initial. Ces informations ont été analysées et les résultats sont décrits ci-dessous.

- 1) A partir d'une ligne classique (non-RNIS) ou d'une interface RNIS, les résultats d'analyse se composent d'un ou de plusieurs des éléments suivants:
 - CalledPartyID (Identification du demandé) – Le numéro utilisé pour identifier le demandé est le sens en avant (c'est-à-dire que ce numéro est utilisé pour renseigner l'élément d'information du numéro du demandé contenu dans le protocole de signalisation du support).
 - TypeOfCall (Type d'appel) – Indique l'un des fournisseurs de liaisons entre autocommutateurs, un exploitant de liaisons internationales, un exploitant d'autocommutateurs locaux, un exploitant de liaisons entre autocommutateurs sur des lignes à grande distance ou un exploitant de liaisons internationales.
 - Exploitant sur des lignes à grande distance (pour les appels nécessitant un exploitant de liaisons à grande distance entre autocommutateurs) – Indique le type d'exploitant demandé.
 - CarrierIdentificationCode (Code d'identification de l'exploitant de réseaux) (pour les appels nécessitant un exploitant de liaisons entre autocommutateurs) – Indique le code de l'exploitant spécifique devant être employé.
 - CarrierSelection (Sélection de l'exploitant) (pour les appels qui demandent un exploitant de liaisons entre autocommutateurs) – Indique si le demandeur a composé le code de l'exploitant choisi et si le demandeur est déjà abonné à l'exploitant choisi.
 - RouteIndex (Index d'acheminement) – Index sur une liste d'acheminements (si l'appel n'aboutit pas à cette entité SSF).
 - Recueil d'informations – Codes d'accès et préfixes, chiffres et informations d'adresses recueillies.

- 2) A partir d'une jonction exploitée par une signalisation classique ou sémaphore, ces informations comprennent un ou plusieurs des éléments suivants:
- ChargeNumber (Numéro à taxer).
 - CalledPartyID (Identification du demandé).
 - TypeOfCall (Type d'appel).
 - CarrierIdentificationCode (pour les appels nécessitant un exploitant de liaisons entre autocommutateurs).
 - Exploitant (pour les appels nécessitant un exploitant de liaisons entre autocommutateurs).
 - CarrierSelection (Choix de l'exploitant) (pour les appels nécessitant un exploitant de liaisons entre autocommutateurs).
 - RouteIndex (Index d'acheminement).
 - Informations recueillies – Les informations d'adresse reçues, préfixes, etc. Les informations recueillies pour un appel à partir d'une jonction sur réseau sémaphore proviennent des informations fournies dans le message d'adresse initial (IAM).

3.3.4.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.4.3 Entité de réponse (SCF)

3.3.4.3.1 Procédure normale

Préconditions de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R de l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCP (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction de l'entité SSF est en préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute et met fin à l'opération.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et s'il y a encore des points EDP armés, aucune des opérations de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou

l'entité SCME-FSM passe à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune des opérations de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou

l'automate SCSM-FSM passe à l'état "Préparation des instructions SSF" si le type de message était la demande.

3.3.4.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.5 AnalyseInformation procedure

3.3.5.1 Description générale

Cette opération demande à l'entité SSF d'effectuer les principales actions initiales de traitement d'appel de base afin d'analyser l'information de destination qui est soit numérotée par un demandeur, soit fournie par l'entité SCF. Cela comprend les actions destinées à valider les informations de destination selon un plan de numérotage spécifié et, si ces informations sont valides, la détermination des informations destinées à établir l'appel.

3.3.5.1.1 Paramètres

- destinationRoutingAddress:
représente une liste de numéros de demandés (primaires et secondaires).
- alertingPattern:
voir la Recommandation Q.1290. Elle ne s'applique que si la signalisation de réseau gère ce paramètre ou si l'entité SSF est le centre local de rattachement de l'abonné.
- iSDNAccessRelatedInformation:
transporte les mêmes informations que le paramètre de transport d'accès du sous-système utilisateur ISUP dans la Recommandation Q.762.
- originalCalledPartyID:
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.
- callingPartyNumber (Numéro du demandeur):
voir la Recommandation Q.762.
- callingPartysCategory (Catégorie de demandeur):
voir la Recommandation Q.762.
- calledPartyNumber (Numéro du demandé):
voir la Recommandation Q.762. Le paramètre calledPartyNumber ou destinationRoutingAddress sera fourni par l'entité SCF dans l'opération AnalyseInformation.
- chargeNumber (Numéro à taxer):
voir la Recommandation Q.1290.
- travellingClassMark (Marque de classe de transport):
voir la Recommandation Q.1290.
- carrier (Exploitant):
voir la Recommandation Q.1290. Dans ce message, le champ de sélection de l'exploitant est égal à zéro (00000000) et l'identification de l'exploitant indique quel exploitant doit être employé pour l'appel.

3.3.5.2 Entité d'appel (SCF)

3.3.5.2.1 Procédure normale

Précondition de l'entité SCF:

- 1) une tentative d'appel au départ a eu lieu;
- 2) l'autorisation ou la possibilité de faire un appel sortant a été vérifiée;
- 3) les informations de destination sont disponibles dans l'entité SSF ou fournies par l'entité SCF;
- 4) Le traitement d'appel de base a été interrompu sur l'un des points de détection suivants:
 - Origination_Attempt_Authorized (Tentative autorisée au départ).
 - Collected_Info (Information recueillie).
 - Analysed_Info (Information analysée).
 - Route_Select_Failure (Echec de sélection d'itinéraire).
 - 0_Called_Party_Busy (Occupation du numéro demandé au départ).
 - 0_No_Answer (Non-réponse au départ).
 - 0_Disconnect (Déconnexion au départ) (Déconnecter le demandé seulement).

- 5) Une relation de commande a été établie et l'instance SLPI traite la demande entrante.

Postconditions de l'entité SCF:

- 1) il est mis fin à l'exécution de l'instance SLPI si aucune surveillance n'est demandée;
- 2) il est mis fin à l'exécution de l'instance SLPI pendant que l'événement contrôlé se produit, si une surveillance est demandée.

Le message AnalyseInformation demande à l'entité SSF/CCF de reprendre le traitement d'origine de l'appel en tenant compte de l'adresse, de l'acheminement, et des informations de facturation fournis dans les paramètres du message. Le traitement reprend au point de traitement d'appel ANALYSE_INFORMATION.

3.3.5.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.5.3 Entité de réponse (SSF)

3.3.5.3.1 Procédure normale

Préconditions de l'entité SSF:

- une demande TDP ou EDP a été invoquée.

Postconditions de l'entité SSF:

- l'entité SSF effectue les actions de traitement d'appel afin d'analyser les informations de destination.

Les configurations suivantes de paramètre de message sont préconisées et elles entraîneront les actions de traitement d'appel SSF/CCF qui sont spécifiées ci-après:

- 1) Pas d'exploitant spécifié:
 - le choix de l'exploitant sera effectué par l'autocommutateur;
 - l'adresse du demandé fournie dans le paramètre CalledPartyNumber sera soumise à l'analyse de déclenchement. Il convient de remarquer que si la même adresse du demandé est renvoyée, le même déclenchement peut être détecté et être de nouveau traité au point TDP de l'information analysée.
- 2) Exploitant spécifié:
 - l'entité SSF/CCF reprendra le traitement de l'appel au point de traitement d'appel ANALYSE_INFORMATION, en utilisant l'exploitant désigné (spécifique à l'exploitant de réseau).

Les spécifications complémentaires suivantes s'appliquent également au message AnalyseInformation:

- l'entité SSF/CCF acceptera les informations d'adresse du demandé contenues dans le paramètre CalledPartyNumber qui est conforme au plan de numérotage du réseau public;
- si un itinéraire nécessite un paramètre TCM, la valeur contenue dans le paramètre TCM devra être utilisée. Si un paramètre TCM est exigé et si ce paramètre n'est pas fourni dans l'opération AnalyseInformation, l'entité SSF/CCF déduira le TCM selon les spécifications qui s'appliquent, en supposant qu'il n'y ait pas d'implication du réseau intelligent dans l'appel (Option spécifique à l'exploitant de réseau);
- si l'entité SCF ne désigne pas un exploitant et si la sélection de l'exploitant est choisie par l'entité SSF/CCF, cet indicateur sera placé en position normale comme s'il n'y avait pas d'implication du réseau intelligent dans l'appel (Option spécifique à l'exploitant de réseau).

AnalyseInformation (Analyse d'information) avec CallingPartysCategory (Catégorie de demandeur) et ChargeNumber (Numéro à taxer):

- si le paramètre CallingPartysCategory (Catégorie du demandeur) est inclus, cette valeur devra être utilisée comme chiffres du signal multifréquence à deux fréquences (DTMF) (*dual tone multifrequency*) ou comme paramètre de catégorie d'appelant du sous-système utilisateur ISUP dans toute signalisation subséquente, y compris la signalisation vers un système exploité par l'intermédiaire de la signalisation des services de l'exploitant (spécifique à l'exploitant de réseau);
- quand des chiffres DTMF sont nécessaires pour une signalisation subséquente, l'équivalent décimal de la valeur CallingPartysCategory sera utilisé pour une numérotation (par exemple "01000000" équivaut à 11 chiffres de "64") (ce qui est spécifique à l'exploitant de réseau);

- si le paramètre CallingPartysCategory (Catégorie d'appelant) n'est pas inclus, une indication appropriée d'incident sera utilisée dans toute signalisation DTMF ou du sous-système utilisateur ISUP subséquente (spécifique à l'exploitant de réseau);
- si le paramètre ChargeNumber (Numéro à taxer) est inclus, cette valeur sera utilisée comme le numéro du demandeur sur des jonctions non sémaphores ou bien comme paramètre Numéro à taxer du sous-système utilisateur ISUP spécifique au réseau dans toute signalisation subséquente, y compris la signalisation vers un système exploité par l'intermédiaire de la signalisation des services de l'exploitant (spécifique à l'exploitant de réseau);
- si le paramètre ChargeNumber n'est pas inclus, la valeur "chiffres 0" sera utilisée pour le ChargeNumber (Numéro à taxer) sur toute jonction non sémaphore subséquente ou toute signalisation du sous-système utilisateur ISUP (spécifique à l'exploitant de réseau).

AnalyseInformation et O_Called_Party_Busy, O_No_Answer:

- quand l'entité SSF/CCF reçoit le message AnalyseInformation en réponse au message O_Called_Party_Busy, l'entité SSF/CCF effectue les actions suivantes:
 - 1) l'entité SSF/CCF doit libérer toutes les ressources qui étaient utilisées pour traiter l'appel entre ANALYSE_INFORMATION et ROUTING & les points de traitement d'appel ALERTING;
 - 2) l'entité SSF/CCF reprendra le traitement de l'appel au point de traitement d'appel ANALYSE_INFORMATION et traitera le message comme cela est décrit dans cette description de procédure;
 - 3) si l'accès d'origine est DSS 1, l'entité SSF/CCF n'enverra pas d'autre message CALLPROceeding. (L'interfonctionnement DSS 1 du protocole INAP fait l'objet d'un complément d'étude.)
- Quand le message AnalyseInformation est reçu en réponse à un message O_No_Answer pour le point TDP-R, l'entité SSF/CCF doit effectuer les actions suivantes:
 - 1) l'entité SSF/CCF, au cas où elle fournirait une tonalité de retour d'appel audible par le demandeur, retirera cette tonalité;
 - 2) l'entité SSF/CCF libèrera toutes les ressources qui ont été utilisées pour traiter l'appel entre ANALYSE_INFORMATION et les points de traitement d'appel ROUTING & _ALERTING;
 - 3) l'entité SSF/CCF reprendra le traitement d'appel au point de traitement d'appel ANALYSE_INFORMATION, et traitera le message comme cela est décrit dans cette description de procédure;
 - 4) Si l'accès d'origine est DSS 1, l'entité SSF/CCF ne devra pas envoyer d'autre message CALLPROceeding. (L'interfonctionnement des protocoles INAP et DSS 1 fait l'objet d'un complément d'étude.)
- Quand le message AnalyseInformation est reçu en réponse à un message O_No_Answer pour le point EDP-R, l'entité SSF-CCF doit effectuer les actions suivantes:
 - 1) l'entité SSF/CCF, au cas où elle fournirait une tonalité de retour d'appel audible par le demandeur, retirera cette tonalité;
 - 2) l'entité SSF/CCF libèrera toute les ressources qui ont été utilisées afin de traiter l'appel entre ANALYSE_INFORMATION et les points de traitement d'appel ROUTING & ALERTING;
 - 3) l'entité SSF/CCF reprendra le traitement de l'appel au point de traitement ANALYSE_INFORMATION et traitera le message comme cela est décrit dans la présente Recommandation;
 - 4) si l'accès d'origine est DSS 1, l'entité SSF/CCF n'enverra pas un autre message CALL PROCEEDing. (L'interfonctionnement DSS 1 du protocole INAP fait l'objet d'un complément d'étude.)

AnalyseInformation avec OriginalCalledPartyID (Identification du demandé initial):

- quand l'entité SSF/CCF reçoit un message AnalyseInformation contenant le paramètre OriginalCalledPartyID, l'entité SSF/CCF devra établir des correspondances entre le paramètre OriginalCalledParty ID et par exemple (interfonctionnement ISUP faisant l'objet d'une étude complémentaire):
 - 1) l'élément d'information OriginalCalledNumber est l'élément d'information de service complémentaire si l'entité SSF/CCF n'a pas encore de valeur pour OriginalCalledNumber ET si l'entité SSF/CCF achemine l'appel vers une ligne du réseau RNIS (signalisation RNIS pour l'accès de base et l'accès primaire);

- 2) le paramètre OriginalCalledNumber dans le message d'adresse initial (IAM) si l'entité SSF/CCF n'a pas encore de valeur pour OriginalCalledNumber ET si l'entité SSF/CCF achemine l'appel vers une jonction sémaphore (pour la signalisation RNIS-UP);

Autres spécifications AnalyseInformation:

- si l'entité SSF/CCF a associé un identificateur de groupe d'abonnés à l'appel et si cet appel est acheminé sur une jonction sur réseau sémaphore, l'entité SSF/CCF inclura ces informations dans le paramètre Groupe d'abonnés spécifique du réseau (Message d'adresse initial spécifique au réseau ISUP) (spécifique à l'exploitant de réseau);
- si le paramètre CalledPartyNumber ou CallingPartyNumber est inclus dans l'opération AnalyseInformation, cette valeur devra être utilisée de façon subséquente pour l'appel et la valeur existante remplacée.

3.3.5.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération sont décrites au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs dans les opérations sont décrits au 3.4.

3.3.6 ApplyCharging procedure (Appliquer la procédure de taxation)

3.3.6.1 Description générale

Cette opération est utilisée pour assurer l'interaction à partir de l'entité SCF avec les mécanismes de taxation de l'entité SSF. L'opération "ApplyChargingReport" assure un retour d'information provenant de l'entité SSF vers l'entité SCF.

Etant donné que plusieurs configurations de connexion peuvent être établies pendant un appel, il existe une possibilité d'invoquer "ApplyCharging" au début de chaque configuration de connexion, et ce pour chaque participant.

Les scénarios de taxation 4.1 et 4.2 sont à la base de cette opération (il convient de se référer à l'Appendice II/Q.1214 "Scénarios de taxation").

3.3.6.1.1 Paramètres

- aChBillingChargingCharacteristics (Caractéristiques d'application de la taxation):
ce paramètre spécifie les informations relatives à la taxation qui doivent être fournies par l'entité SSF, de même que les conditions dans lesquelles ces informations doivent être retournées à l'entité SCF par l'intermédiaire de l'opération "ApplyChargingReport". Leur contenu est spécifique à l'exploitant de réseau.
- partyToCharge (correspondant à taxer):
ce paramètre indique le participant auquel devrait s'appliquer l'opération "ApplyCharging". Si ce participant n'est pas présent, l'opération est appliquée au correspondant A.

3.3.6.2 Entité d'appel (SCF)

3.3.6.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) l'instance SLPI a déterminé qu'une opération "ApplyCharging" doit être envoyée.

Préconditions de l'entité SCF:

- 1) aucun changement d'état FSM;
- 2) l'instance SLPI attend des opérations "ApplyChargingReport" en provenance de l'entité SSF.

L'automate SCSM-FSM est dans l'état "Préparer les instructions SSF" ou dans l'état "Mise en files d'attente FSM". Cette opération est invoquée par l'entité SCF si une instance SLPI aboutit à la demande d'interaction avec les mécanismes de taxation à l'intérieur de l'entité SSF pour obtenir des informations relatives à la taxation.

3.3.6.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.6.3 Entité de réponse (SSF)

3.3.6.3.1 Procédure normale

Préconditions de l'entité SSF:

- Le modèle SSF-FSM se trouve dans l'un des états suivants:
 - "Attente des instructions" (état c);
 - "Attente de la fin de l'interaction avec l'utilisateur" (état d);
 - "Attente de la fin de la connexion temporaire" (état e);
 - "Surveillance" (état f);ou l'automate SSF-FSM assistant/de transfert de tâche est dans l'état:
 - "Attente des instructions" (état b).

Postconditions de l'entité SSF:

- aucun changement d'état dans la machine FSM.

A la réception de cette opération, l'entité SSF définit les données de chargement en utilisant les éléments d'information inclus dans l'opération et agit en conséquence. En outre, l'entité SSF commencera la surveillance de la fin de la configuration de connexion et, le cas échéant, aux autres événements de taxation.

3.3.6.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit au 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.7 Procédure ApplyChargingReport

3.3.7.1 Description générale

Cette opération est utilisée par l'entité SSF pour rendre compte des informations de taxation vers l'entité SCF comme l'entité SCF l'a demandé, en utilisant l'opération "ApplyCharging".

Pendant une configuration de la connexion, l'opération "ApplyChargingReport" peut être invoquée à de multiples reprises. Pour chaque configuration de correspondant et pour chaque configuration de connexion, l'opération "ApplyChargingReport" peut être utilisée plusieurs fois. Il convient de noter qu'au moins une opération "ApplyChargingReport" doit être envoyée à la fin du processus de taxation selon la configuration de la connexion.

Les scénarios de taxation 4.1 et 4.2 sont à la base de cette opération (il convient de se référer à l'Appendice II/Q. 1214 "Scénarios de taxation").

3.3.7.1.1 Paramètres

- CallResult:
 - ce paramètre fournit à l'entité SCF des informations relatives à la taxation qui ont été préalablement demandées en utilisant l'opération "ApplyCharging". Le résultat "CallResult" inclura le paramètre "partyToCharge" tel qu'il a été reçu dans l'opération "ApplyCharging" afin d'établir une corrélation entre le résultat et la demande. Les autres informations de "CallResult" sont spécifiques à l'exploitant de réseau. Parmi ces autres informations, il peut y avoir les valeurs de compteur de volume, les coûts, les changements de tarif et l'heure du changement, les horodateurs, les durées, etc.

3.3.7.2 Entité d'appel (SSF)

3.3.7.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) un événement de taxation qui était demandé par l'entité SCF par l'intermédiaire de l'opération "ApplyCharging" a été détecté.

Postconditions de l'entité SSF:

- si la configuration de la connexion ne change pas, aucun changement d'état FSM ne devra intervenir. Si la configuration de la connexion change, l'entité FSM se mettra en position:
 - état "Idle" s'il n'y a pas d'autre point EDP armé et s'il n'y a pas de demandes de compte rendus en cours; ou
 - autrement restera dans le même état.

Cette opération est invoquée si un événement de taxation qui était demandé par l'entité SCF a été détecté. L'opération "ApplyChargingReport" ne traite que des événements de taxation à l'intérieur de l'entité SSF elle-même. Des exemples d'événements de taxation peuvent être: la valeur de seuil obtenue, la fin de la temporisation, le changement de tarif, la fin de la configuration de connexion, etc.

3.3.7.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.7.3 Entité de réponse (SCF)

3.3.7.3.1 Procédure normale

Préconditions de l'entité SCF:

- une opération "ApplyCharging" a été envoyée à la demande d'une instance SLPI et l'instance SLPI attend un compte rendu "ApplyChargingReport" en provenance de l'entité SSF.

Postconditions de l'entité SCF:

- aucun changement d'état FSM n'intervient, si d'autres comptes rendus, y compris "Event ReportBCSM" et "CallInformationReport" sont prévus, ou il se produit une transition vers l'état "Repos" si le compte rendu est le dernier de l'opération, à condition qu'aucune opération "EventReportBCSM" ou "CallInformationReport" ne soit attendue.

A la réception de cette opération, l'instance SLPI qui attend cette opération continuera.

3.3.7.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit au 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.8 AssistRequestInstructions procedure (Procédure d'instructions de demande d'assistance)

3.3.8.1 Description générale

Cette opération est envoyée à l'entité SCF par une entité SSF, qui agit comme l'entité SSF assistante dans une procédure d'assistance ou de transfert de tâche, ou par une entité SRF. Elle est envoyée quand l'entité SSF ou SRF assistante reçoit une indication d'une entité SSF déclencheuse contenant des informations indiquant une procédure d'assistance ou de transfert.

3.3.8.1.1 Paramètres

- correlationID:
ce paramètre est utilisé par l'entité SCF afin d'associer les "AssistRequestInstructions" à partir de l'entité SSF assistante ou par une entité SRF avec la demande émanant de l'entité SSF déclencheuse. La valeur de "correlationID" peut être obtenue à partir de tous les chiffres dont on dispose, ou extraite des chiffres reçus par l'entité SSF déclencheuse.
- iPAavailable (Périphérique intelligent disponible):
voir la Recommandation Q.1290. Ce paramètre s'applique à l'opération seulement dans les scénarios physiques correspondants afin de fournir une assistance avec relais ou avec transfert de tâche. L'utilisation de ce paramètre dépend de l'exploitant de réseau.
- Capacités du périphérique intelligent intégré au point SSP ("service switching point" ou point de commutation de service):
voir la Recommandation Q.1290. Ce paramètre n'est applicable à cette opération que dans les scénarios physiques correspondant à une assistance avec relais ou avec transfert de tâche. L'utilisation de ce paramètre dépend de l'exploitant de réseau.

3.3.8.2 Entité d'appel (SSF/SRF)

3.3.8.2.1 Procédure normale

Précondition de l'entité SSF:

- une indication d'assistance est détectée par l'entité SSF assistante ou de transfert de tâche.

Postcondition de l'entité SSF:

- l'entité SSF assistante ou de transfert de tâche attend les instructions.

A la réception d'une indication d'assistance à partir de l'entité SSF déclencheuse, les entités SSF ou SRF devront s'assurer que les ressources nécessaires sont disponibles pour invoquer une opération "AssistRequestInstructions" dans SSF/SRF et indiquer à l'entité SSF déclencheuse que l'appel est accepté (voir la Recommandation Q.71). L'opération "AssistRequestInstructions" est invoquée par l'entité SSF ou par l'entité SRF une fois que l'appel qui a déclenché l'indication d'assistance est accepté. L'entité assistante SSF FSM passe à l'état "Attente des instructions".

3.3.8.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.8.3 Entité de réponse (SCF)

3.3.8.3.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre l'entité SCF et l'entité SSF déclencheuse en cas de procédure d'assistance;
- 2) l'entité SCF attend "AssistRequestInstructions".

Postcondition de l'entité SCF:

- une instruction SSF ou SRF est en voie de préparation.

A la réception de cette opération dans l'état "Attente des instructions de demande d'assistance" de l'entité SCSM, l'entité SCP doit effectuer les actions suivantes.

Si l'opération "AssistRequestInstructions" a été reçue à partir d'une entité SSF assistante, et si la ressource est disponible, l'entité SCSM prépare l'une des opérations "ConnectToResource" et "PlayAnnouncement" ou "PromptAndCollectUserInformation" devant être envoyée à l'entité SSF assistante.

L'entité SCF détermine SSF/SRF au moyen de "correlationID" (l'identificateur de corrélation), de "destinationNumber" (le numéro de destination) ou de sa connaissance des configurations du réseau.

Si l'opération "AssistRequestInstructions" a été reçue à partir d'une entité SRF, et si la ressource est disponible, l'entité SCSM prépare l'une des opérations "PlayAnnouncement" ou "PromptAndCollectUserInformation" devant être envoyée à l'entité SRF.

A la réception de cette opération provenant de l'entité SSF de transfert de tâche, l'entité SCSM associée à l'entité SSF de transfert de tâche passe de l'état "Idle" à l'état "Préparer les instructions SSF".

3.3.8.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.9 Bind Procedure (Procédure de rattachement)

3.3.9.1 Description générale

L'opération "Bind" de la Recommandation X.500 est utilisée par l'entité SDF afin de créer une association authentifiée entre une entité SCF et une entité SDF dans l'intérêt de l'utilisateur. Elle transporte les informations d'authentification de l'utilisateur, le cas échéant. Pour une description complète de l'opération "Bind", voir 8.1/X.511.

3.3.9.1.1 Paramètres

Voir 8.1.2/X.511 et 8.1.3/X.511.

3.3.9.2 Entité d'appel (SCF)

3.3.9.2.1 Procédure normale

Précondition de l'entité SCF:

- SCSM: "Idle".

Postconditions de l'entité SCF:

- 1) SCSM: "Rattachement SDF" en cas de succès;
- 2) SCSM: "Idle" en cas d'échec.

Quand l'entité SCSM est dans l'état "Idle" et quand il existe un besoin de logique de traitement de service d'interroger une entité SDF, un événement interne se produit. Cet événement, appelé (e1) Bind_Request provoque une transition vers l'état "Wait for Subsequent Request" (Attente des demandes subséquentes) et d'autres opérations sont attendues. Jusqu'à ce que le processus d'application n'ait indiqué par un délimiteur que l'opération de rattachement devait être envoyée, l'entité SCSM reste dans l'état "Wait for Subsequent Requests" (Attente des demandes subséquentes) et l'opération n'est pas envoyée. La réception du délimiteur provoque une transition vers l'état "Wait for Bind result" par la transition interne (e3) Request_to_SDF_with_Bind. L'opération est envoyée vers l'entité SDF. L'entité SCSM attend la réponse de l'entité SDF. La réception de la réponse [(E5) Response_from_SDF_with_Bind] à l'opération de rattachement précédemment présentée à l'entité SDF provoque une transition de l'entité SCF vers l'état "Rattachement SDF", si le résultat de l'opération de rattachement est positif. Autrement, la réception d'une erreur [(E4) Bind_Error] remet l'entité SCSM dans l'état "Idle".

3.3.9.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit au 8.1.4/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 2.2.2.

3.3.9.3 Entité de réponse (SDF)

3.3.9.3.1 Procédure normale

Préconditions SDF:

- SDSM: "Idle".

Postconditions SDF:

- 1) SDSM: "Rattachement SCF" (réussi);
- 2) SDSM: "Idle" (échec).

L'entité SDF est initialement dans l'état "Idle". Après avoir reçu l'événement externe (E1) Bind_from_SCF provoqué par la réception d'une opération "Rattachement" à partir de l'entité SCF, une transition vers l'état "Bind Pending" (Rattachement en cours) se produit. L'entité SDF exécute l'opération de rattachement en fonction du contenu de l'argument de rattachement. Une fois que l'entité SDF a exécuté l'opération de rattachement, le résultat de l'indication d'erreur est renvoyé vers l'entité SCF. L'entité SDF revient à l'état "Idle" s'il y a échec de l'opération de rattachement, ou vers l'état "Rattachement SCF" si le rattachement réussit. En cas de succès de la demande de rattachement, le résultat renvoyé peut comprendre les mots de passe de DirectoryBindResult. Ces mots de passe permettent à l'utilisateur d'établir l'identité de l'Annuaire. Ils permettent que les informations d'identification de l'agent DSA (qui fournit directement le service de l'Annuaire) soient acheminées vers l'agent DUA. Les mots de passe ont la même forme (c'est-à-dire nom, mot de passe) que ceux fournis par l'utilisateur.

3.3.9.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit au 8.1.4/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 2.2.2.

3.3.10 CallGap procedure (Procédure d'espacement des appels)

3.3.10.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de réduire le rythme auquel les demandes spécifiques de service sont envoyées à l'entité SCF.

3.3.10.1.1 Paramètres

a) `gapCriteria`:

ce paramètre identifie les critères pour qu'un appel soit soumis à l'espacement des appels.

i) `calledAddressValue`:

ce paramètre indique que l'espacement des appels sera appliqué lorsque les chiffres initiaux du numéro composé pour une tentative d'appel correspondront à ceux qui sont spécifiés dans le paramètre "`gapCriteria`".

ii) `gapOnService`:

ce paramètre indique que l'espacement des appels sera appliqué quand le critère "`serviceKey`" d'une tentative d'appel correspondra aux critères spécifiés dans "`gapCriteria`".

iii) `calledAddressAndService`:

ce paramètre indique que l'espacement des appels sera appliqué quand le critère "`serviceKey`" et les chiffres initiaux du numéro composé d'une tentative d'appel satisferont aux critères spécifiés dans "`gapCriteria`".

iv) `callingAddressAndService`:

ce paramètre indique que l'espacement des appels sera appliqué quand le critère "`serviceKey`" et les chiffres initiaux du numéro du demandeur, ou le numéro de zone d'une tentative d'appel satisferont aux critères spécifiés dans "`gapCriteria`".

b) `gapIndicators`:

ce paramètre indique quelles sont les caractéristiques de l'espacement.

i) `durée`:

la durée spécifie l'intervalle total de temps pendant lequel l'espacement des appels pour les critères spécifiés d'espacement sera activé;

une durée de 0 indique que l'espacement doit être supprimé;

une durée de -1 indique une durée illimitée;

une durée de -2 indique une durée spécifique au réseau;

d'autres valeurs indiquent une durée en secondes.

ii) `gapInterval` (Intervalle d'espacement):

ce paramètre spécifie le temps minimal entre les appels devant être autorisé;

un intervalle de 0 indique que les appels satisfaisant aux critères d'espacement ne doivent pas être rejetés;

un intervalle de -1 indique que tous les appels satisfaisant aux critères d'espacement doivent être rejetés;

d'autres valeurs indiquent un intervalle en millisecondes.

c) `controlType` (Type de contrôle):

ce paramètre indique la raison pour laquelle l'espacement des appels doit être activé;

la valeur "`controlType`" ou "`SCPOverloaded`" (Surcharge SCP) indique qu'une détection automatique d'encombrement et qu'un mécanisme de contrôle dans l'entité SCP a détecté une situation d'encombrement;

la valeur "`controlType`" de type "`manuallyInitiated`" (déclenché manuellement) indique que le service et/ou le centre de gestion de réseau/de service a détecté une situation d'encombrement, ou toute autre situation nécessitant des contrôles déclenchés manuellement²⁾.

²⁾ Le paramètre "`controlType`" de type "`manuallyInitiated`" (déclenchement manuel) aura la priorité sur l'espacement des appels "`SCPOverloaded`" (surcharge SCP).

Il convient de noter que le mécanisme de commande de trafic non-RI peut s'appliquer à un commutateur ayant les fonctions de l'entité SSF. Le contrôle de trafic non-RI peut également avoir une influence sur l'appel RI. C'est pourquoi il est recommandé de prendre des mesures afin de coordonner plusieurs mécanismes de contrôle de trafic. Le contrôle de trafic non-RI et la coordination de plusieurs mécanismes de contrôle de trafic sont hors du domaine d'application du sous-protocole INAP.

d) `gapTreatment` (Traitement des espacements):

ce paramètre indique la façon dont les appels qui ont été interrompus par le mécanisme d'espacement des appels doivent être traités.

i) `informationToSend` (Informations à envoyer):

ce paramètre indique une annonce, une tonalité ou une information à visualiser devant être envoyée au demandeur. L'appel sera libéré après que l'information aura été envoyée.

- `inbandInfo` (Informations dans la bande):

ce paramètre spécifie les informations dans la bande qui doivent être envoyées.

- `messageID` (identificateur de message):

ce paramètre indique le(s) message(s) devant être envoyé(s), pouvant être choisi(s) parmi les messages suivants:

- 1) `elementaryMessageID` (identificateur de message élémentaire):

ce paramètre indique une annonce simple.

- 2) `texte`:

ce paramètre indique un texte devant être envoyé. Le texte sera transformé en informations dans la bande (parole). Ce paramètre comprend deux sous-paramètres, `messageContent` (contenu du message) et `attributs`. Les attributs du texte peuvent comprendre des éléments comme le langage.

- 3) `elementary MessageIDs` (identificateurs de message élémentaire):

ce paramètre spécifie une séquence d'annonces.

- 4) `variableMessage` (message variable):

ce paramètre spécifie une annonce ayant une ou plusieurs parties de longueur variable.

- `numberOfRepetitions` (nombre de répétitions):

ce paramètre indique combien de fois au maximum le message sera envoyé à l'utilisateur.

- `durée`:

ce paramètre indique la durée maximale en secondes pendant laquelle le message devra être joué/répété. ZÉRO indique une répétition sans fin.

- `intervalle`:

ce paramètre indique l'intervalle de temps maximal en secondes entre les répétitions, c'est-à-dire le temps en secondes qui s'est écoulé entre la fin de l'annonce et le début de la répétition suivante. Ce paramètre peut être uniquement utilisé quand le nombre de répétitions est > 1.

- `tonalité`:

ce paramètre spécifie la tonalité qui doit être envoyée à l'utilisateur.

- `toneID` (identificateur de tonalité):

ce paramètre indique la tonalité qui doit être envoyée.

- `durée`:

ce paramètre indique la durée en secondes de la tonalité qui doit être envoyée. ZÉRO indique une durée illimitée.

- `displayInformation`:

ce paramètre indique une chaîne de texte devant être envoyée à l'utilisateur. Ces informations ne peuvent pas être reçues par un usager du RTPC.

ii) `releaseCause` (cause de libération):

ce paramètre indique que l'appel sera libéré en utilisant la cause de libération indiquée. Voir la Recommandation Q.762.

iii) les deux derniers:

ce paramètre indique les informations dans la bande, une tonalité ou une information de visualisation devant être envoyée au demandeur. Lorsque l'envoi des informations est terminé, l'appel est libéré en utilisant la cause de libération indiquée.

3.3.10.2 Entité d'appel (SCF)

3.3.10.2.1 Procédure normale

Précondition de l'entité SCF:

- l'entité SCF détecte qu'une condition de surcharge persiste et que l'espacement des appels doit être déclenché dans l'unité SSF; ou
- l'entité SCF reçoit une demande d'espacement des appels déclenchée manuellement.

Postcondition de l'entité SCF:

- la machine SCME-FSM reste dans le même état en présentant l'opération "CallGap" (espacement des appels).

Un algorithme de détection et de gestion des encombrements surveille la charge des ressources SCP. Après la détection d'une situation d'encombrement, les paramètres de l'opération "CallGap" (espacement des appels) sont fournis.

Si le niveau d'encombrement change, de nouvelles opérations "CallGap" peuvent être envoyées pour des critères actifs d'espacement, mais avec un nouvel intervalle d'espacement. L'espacement peut être supprimé si aucun encombrement n'est détecté.

Un espacement des appels déclenché manuellement aura la priorité sur un espacement des appels déclenché automatiquement.

3.3.10.2.2 Traitement des erreurs

Le traitement des erreurs relatives aux opérations n'est pas applicable, en raison de l'opération de classe 4.

3.3.10.3 Entité de réponse (SSF)

3.3.10.3.1 Procédure normale

Précondition de l'entité SSF:

- l'espacement des appels pour gapCriteria (critères d'espacement) n'est pas activé; ou l'espacement des appels pour gapCriteria (critères d'espacement) est activé.

Postconditions de l'entité SSF:

- 1) l'entité SSME-FSM est dans l'état "Traitement non associé à l'appel";
- 2) l'espacement des appels pour gapCriteria est activé; ou l'espacement des appels pour gapCriteria est renouvelé; ou l'espacement des appels pour gapCriteria est supprimé.

S'il n'y a pas d'entité SSME-FSM déjà existante pour les critères d'espacement fournis, une nouvelle entité SSME-FSM est créée. Cette entité SSME-FSM entre dans l'état "Traitement non associé à l'appel" et déclenche l'espacement des appels pour les appels spécifiés du RI. Les paramètres "gapIndicators", "controlType" et "gapTreatment" pour les critères d'espacement indiqués seront mis à la valeur prévue par l'opération "CallGap".

En général, l'espacement des appels déclenché manuellement aura la priorité sur l'opération "sCPOverloaded" (surcharge SCP) déclenchée automatiquement. De façon plus spécifique, les règles suivantes seront appliquées dans l'entité SSF pour gérer la priorité de différents Types de contrôle associés aux mêmes critères d'espacement "gapCriteria".

- Si une entité SSME-FSM existe déjà pour les critères "gapCriteria" fournis:
 - 1) si le (nouveau) paramètre "controlType" équivaut déjà au paramètre "controlType" existant, alors le nouveau paramètre (c'est-à-dire "gapIndicators" et "gapTreatment") écrasera les valeurs de paramètres existantes;
 - 2) si le (nouveau) paramètre "controlType" est différent du paramètre "controlType" existant, les nouveaux paramètres (c'est-à-dire "controlType", "gapIndicators", et "gapTreatment") seront ajoutés à l'entité SSME-FSM appropriée (en plus des paramètres existants). L'entité SSME-FSM reste dans l'état "Traitement non associé à l'appel".

Si l'entité SSF rencontre un point TDP, elle vérifie si l'espacement des appels a été déclenché soit pour le paramètre "serviceKey" soit pour la valeur "calledAddressValue" attribuée à ce point TDP. Dans le cas contraire, il est possible d'envoyer un point de détection initial "InitialDP" ou une opération spécifique au point DP. Au cas où l'espacement

d'appel serait déclenché pour l'une des opérations "calledAddressAndService" ou "callingAddressAndService" et où le paramètre "serviceKey" correspond, une vérification des paramètres "calledAddressValue" et "callingAddressValue" - et facultativement du paramètre "location Number" - pour l'espacement activé des appels sera effectuée. Dans le cas contraire, il est possible d'envoyer un "InitialDP"(point de détection initial) ou une opération spécifique au point DP.

En cas d'espacement des appels dans l'opération "callingAddressAndService" et quand on dispose du paramètre "locationNumber", l'espacement sera exécuté sur "locationNumber" au lieu de "callingAddressValue".

Si un appel sur un numéro sous commande ne satisfait qu'un critère "gapCriteria" activé, la commande correspondante est appliquée. Si l'une des deux commandes "manuallyInitiated" ou "sCPOverload" est activée, seule la commande à déclenchement manuel sera appliquée.

Si un appel sur un numéro sous commande satisfait plusieurs critères "gapCriteria" activés, seuls les critères d'espacement associés au numéro le plus long de demandé doivent être utilisés, et la commande correspondante doit être effectuée. Par exemple, les codes 1234 et 12345 sont sous commande. Alors, l'appel avec 123456 est soumis à la commande sur 12345. En outre, si les opérations des deux types de commande, à savoir "manuallyInitiated" et "sCPOverloaded" (Surcharge SCP) sont activées pour ces critères "gapCriteria", la commande "manuallyInitiated" sera utilisée.

Si l'espacement des appels doit être appliqué et s'il n'y pas d'intervalle d'espacement activé, une opération de type "InitialDP" ou une opération spécifique au point DP peut être envoyée, à l'inclusion du paramètre "espacement des appels rencontrés" selon le type de contrôle spécifié. Un nouvel intervalle d'espacement sera déclenché comme l'indique "gapInterval".

Si un intervalle d'espacement est actif, aucune opération "InitialDP" ou spécifique au point DP n'est envoyée et l'appel est traité comme l'indique le paramètre "gapTreatment".

Le processus d'espacement des appels est arrêté si la durée indiquée est égale à ZÉRO.

A mesure que l'espacement des appels se poursuit, l'entité SSME-FSM passe à l'état "Traitement non associé à l'appel". Autrement, l'entité SSME-FSM passe à l'état "idle management".

3.3.10.3.2 Traitement des erreurs

L'opération de traitement des erreurs n'est pas applicable, en raison de l'opération de classe 4.

3.3.11 CallInformationReport procedure (Procédure de comptes rendus d'appel)

3.3.11.1 Description générale

Cette opération est utilisée pour envoyer des informations d'appel spécifiques relatives à un appel unique vers l'entité SCF comme le demande l'entité SCF dans une opération "CallInformationRequest" précédente. Le compte rendu est envoyé à la fin de l'appel qui est indiqué par l'un des événements spécifiés ci-dessous.

3.3.11.1.1 Paramètres

- requestedInformationList (Liste des informations demandées):
selon les informations demandées, l'entité SSF envoie les types et les valeurs qui conviennent vers l'entité SCF.

3.3.11.2 Entité d'appel (SSF)

3.3.11.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) l'appel est interrompu par au moins un correspondant ou l'établissement de l'appel n'est pas réalisé;
- 2) les informations demandées relatives à l'appel ont été recueillies;
- 3) l'opération "CallInformationReport" est en cours de traitement en raison d'une opération "CallInformationRequest" précédemment reçue;
- 4) une relation de commande existe entre les entités SCF et SSF.

Postcondition de l'entité SSF:

- l'entité SSF-FSM passera dans la position "Idle" dans le cas où il n'y aurait pas d'autres demandes de comptes rendus en cours et dans le cas où un autre point EDP ne serait pas armé, sinon cette entité devra rester dans le même état.

Si l'entité SSF-FSM effectue un changement d'état provoqué par l'un des événements suivants:

- libération de correspondant;
- interruption du correspondant;
- libération de correspondant B;
- correspondant B occupé;
- fin de temporisation sur l'entité non-réponse SSF;
- échec de sélection d'itinéraire détecté par le réseau;
- libération d'appel déclenchée par l'entité SCF,

et l'opération "CallInformationRequest" est en cours pendant qu'une opération "CallInformationReport" est envoyée vers l'entité SCF contenant toutes les informations demandées.

Si une opération "CallInformationReport" a été envoyée vers l'entité SCF, alors aucune opération "CallInformation Report" n'est en cours, c'est-à-dire qu'une autre opération "CallInformationReport", par exemple dans le cas de reprise de numérotation, doit être explicitement demandée par l'entité SCF.

Si un événement provoquant l'opération "CallInformationReport" est également déclenché par un point EDP-R armé, alors l'opération correspondante "EventReportBCSM" doit être envoyée immédiatement après l'opération "CallInformationReport".

Si un événement provoquant l'opération "CallInformationReport" est également déclenché par un point EDP-N armé, l'opération "EventReportBCSM" correspondante doit être envoyée juste avant l'opération "CallInformationReport".

3.3.11.2 Traitement des erreurs

Le traitement des erreurs relatives aux opérations n'est pas applicable, en raison de l'opération de classe 4.

3.3.11.3 Entité de réponse (SCF)

3.3.11.3.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une instance SLPI attend une opération "CallInformationReport";
- 2) une relation de commande existe entre les entités SCF et SSF.

Postcondition de l'entité SCF:

- l'instance SLPI peut poursuivre l'exécution.

Dans tout état (sauf l'état "Idle"), l'entité SCSM peut recevoir une opération "CallInformation Report" provenant de l'entité SSF, quand l'opération "CallInformationReport" est en cours d'exécution.

Si l'opération "CallInformationReport" est en cours d'exécution et si le programme de logique de traitement de service indique que le traitement est terminé, l'entité SCSM reste dans le même état jusqu'à ce que l'opération "CallInformationReport" ait été reçue.

Si l'entité SCF reçoit l'opération "CallInformationReport" et si la logique de traitement de service est terminée, l'entité SCSM passe à l'état "Idle".

Si l'entité SCF reçoit l'opération "CallInformationReport" et si la logique de traitement de service n'est pas encore terminée, l'entité SCSM reste dans le même état (une des opérations EventReportBCSM ou ApplyChargingReport est en cours d'exécution).

3.2.11.3.2 Traitement des erreurs

Si les informations demandées ne sont pas disponibles, une opération "CallInformationReport" sera envoyée, indiquant le type des informations demandées mais avec un paramètre "RequestedInformationValue" correspondant à la valeur par défaut appropriée.

Le traitement des erreurs relatives aux opérations n'est pas applicable, en raison de l'opération de classe 4.

3.3.12 CallInformationRequest procedure

3.3.12.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF d'enregistrer des informations spécifiques relatives à un appel unique et pour en rendre compte à l'entité SCF en utilisant l'opération "CallInformationReport".

3.3.12.1.1 Paramètres

- requestedInformationTypeList (liste de types d'informations demandées):

ce paramètre spécifie une liste d'éléments d'information qui est demandée. Cette liste peut contenir:

- callAttemptElapsedTime (Temps écoulé pendant la tentative d'appel):

ce paramètre indique la durée écoulée entre la fin du traitement des opérations du sous-système INAP déclenchant l'établissement de l'appel ("Connexion", "AnalyseInformation", "CollectInformation", "Continue" et "SelectRoute" et l'indication de la réponse reçue en provenance du demandé.

En cas de non-établissement de l'appel, le réseau indiquant que l'appel n'a pas été établi arrête la mesure du paramètre "callAttemptElapsedTime" (temps écoulé pendant la tentative d'appel).

- callStopTime (Heure d'arrêt de l'appel):

ce paramètre indique le moment où la connexion est libérée.

- callConnectedElapsedTime (Temps écoulé pour l'appel connecté):

ce paramètre indique la durée écoulée entre l'indication de la réponse reçue par le demandé et la libération de la connexion.

- calledAddress (Adresse demandée):

ce paramètre indique l'adresse du demandé entrant qui a été reçue par l'entité SSF (c'est-à-dire avant traduction par l'entité SCF) et est disponible sur les interfaces UNI et NNI et interprété selon le plan de numérotation.

- releaseCause (cause de libération):

voir la Recommandation Q.762.

Tout ensemble comprenant ces valeurs peut être demandé.

3.3.12.2 Entité d'appel (SCF)

3.3.12.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) l'instance SLPI a déterminé qu'une opération "CallInformationRequest" devait être envoyée par l'entité SCF.

Postcondition de l'entité SCF:

- l'instance SLPI prévoit une opération "CallInformationReport" en provenance de l'entité SSF.

Si le programme de service de logique de traitement de service demande les informations d'appel, l'entité SCF envoie l'opération "CallInformationRequest" vers l'entité SSF pour demander à l'entité SSF de fournir les informations relatives à l'appel.

L'opération "CallInformationRequest" spécifie les articles d'information devant être fournis par l'entité SSF.

3.3.12.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.12.3 Entité de réponse (SSF)

3.3.12.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) une relation de commande existe entre les entités SSF et SCF.

Postconditions de l'entité SSF:

- 1) les informations d'information demandées sont conservées par l'entité SSF;
- 2) l'entité SSF attend d'autres instructions.

L'entité SSF peut recevoir l'opération "CallInformationRequest" uniquement à l'intérieur d'un dialogue existant associé à l'appel.

L'opération "CallInformationRequest" est acceptée par l'automate SSF (SSF-FSM) uniquement dans l'état "Attente des instructions". L'opération n'entraîne pas de changement vers un autre état.

L'entité SSF attribue un enregistrement et emmagasine les informations demandées si elles sont déjà disponibles et réserve la place des articles qui deviendront disponibles ultérieurement comme par exemple dans "callStopTimeValue".

3.3.12.3.2 Traitement des erreurs

Dans tout état autre que l'opération "Attente des instructions", l'opération "CallInformationRequest" sera traitée comme une erreur dont le code d'erreur sera "UnexpectedComponentSequence".

Le traitement générique des erreurs relatives aux opérations est décrit dans 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits dans 3.4.

3.3.13 Procédure d'annulation

3.3.13.1 Description générale

L'entité SCF utilise cette opération de classe 2 afin de demander à l'entité SRF/SSF d'annuler une opération précédente corrélée.

L'opération à supprimer peut soit être une opération "PlayAnnouncement" (PA) ou une opération "PromptAndCollectUserInformation".

L'annulation d'une opération est indiquée par l'intermédiaire d'une indication d'erreur "Annulé", en direction de l'entité d'appel respectivement à une des opérations "PlayAnnouncement" ou "PromptAndCollectUserInformation". L'opération "Annuler" peut également être utilisée pour annuler toutes les demandes en cours d'exécution et pour permettre à la machine à états SSF d'aller vers l'état "Idle". Dans ce cas, l'opération "Annuler" ne précise aucune opération spécifique devant être annulée.

3.3.13.1.1 Paramètres

- invokeID (identificateur d'appel):
ce paramètre spécifie quelle opération doit être annulée, c'est-à-dire l'une des opérations PromptAndCollectUserInformation ou PlayAnnouncement.
- allRequests (toutes demandes):
ce paramètre indique que toutes les demandes actives de comptes rendus EDP (génériques ou spécifiques au point DP) d'opération "ApplyChargingReport" et "CallInformationReport" doivent être annulées.

NOTE – Cette annulation est différente du mécanisme d'annulation sur lequel repose l'identificateur d'appel invokeID décrit ci-dessus.

3.3.13.2 Entité d'appel (SCF)

3.3.13.2.1 Procédure normale

L'entité SCF peut invoquer cette opération vers l'une des entités SSF ou SRF, différentes conditions prévalant dans chaque cas.

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF/SRF;
- 2) une instance SLPI dans l'état "Attente de la réponse en provenance de l'entité SRF" a détecté qu'une opération précédemment demandée devait être annulée; ou
une instance SLPI a détecté qu'elle n'était plus intéressée par des comptes rendus ou des notifications provenant de l'entité SSF et qu'il devrait être mis fin à la relation de commande.

Postcondition de l'entité SCF:

- l'instance SLPI reste dans l'état "Attente de la réponse provenant de l'entité SRF"; ou
au cas où toutes les demandes sont annulées, il est mis fin à la relation de commande avec l'entité fonctionnelle (SSF) et l'automate SCSM-FSM retourne à l'état "Idle".

3.3.13.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.13.3 Entité de réponse (SRF)

3.3.13.3.1 Procédure normale

Précondition de l'entité SRF:

- une opération PlayAnnouncement ou PromptAndCollectUserInformation a été reçue et l'entité SRF est dans l'état "Interaction avec l'utilisateur".

Postcondition de l'entité SRF:

- l'exécution de l'opération PlayAnnouncement ou PromptAndCollectUserInformation a été interrompue et l'entité SRF reste dans l'état "Interaction avec l'utilisateur".

3.3.13.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.13.4 Entité de réponse (SSF)

3.3.13.4.1 Procédure normale

Précondition de l'entité SSF:

- l'automate SSF-FSM est dans l'état "Attente des instructions" ou "Surveillance".

Postcondition de l'entité SSF:

- 1) toutes les demandes actives de comptes rendus et de notifications ont été annulées;
- 2) dans le cas où l'automate SSF-FSM était dans l'état "Surveillance", il devra retourner à l'état "Idle"; ou dans le cas où l'automate SSF-FSM était dans l'état "Attente des instructions", il restera dans cet état.
Une opération subséquente de traitement d'appel entraînera le passage de l'automate SSF-FSM dans l'état "Idle". L'appel, s'il est dans l'état actif, est en outre traité par l'entité SSF de façon autonome comme un appel normal (non-RI).

3.3.13.4.2 Traitement des erreurs

L'envoi du retour erreur sur "annuler" n'est pas applicable dans le cas de l'annulation "allRequests". Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.14 Procedure CollectedInformation (informations recueillies)

3.3.14.1 Description générale

Cette opération est envoyée par l'entité SSF après détection d'un des points TDP-R, TDP-N, EDP-R ou EDP-N dans l'automate BCSM d'origine, afin d'indiquer qu'une chaîne complète de numérotation ou qu'un paquet initial de services est disponible en provenance de l'origine.

3.3.14.1.1 Paramètres

- serviceAddressInformation (triggerType, serviceKey, miscCallInfo):
voir la Recommandation Q.1290. Les événements actuellement définis qui sont applicables au point de détection Collected_Information sont Off_Hook_Delay, Channel_Setup_PRI, et Shared_Interoffice_Trunk. Les informations miscCallInfo indiquent qu'une demande d'instructions a été présentée à l'entité SCF et pour la catégorie de déclenchement choisie (ligne, faisceau, entreprise). Le paramètre serviceKey est utilisé pour indiquer la logique de traitement de service devant être invoquée.
- bearerCapability (Capacité du support):
ce paramètre indique le type de la capacité du support de la connexion vers l'utilisateur. Il convient de se référer au 6.4.4/Q.1214 en ce qui concerne les règles de codage du paramètre capacité du support.
- calledPartyNumber (Numéro du demandé):
voir la Recommandation Q.762 "Paramètre de l'information de signalisation du numéro du demandé". Ce paramètre sert à indiquer le numéro du demandé dans le sens en avant.
- callingPartyNumber (Numéro du demandeur):
voir la Recommandation Q.762 "Calling Party Number signalling information" (Information de signalisation: numéro du demandeur). Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre d'information de signalisation: catégorie du demandeur.
- callingPartysCategory (Catégorie du demandeur):
voir la Recommandation Q.762 "Catégorie du demandeur" signalant les informations. Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre "catégorie de demandeur".
- capacités du périphérique intelligent intégré au point SSP ("service switching point" ou point de commutation de service):
voir la Recommandation Q.1290.
- iPAvailable (Périphérique intelligent disponible):
voir la Recommandation Q.1290.
- iSDNAccessRelatedInformation (Informations relatives à l'accès RNIS):
ce paramètre contient des éléments d'information (éventuellement multiples) selon Q.931. Voir l'information au sujet de la signalisation du paramètre de transport à l'accès Q.762, Q.763 et Q.931. Il convient de se référer aux règles de définition au 6.4.4/Q.1214 pour les règles de codage des informations relatives à l'accès RNIS.
- cGEncountered (détection d'un espacement des appels):
voir la Recommandation Q.1290.
- locationNumber (Numéro d'emplacement):
ce paramètre est utilisé pour transporter l'adresse de zone géographique pour les services mobiles, voir la Recommandation Q.762. Il est utilisé quand le paramètre "callingPartyNumber" ne contient pas d'informations relatives à l'emplacement géographique du demandeur (par exemple l'origine dépend de l'acheminement quand le demandeur est un abonné mobile).
- serviceProfileIdentifier (Identificateur de profil de service):
voir l'Annexe A/Q.932. Voir 6.4.4/Q.1214 pour les règles d'insertion du paramètre serviceProfileIdentifier.
- terminalType (Type de terminal):
voir la Recommandation Q.1290. Caractérise le type de terminal de sorte que l'entité SCF puisse spécifier à l'entité SRF la fonction de terminal (reconnaissance de la parole, emploi du signal multifréquence à deux tonalités (DTMF), emploi de l'affichage, etc.).
- chargeNumber (Numéro à taxer):
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 en ce qui concerne les règles d'insertion du paramètre chargeNumber (numéro à taxer).
- servingAreaID (identificateur de zone de desserte):
voir la Recommandation Q.1290.

- dialledDigits (chiffres numérotés):
voir la Recommandation Q.1290. Il convient de se référer au 6.4.4/Q.1214 en ce qui concerne les règles d'insertion s'appliquant à dialledDigits.
- callingPartyBusinessGroupID (Identification d'un groupe professionnel appelant):
voir la Recommandation Q.1290.
- callingPartySubaddress (Sous-adresse du demandeur):
voir la Recommandation Q.931 pour les détails.
- callingFacilityGroup (Faisceau de ressources du demandeur):
voir la Recommandation Q.1290.
- callingFacilityGroupMember (Elément du faisceau de ressources du demandeur):
voir la Recommandation Q.1290.
- originalCalledPartyID (Identification du demandé initial):
voir la Recommandation Q.762. Informations de signalisation du numéro de demandé initial. Se référer au 6.4.4/Q.1214 pour les règles d'insertion du paramètre originalCalledPartyID.
- préfixe:
voir la Recommandation Q.1290.
- redirectingPartyID (Identificateur de réachemineur):
ce paramètre contient le numéro d'annuaire du dernier réachemineur.
- redirectionInformation (Information de réacheminement):
voir la Recommandation Q.763. Informations de signalisation des informations de réacheminement.
- travellingClassMark (Marque de classe de transport):
voir la Recommandation Q.1290. Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre travellingClassMark.
- featureCode:
voir la Recommandation Q.1290.
- accessCode:
voir la Recommandation Q.1290.
- exploitant:
voir la Recommandation Q.1290.

3.3.14.2 Entité d'appel (SSF)

3.3.14.2.1 Procédure normale

Préconditions de l'entité SSF pour l'un des points TDP-R et TDP-N:

- 1) un événement dont le contenu satisfait aux critères de déclenchement s'est produit au point de détection Collected_Information (c'est-à-dire Off_Hook_Delay, Channel_Setup_PRI, Shared_Interoffice_Trunk);
- 2) l'espacement des appels et la surcharge SS7 ne s'appliquent pas à l'appel, et l'appel n'a pas à être filtré.

Préconditions de l'entité SSF pour l'un des points EDP-R et EDP-N:

- 1) une relation de commande a été établie et l'entité SCF demande l'opération Collected_Information comme point EDP-R ou EDP-N;
- 2) le traitement d'appel atteint le point de détection Collected_Information.

Postcondition de l'entité SSF pour le point TDP-R:

- une relation de commande est maintenue et l'entité SSF attend des instructions en provenance de l'entité SCF.

Postcondition de l'entité SSF pour le point EDP-R:

- la relation de commande est maintenue et l'entité SSF attend des instructions en provenance de l'entité SCF.

Postcondition de l'entité SSF pour les points TDP-N et EDP-N:

- le traitement de l'appel continue au point de traitement `Analysed_Information`.

L'entité SSF/CCF recueille suffisamment d'informations (c'est-à-dire un code d'option, un préfixe, des informations relatives à l'adresse, etc.) à partir de l'accès d'origine pour traiter l'appel. Ces informations sont recueillies selon le plan de numérotage attribué à l'accès d'origine:

- 1) pour un appel provenant d'une ligne non-RNIS, l'entité SSF/CCF envoie une invite dans la bande (par exemple une tonalité d'invitation à numéroté) pour prévenir l'utilisateur, elle affecte un enregistreur, recueille les chiffres et applique le rythme interchiffre. A partir d'une ligne non-RNIS, le "plan de numérotage en vigueur" est celui attribué à la ligne. Par exemple, si un plan de numérotage privé (PDP) (auquel il est parfois fait référence comme le Plan de numérotage adapté aux besoins de l'utilisateur) est attribué, le PDP est considéré être "le plan de numérotage en vigueur". Si, pour un appel donné, le demandeur compose un code d'accès afin d'atteindre un "Plan de numérotage de service public" (PODP) (c'est-à-dire E.164), on considère que le PODP est le plan de numérotage en vigueur. Dans ce cas, le traitement d'appel ne peut pas retourner vers le plan de numérotage privé (PDP) dans cette entité SSF/CCF. Si un plan privé PDP n'est pas attribué, on suppose qu'un plan public PODP est le "plan de numérotation privé en vigueur";
- 2) pour un appel émanant d'un usager du réseau RNIS utilisant les procédures d'envoi *en bloc* (c'est-à-dire que toutes les informations nécessaires pour traiter l'appel, ce qui peut inclure un code d'option, un activateur d'option, un numéro demandé, etc., sont comprises dans le message ÉTABLISSEMENT), aucune autre action n'est entreprise sur l'interface d'accès de départ. Pour un appel effectué par un usager de réseau RNIS qui n'utilise pas les procédures d'envoi *en bloc*, l'entité SSF/CCF avertit l'utilisateur avec un message d'accusé de réception ÉTABLISSEMENT afin d'envoyer davantage d'informations. Les procédures d'envoi par chevauchement sont alors employées.

Pour un appel RNIS émanant d'un accès de base ou d'un accès primaire, "le plan de numérotage en vigueur" est déterminé par le type de numérotation et le champ du plan de numérotage de l'élément d'information numéro du demandé. Si ce champ indique qu'un élément "inconnu" ou qu'un élément d'information du clavier est utilisé, le plan de numérotage est alors déterminé comme cela est défini ci-dessus pour les lignes non-RNIS.

- 3) pour des jonctions classiques, si le type de jonction le requiert, un signal de départ approprié est renvoyé. Un enregistreur est affecté, les chiffres étant recueillis et un rythme interchiffre appliqué.

Pour un appel émanant d'une jonction classique, il est prévu que le numéro reçu dans la signalisation soit un numéro public qui sera conforme à la Recommandation E.164;

- 4) pour les jonctions sur réseau sémaphore, toutes les informations peuvent être fournies dans le message initial d'adresse (IAM) ou arriver dans des messages subséquents (c'est-à-dire envoi par chevauchement). Une vérification est effectuée pour s'assurer que le message IAM contient toutes les informations nécessaires au traitement de l'appel.

Pour un appel émanant d'une jonction sur réseau sémaphore, il est prévu que le nombre reçu dans la signalisation soit un numéro public qui sera conforme à la Recommandation E.164;

- 5) pour les jonctions d'équipements privés, un signal de départ approprié est retourné si le groupe de faisceaux de circuits d'équipements privés le demande. Selon le groupe de faisceaux de circuits d'équipements privés, des chiffres peuvent être recueillis.

Suite à la détection de déclenchement (les critères attribués au point de détection étant satisfaits) qui s'applique à un point TDP-R ou TDP-N armé dans l'automate BCSM au point de détection `Collected_Information`, causé par une tentative d'appel initial, l'entité SSF vérifie si l'espacement des appels, la surcharge SS7 ou le filtrage de service ne s'appliquent pas au segment d'appel considéré. Si ces conditions sont satisfaites, l'opération `CollectedInformation` est invoquée par l'entité SSF. L'adresse de l'entité SCF vers laquelle l'opération `CollectedInformation` doit être envoyée est déterminée sur la base des données de l'instance de déclenchement. L'entité SSF fournit des paramètres selon les règles définies au 6.4.4/Q.1214. Pour un point TDP-R, une relation de commande est établie vers l'entité SCF et le temporisateur d'application T_{SSF} de l'entité SCF est chargé quand l'entité SSF envoie l'opération `CollectedInformation` pour demander des instructions à l'entité SCF. Le temporisateur d'application T_{SSF} est chargé pour empêcher un temps d'attente excessif. Pour un point EDP-R, le temporisateur T_{SSF} est utilisé également. Pour les points TDP-N et EDP-N, aucune nouvelle relation de commande n'est établie. C'est pourquoi le temporisateur d'application T_{SSF} n'est pas chargé.

3.3.14.2 Traitement des erreurs

Si l'entité SCF de destination n'est pas accessible, alors la fin de traitement d'appel est effectuée (les autres traitements nécessitent un complément d'étude). A la fin de la temporisation T_{SSF} , si aucune opération n'est reçue, l'entité SSF interrompt l'interaction avec l'entité SCF et le traitement de fin d'appel est effectué (par exemple acheminement vers une annonce de fin d'appel). Si le demandeur interrompt l'appel après l'envoi de l'opération "CollectedInformation", l'entité SSF interrompt la relation de commande après la réception du premier message en provenance de l'entité SCF. Si le demandeur interrompt l'appel et si la temporisation T_{SSF} est en cours d'exécution, l'identificateur de transaction reste libre jusqu'à ce qu'un critère de fin soit satisfait. Si le demandeur interrompt l'appel et si la temporisation n'est pas en cours d'exécution, les procédures applicables à la situation donnée (par exemple les procédures destinées à rendre compte de la condition "Caller Abandon", les procédures associées à la condition de fin de temporisation T_{SSF}) seront utilisées.

Le traitement générique des erreurs relatives aux opérations est décrit au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.14.3 Entité de réponse (SCF)

3.3.14.3.1 Procédure normale

Précondition SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R de l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle déroule l'exécution et prend fin.

Postcondition de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et s'il y a d'autres points EDP armés ou si une opération du type "CallInformationReport" ou "ApplyChargingReport" a été demandée; ou
l'automate SCSM-FSM passe à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération du type "CallInformationReport", ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM passe à l'état "Préparer les instructions SSF" si le type de message était une demande.

En ce qui concerne le point TDP-R, à la réception de l'opération CollectedInformation, l'entité SCSM passe de l'état "Repos" à l'état "Préparation des instructions SSF". Une relation de commande est créée avec l'entité SSF connexe. Une instance de programme de logique de traitement de service (SLPI) est invoquée pour traiter l'opération CollectedInformation selon le paramètre triggerType. Au moyen de cette relation de commande, l'entité SCF peut influencer le traitement d'appel de base selon la logique de traitement de service invoquée. Les actions devant être exécutées dans l'instance SLPI dépendent des paramètres transmis par l'intermédiaire de cette opération et de l'instance SLPI elle-même (c'est-à-dire le service RI qui a été lui-même demandé).

En ce qui concerne le point EDP-R, à la réception de l'opération CollectedInformation, l'entité SCSM passe de l'état "Attente de la notification" à l'état "Préparation des instructions SSF". Une relation de commande avec l'entité SSF connexe est maintenue. Une instance de programme de logique de traitement de service (SLPI) est invoquée pour traiter l'opération CollectedInformation selon le paramètre triggerType. Au moyen de cette relation de commande, l'entité SCF peut influencer le traitement d'appel de base selon la logique de traitement de service invoquée. Les actions devant être exécutées dans l'instance SLPI dépendent des paramètres transmis par l'intermédiaire de cette opération et l'instance SLPI elle-même (c'est-à-dire le service RI qui a lui-même été demandé).

Pour le point TDP-N, l'entité SCSM reste dans l'état "Repos" et prend les mesures appropriées sur la notification.

Pour le point EDP-N, s'il s'agit de la dernière notification de l'événement et s'il n'y a pas d'opération CallInformationReport et ApplyChargingReport en cours, l'entité SCSM passe de nouveau à l'état "Repos". Dans le cas contraire, elle restera dans l'état "Attente de la notification ou du compte rendu". Des mesures appropriées sont prises en ce qui concerne la notification.

3.3.14.3.2 Traitement des erreurs

Si l'opération CollectedInformation est rejetée, l'entité SCSM reste dans le même état. La fonction de maintenance est informée et aucune instance SLPI n'est invoquée. Le traitement générique des erreurs relatives aux opérations est décrit au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.15 Procédure CollectInformation

3.3.15.1 Description générale

L'information CollectInformation est une opération de classe 2 qui est utilisée par l'entité SCF pour demander que l'appel retourne au point de traitement d'appel Collect_Information, et pour effectuer les actions de traitement d'appel initial de base associées à ce point de traitement (par exemple la vérification des informations dans le paramètre calledPartyNumber avec le plan de numérotage en service). Cette opération n'utilise que les ressources de l'entité SSF/CCF afin de recueillir les informations. L'utilisation de cette opération n'est appropriée que pour un appel qui n'a pas encore quitté la phase d'établissement.

Lorsque l'utilisateur fournit les paramètres du numéro calledPartyNumber, le traitement du point de traitement d'appel Collect_Information dispose des informations de destination fournies par le demandeur. Quand le numéro calledPartyNumber (numéro du demandé) est inclus (comme dialledDigits) dans l'opération Collect_Information, il n'y a pas d'autres informations recueillies (par exemple l'entité SSF/CCF vérifie la conformité du paramètre calledPartyNumber fourni par rapport au plan de numérotage en service).

3.3.15.1.1 Paramètres

- alerting Pattern (Séquence d'alerte):
voir la Recommandation Q.1290. Elle ne s'applique que si la signalisation de réseau gère ce paramètre ou si l'entité SSF est le centre local de rattachement de l'abonné.
- numberingPlan (Plan de numérotage):
voir la Recommandation Q.763 Plan de numérotage pour le codage.
- originalCalledPartyID (Identificateur du demandé initial):
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.
- travelingClassMark (Marque de classe de transport):
voir la Recommandation Q.1290. Ce paramètre contient les informations relatives à la marque de classe de transport du demandeur. Elle utilise le format LocationNumber qui fait référence au format du numéro d'emplacement contenu dans la Recommandation Q.763.
- callingPartyNumber (Numéro du demandeur):
voir la Recommandation Q.762.
- dialledDigits (Chiffres composés):
la conformité de ce paramètre est vérifiée par rapport au plan de numérotage en service dans l'entité SSF/CCF et, s'il est correct, il est utilisé dans l'acheminement de l'appel. Le cas échéant, il remplace le numéro calledPartyNumber pour l'appel.

3.3.15.2 Entité d'appel (SCF)

3.3.15.2.1 Procédure normale

Précondition de l'entité SCF:

- une instance SLPI a déterminé que davantage d'informations en provenance du demandeur étaient nécessaires pour permettre la suite du traitement.

Postcondition de l'entité SCF:

- l'exécution de l'instance SLPI est interrompue pendant la réception des chiffres composés.

Cette opération est invoquée dans l'état "Préparation des instructions SSF" de l'automate SCSM-FSM si le programme de logique de traitement de service demande des informations complémentaires pour permettre la poursuite du traitement de l'appel. Elle provoque le passage de l'entité FSM vers l'état "Attente de la notification ou du compte rendu".

3.3.15.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.15.3 Entité de réponse (SSF)

3.3.15.3.1 Procédure normale

Précondition de l'entité SSF:

- une demande TDP a été invoquée.

Postconditions de l'entité SSF:

- 1) l'entité SSF a exécuté une transition vers l'état "Surveillance";
- 2) l'entité SSF effectue les actions de traitement d'appel afin de recueillir des informations de destination en provenance du demandeur. Cela peut comprendre l'avertissement du demandeur avec des signaux dans la bande ou en dehors de la bande;
- 3) le traitement de l'appel de base est repris au point de traitement du point d'appel PIC 2.

L'opération est uniquement valide dans l'état "Attente de l'instruction" et après avoir reçu une opération "RequestReportBCSMEvent" pour le point DP2. L'entité SSP doit effectuer les actions suivantes:

- l'entité SSF annule T_{SSF} ;
- quand le point DP2 sera rencontré, une opération du type "EventReportBCSM" sera invoquée, et l'entité SSF-FSM retournera à l'état "Attente de l'instruction".

3.3.15.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.16 Connect procedure (Procédure de connexion)

3.3.16.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF d'effectuer les actions nécessaires pour acheminer l'appel vers une destination spécifique. A cet effet, l'entité SSF peut utiliser des informations de destination provenant du demandeur (par exemple chiffres composés) et des informations existantes pour l'établissement de l'appel (c'est-à-dire index des acheminements sur une liste de faisceaux de circuits) selon les informations fournies par l'entité SCF.

En général, tous les paramètres qui sont fournis dans une opération "Connect" vers l'entité SSF devront remplacer le paramètre correspondant de signalisation dans le CCF, si le paramètre en question a déjà été reçu dans l'entité CCF, et devront être utilisés pour le traitement d'appel subséquent. Les paramètres qui ne sont pas fournis par l'opération "Connect" garderont leur valeur (si elle a déjà été attribuée) dans l'entité CCF pour le traitement subséquent des appels.

3.3.16.1.1 Paramètres

- destinationRoutingAddress (Adresse d'acheminement à destination):
ce paramètre contient la liste des numéros du demandé (voir Q.762) vers lesquels l'appel doit être acheminé. Le codage de ce paramètre est défini dans Q.763. L'adresse "destinationRoutingAddress" peut également inclure un identificateur "correlationID" et un identificateur "scfID", s'ils sont utilisés dans le contexte de la procédure de transfert de tâches, mais uniquement s'ils ne sont pas spécifiés séparément.
- alertingPattern:
voir la Recommandation Q.1290. Elle ne s'applique que si la signalisation de réseau gère ce paramètre ou si l'entité SSF est le centre local de rattachement de l'abonné.
- correlationID (Identificateur de corrélation):
ce paramètre est utilisé par l'entité SCF pour associer l'opération "AssistRequestInstructions" provenant de l'entité SSF assistante avec la demande de l'entité SSF déclencheuse. L'identificateur "correlationID" est utilisé dans le contexte de la procédure de transfert de tâche et uniquement si l'identificateur de corrélation id n'est pas inclus dans "destinationRoutingAddress". Les exploitants de réseau doivent décider des correspondances exactes de ce paramètre sur le système de signalisation utilisé.

- cutAndPaste (couper et coller):
ce paramètre est utilisé par l'entité SCF pour donner l'ordre à l'entité SSF d'effacer (de couper) un nombre spécifique de chiffres d'en-tête reçus du demandeur et de coller les chiffres restants à la fin des chiffres fournis par l'entité SCF dans "destinationRoutingAddress".
- forwardingCondition:
indique la condition qui doit être satisfaite pour effectuer l'opération de Connexion.
- iSDNAccessRelatedInformation:
transporte les mêmes informations que l'élément de protocole ISUP paramètre de transport d'accès de la Recommandation Q.762.
- originalCalledPartyID (Identificateur du numéro de demandé initial):
voir la Recommandation Q.762 "Informations de signalisation du numéro de demandé initial". L'utilisation de ce paramètre dans le contexte de l'opération de "Connexion" doit être spécifiée par l'exploitant de réseau.

routeList (Liste d'acheminement):
ce paramètre est utilisé pour sélectionner le groupe de faisceaux de circuits de départ utilisé pour acheminer l'appel. Une séquence d'itinéraires est fournie pour permettre un routage souple d'applications tel que le réseau privé virtuel, sans accroître le nombre de demandes nécessaires à telles applications.
- scfID (Identificateur SCF):
voir la Recommandation Q.1290. L'identificateur scfID est utilisé dans le contexte d'une procédure de transfert de tâche et seulement si l'entité SCF n'est pas incluse dans l'adresse "destinationRoutingAddress". Les exploitants de réseau doivent décider des véritables correspondances de ce paramètre sur le système de signalisation utilisé.
- travellingClassMark:
l'entité SCF utilise le paramètre travellingClassMark pour fournir des informations essentielles relatives au choix de l'acheminement à l'entité SSF. L'entité SSF utilise ces informations pour coder le message ISUP IAM sortant, le codage et les correspondances de ce paramètre étant spécifiques à l'exploitant de réseau.
- carrier (Exploitant de réseau):
voir la Recommandation Q.1290. Dans ce message, le champ de sélection de l'exploitant est zéro (00000000) et l'indicateur d'exploitant indique l'exploitant devant utiliser l'appel.
- serviceInteractionIndicators (Indicateurs d'interaction de service):
ce paramètre contient des indicateurs envoyés de l'entité SCF vers l'entité SSF pour contrôler les services fournis par le réseau au centre d'origine, de même qu'au centre de destination.
- callingPartyNumber (Numéro de demandeur):
ce paramètre, s'il existe, est utilisé pour remplacer le "numéro du demandeur" (voir Q.762 Numéro du demandeur). Il peut être utilisé pour des applications telles que les télécommunications personnelles universelles (UPT), lorsque seule l'entité SCF peut vérifier l'identité du demandeur.
- callingPartysCategory (Catégorie de demandeur):
voir la Recommandation Q.762. Informations de signalisation de la catégorie de demandeur.
- redirectingParty ID (Identificateur de réacheminneur):
ce paramètre, s'il existe, indique le dernier numéro d'annuaire à partir duquel l'appel a été réacheminé.
- redirectionInformation (Informations de réacheminement):
voir l'information de signalisation d'information de réacheminement de la Recommandation Q.763.

3.3.16.2 Entité d'appel (SCF)

3.3.16.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) une instance SLPI a déterminé qu'une opération de "Connexion" devait être envoyée par l'entité SCF.

Postcondition de l'entité SCF:

- l'exécution de l'instance SLPI peut continuer.

Dans l'état "Préparer les instructions SSF" de l'automate SCSM-FSM, cette opération est invoquée par l'entité SCF si la logique de traitement de service aboutit à la demande présentée à l'entité SSF pour acheminer un appel vers une destination spécifique. Si aucune surveillance d'événement n'a été demandée ou si aucun compte rendu (CallInformationReport et ApplyChargingReport) n'a été demandé dans une opération envoyée antérieurement, un automate SCSM-FSM passe à l'état "Idle". Dans le cas contraire, si une surveillance d'événement a été demandée ou si un compte rendu quelconque (CallInformationReport et ApplyChargingReport) a été demandé, l'automate SCSM-FSM passe à l'état "Attente de la notification ou du compte rendu". Quand l'opération "Connect" est utilisée dans le contexte de la procédure de transfert de tâche, l'automate SCSM-FSM passe à l'état "Idle". Cependant, dans ce cas, l'entité SCF doit conserver suffisamment d'informations pour assurer la corrélation entre l'opération subséquente "AssistRequestInstructions" (à partir de l'entité SSF ou SRF assistante) et l'instance SLPI existante.

3.3.16.2.2 Traitement des erreurs

Si des messages de rejet ou d'erreur sont reçus, l'entité SCSM informe l'instance SLPI et reste dans l'état "Préparer les instructions SSF".

Le traitement générique des erreurs relatives aux opérations est décrit au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.16.3 Entité de réponse (SSF)

3.3.16.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) un traitement d'appel de base a été interrompu au point de détection;
- 3) l'entité SSF attend des instructions.

Postconditions de l'entité SSF:

- 1) l'entité SSF exécute les actions de traitement pour acheminer l'appel vers la destination spécifiée;
- 2) dans l'automate BCSM de départ, quand seules les informations d'adresse sont incluses dans l'opération "Connect", le traitement d'appel reprend au point de traitement d'appel PIC 3;
- 3) dans l'automate BCSM de départ, quand les informations d'adresse et les informations d'acheminement sont incluses dans l'opération "Connect", le traitement d'appel reprend au point de traitement d'appel PIC 4.

A la réception de cette opération dans l'état SSF-FSM "Attente des instructions", l'entité SSP exécute les actions suivantes:

- l'entité SSF annule T_{SSF} ;
- si l'opération "cutAndPaste" (couper-coller) est présente, alors l'entité SSF supprime ("coupe"), à partir du numéro de réseau intelligent composé, le nombre indiqué de chiffres et colle les chiffres composés restants à la fin du paramètre "destinationRoutingAddress" (Adresse d'acheminement de destination) présenté par l'entité SCF. Le numéro d'annuaire qui en résulte est utilisé pour l'acheminement afin d'achever la procédure d'appel effectuée;
- si l'opération "cutAndPaste" n'est pas présente, alors le paramètre "destinationRoutingAddress" présenté par l'entité SCF est utilisé pour achever la procédure d'appel effectuée. Il convient de noter que dans le cas du transfert de tâche, cela aboutit à l'acheminement vers une entité SSP assistante ou vers un périphérique intelligent assistant;
- si le numéro "callingPartyNumber" est fourni, cette valeur peut être utilisée pour le traitement subséquent de l'entité SSF;
- si aucun point EDP n'a été armé et si aucune des opérations "CallInformationReport" ou "ApplyChargingReport" n'a été demandée, l'entité FSM passe à l'état "Repos" (e9). Dans l'état contraire, l'entité FSM passe à l'état "Surveillance" (e11).

Aucune activation ou désactivation implicite de points de détection ne se produit.

Un ou plusieurs compteur(s) statistique(s) n'est (ne sont) pas affecté(s).

L'opération "Connect" s'achève après que le traitement INAP de l'opération soit terminé et avant que l'entité SSP ne commence le traitement nécessaire en vue de sélectionner un circuit.

De ce fait, il est nécessaire d'armer explicitement le point EDP "d'erreur de sélection d'acheminement" avant d'envoyer l'opération "Connect" (bien qu'ils puissent être dans le même message), afin de détecter un incident de sélection d'acheminement après une opération "Connect".

3.3.16.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.17 Procédure ConnectToResource (Connecter à la ressource)

3.3.17.1 Description générale

Cette opération est utilisée pour connecter un appel provenant de l'entité SSF vers une ressource spécialisée. Après une connexion réussie vers l'entité SRF, l'interaction avec le demandeur peut avoir lieu. L'entité SSF assure le relais entre toutes les opérations en direction de l'entité SRF et toutes les réponses en provenance de l'entité SRF.

3.3.17.1.1 Paramètres

- resourceAddress (adresse de ressource):
ce paramètre identifie l'emplacement physique de l'entité SRF.
 - IPRoutingAddress:
ce paramètre indique l'adresse d'acheminement afin d'établir une connexion en direction de l'entité SRF.
 - aucun:
ce paramètre indique que le correspondant doit être connecté à une entité SRF prédéfinie.
- serviceInteractionIndicators (Indicateurs d'interaction de service):
ce paramètre contient des indicateurs envoyés de l'entité SCP vers l'entité SSP pour le contrôle des services du réseau au centre de départ et au centre d'arrivée.

3.3.17.2 Entité d'appel (SCF)

3.3.17.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) l'instance SLPI a déterminé que des informations supplémentaires étaient nécessaires en provenance du correspondant;
- 3) le correspondant n'est pas relié à un autre correspondant;
- 4) l'automate SCSM-FSM est dans l'état "Acheminement vers la ressource", sous-état "Déterminer le mode";
- 5) l'instance SLPI a déterminé que l'entité SRF pouvait être atteinte en provenance de l'entité SSF.

Postconditions de l'entité SCF:

- 1) l'entité SCSM envoie une opération du type "PlayAnnouncement" ou "PromptAndCollectUserInformation" accompagnant l'opération "ConnectToResource";
- 2) l'automate SCSM-FSM passe à l'état "Interaction avec l'utilisateur".

3.3.17.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.17.3 Entité de réponse (SSF)

3.3.17.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) le traitement d'appel de base a été interrompu au point de détection et une relation de commande a été établie;
- 2) l'automate SSF-FSM est dans l'état "Attente des instructions".

Postconditions de l'entité SSF:

- 1) l'appel est commuté vers l'entité SRF;
- 2) une relation de commande vers l'entité SRF est établie;
- 3) l'automate SSF-FSM passe à l'état "Attente de la fin de l'interaction avec l'utilisateur". Le temporisateur d'application T_{SSF} est chargé, si nécessaire.

NOTES

1 Que le T_{SSF} soit ou non utilisé dans ce cas dépend de l'exploitant de réseau. Il doit être synchronisé avec $T_{SCF-SSF}$ dans l'entité SCSM.

2 Le succès de la connexion vers l'entité SRF provoque une transition d'état dans l'automate SRF-FSM de l'état "Idle" vers l'état "Connected".

3.3.17.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.18 Continue procedure (Procédure de maintien de l'appel)

3.3.18.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de procéder au traitement de l'appel au point de détection auquel elle a précédemment interrompu le traitement de l'appel pour attendre les instructions SCF. L'entité SSF continue le traitement d'appel sans remplacer les informations nouvelles en provenance de l'entité SCF.

3.3.18.1.1 Paramètres

Aucun.

3.3.18.2 Entité d'appel (SCF)

3.3.18.2.1 Procédure normale

Précondition de l'entité SCF:

- l'entité SCSM est dans l'état "Préparer les instructions SSF".

Postcondition de l'entité SCF:

- l'entité SCSM est dans l'état "Attente de la notification ou de la demande", dans le cas où la surveillance serait nécessaire, ou dans l'état "Idle" au cas où aucune surveillance ne serait demandée.

L'entité SCSM est dans l'état "Préparer les instructions SSF". L'opération "Maintien de l'appel" est invoquée par une instance SLPI. Cela entraîne un passage de l'entité SCSM vers l'état "Idle" si aucune surveillance n'est nécessaire. Cependant, si une surveillance est nécessaire comme dans le cas de points EDP armés ou de demandes de comptes rendus en cours d'exécution, l'entité SCSM passe à l'état "Attente de la notification ou du compte rendu".

3.3.18.2.2 Traitement des erreurs

Le traitement des erreurs d'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.18.3 Entité de réponse (SSF)

3.3.18.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) l'automate BCSM: le traitement d'appel de base a été interrompu à n'importe quel point de détection;
- 2) l'automate SSF-FSM est dans l'état "Attente des instructions".

Postconditions de l'entité SSF:

- 1) l'automate BCSM: le traitement d'appel normal se poursuit;
- 2) l'automate SSF-FSM est dans l'état "Surveillance" parce que au moins un point EDP a été armé, ou parce que aucune opération du type "CallInformationReport" ou "ApplyChargingReport" n'a été demandée; ou
l'automate SSF-FSM est dans l'état "Idle" parce que aucun point EDP n'a été armé et parce que aucune opération du type "CallInformationReport" ou "ApplyChargingReport" n'a été demandée.

L'automate SSF-FSM est dans l'état "Attente des instructions". L'organe de commande du modèle SSME reçoit l'opération "Continue" et la relaie vers l'automate SSF-FSM approprié. L'automate SSF-FSM passe à l'état "Idle" au cas où aucun EDP n'est armé et où il n'existe pas de compte rendu en cours. L'automate SSF-FSM passe à l'état "Surveillance" si au moins un point EDP est armé, ou s'il y a au moins une demande de compte rendu en cours d'exécution. Le traitement d'appel de base est repris.

3.3.18.3.2 Traitement des erreurs

Le traitement des erreurs relatives aux opérations n'est pas applicable, en raison de l'opération de classe 4.

3.3.19 Procédure DisconnectForwardConnection (Déconnecter la connexion vers l'avant)

3.3.19.1 Description générale

Cette opération est utilisée dans les deux cas suivants:

- 1) libérer une connexion vers une entité SRF:
l'opération est utilisée pour déconnecter explicitement une connexion vers une ressource (SRF) qui a été préalablement effectuée par une opération "ConnectToResource" ou "EstablishTemporaryConnection". Elle est utilisée pour une déconnexion vers l'avant à partir de l'entité SSF. Une autre solution possible est la déconnexion vers l'arrière à partir de l'entité SRF, contrôlée par le paramètre "DisconnectFromIPForbidden" dans les opérations "PlayAnnouncement" et "PromptAndCollectUserInformation";
- 2) libérer une connexion vers une entité SSF assistante:
cette opération est envoyée vers l'entité SSF non assistante de deux entités SSF impliquées dans une procédure d'assistance. Elle est utilisée pour déconnecter la connexion temporaire entre l'entité SSF de déclenchement et l'entité SSF assistante, et l'entité SSF assistante et son entité SRF connexe.

3.3.19.1.1 Paramètres

Aucun.

3.3.19.2 Entité d'appel (SCF)

3.3.19.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) une procédure d'assistance ou de relais est en cours;
- 3) une instance SLPI a déterminé qu'une opération "DisconnectForwardConnection" devait être envoyée par l'entité SCF.

Postcondition de l'entité SCF:

- l'exécution de l'instance SLPI peut continuer.

L'opération "DisconnectForwardConnection" est utilisée pour donner l'ordre à l'entité SSF de déconnecter la connexion vers l'avant vers l'entité SSF assistante ou l'entité physique contenant l'entité SRF.

Dans l'état "Interaction de l'utilisateur" de l'automate SCSM-FSM, sous-état "Attente de la réponse en provenance de l'entité SRF", cette opération est invoquée par l'entité SCF quand la logique de traitement de service détermine qu'une interaction avec l'utilisateur est terminée et demande à l'entité SSF de déconnecter la connexion temporaire vers l'entité SSF ou l'entité SRF assistante. L'automate SCSM-FSM passe à l'état "Préparer les instructions SSF".

3.3.19.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.19.3 Entité de réponse (SSF)

3.3.19.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel au départ a eu lieu;
- 2) un traitement d'appel de base a été interrompu au point de détection;
- 3) l'entité SSF de déclenchement est dans l'état "Attente de la fin de l'interaction avec l'utilisateur" ou "Attente de la fin de la connexion temporaire".

Postconditions de l'entité SSF:

- 1) la connexion vers l'entité SRF ou l'entité SSF assistante est libérée;
- 2) l'entité SSF attend des instructions.

La réception des résultats de l'opération "DisconnectForwardConnection" (déconnecter la connexion vers l'avant) provoque la déconnexion de l'entité SSF assistante ou de l'entité physique contenant l'entité SRF à partir de l'appel considéré. Elle ne libère pas la connexion de l'entité SSF pour la renvoyer vers l'utilisateur.

Cette opération est acceptée dans l'état "Attente de la connexion temporaire" ou "Attente de la fin de l'interaction avec l'utilisateur" de l'automate SSF-FSM. A la réception de cette opération dans ces états, l'entité SSF doit exécuter les actions suivantes:

- l'entité SSF de déclenchement libère la connexion vers l'entité SSF assistante ou l'entité SRF relais;
- l'entité SSF remet à zéro le temporisateur d'application T_{SSF} ;
- l'automate SSF-FSM passe à l'état "Attente des instructions" (e8).

NOTE – Le succès de la déconnexion vers l'entité SRF provoque une transition d'état dans l'entité SRF-FSM vers l'état "Repos". Une instruction en cours ("PlayAnnouncement" ou "PromptAndCollectUserInformation") est annulée et toute instruction de file d'attente ("PlayAnnouncement" ou "PromptAndCollectUserInformation") est supprimée.

3.3.19.3.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.20 Procédure "EstablishTemporaryConnection" (Etablissement d'une connexion temporaire)

3.3.20.1 Description générale

Cette opération est utilisée pour établir une connexion entre une entité SSF de déclenchement et une entité SSF assistante faisant partie d'une procédure d'assistance de service. Elle peut également être utilisée pour établir une connexion entre une entité SSF et une entité SRF, dans le cas où l'entité SRF existerait dans une entité physique adressable séparément.

3.3.20.1.1 Paramètres

- `assistingSSPIPRoutingAddress` (Adresse d'acheminement du périphérique intelligent SSPIP assistant):
ce paramètre indique l'adresse de destination de l'entité SRF pour la procédure d'assistance. L'adresse "`assistingSSPIPRoutingAddress`" peut contenir un identificateur de corrélation et un identificateur `scfId` qui sont intégrés dans cette adresse, mais seulement si ces deux identificateurs ne sont pas spécifiés séparément.
- `correlationID` (Identification de corrélation):
ce paramètre est utilisé par l'entité SCF pour associer l'opération "AssistRequestInstructions" à partir de l'entité SSF existante (ou l'entité SRF assistante) avec la demande de l'entité SSF de déclenchement. L'identificateur "`correlationID`" n'est utilisé que si l'identificateur de corrélation n'est pas intégré dans "`assistingSSPIPRoutingAddress`". L'exploitant de réseau doit décider des correspondances exactes de ce paramètre sur le système de signalisation utilisé.

- scfID:
voir la Recommandation Q.1290. L'identificateur "scfID" est utilisé uniquement si l'entité SCF n'est pas intégrée dans "assistingSSPIPRoutingAddress". L'exploitant de réseau doit décider des correspondances exactes de ce paramètre sur le système de signalisation utilisé.
- carrier (exploitant):
voir la Recommandation Q.1290. Dans ce message, le champ de sélection de l'exploitant est nul (00000000) et l'indicateur d'exploitant indique l'exploitant devant être utilisé pour l'appel.
- serviceInteractionIndicators (Indicateurs d'interaction de service):
ce paramètre contient des indicateurs envoyés de l'entité SCP vers l'entité SSP pour le contrôle des services de réseau dans le centre de départ et dans le centre de destination.

3.3.20.2 Entité d'appel (SCF)

3.3.20.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) la logique de traitement de service a déterminé qu'une connexion était nécessaire entre les entités SSF et SRF ou entre l'entité SSF et une entité SSF assistante;
- 3) le correspondant n'est pas connecté à un autre correspondant.

Postcondition de l'entité SCF:

- l'entité SCF attend l'opération "Attente des instructions de demande d'assistance".

Dans l'état "Acheminement vers la ressource" de l'automate SCSM-FSM, cette opération est invoquée par l'entité SCF quand la logique de traitement de service détermine qu'une "relation directe SCF-SRF" est nécessaire. L'automate SCSM-FSM passe ensuite à l'état "Attente des instructions de demande d'assistance".

3.3.20.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.20.3 Entité de réponse (SSF)

3.3.20.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) le traitement d'appel de base a été interrompu au point de détection 1, 2, 3, 4, 5, 6, 7, 12, 13, 14 ou 15;
- 3) l'entité SSF attend les instructions;
- 4) l'entité SSF n'est pas une entité SSF assistante.

Postconditions de l'entité SSF:

- 1) l'entité SSF exécute les actions de traitement d'appel pour acheminer l'appel vers l'entité SSF ou SRF assistante selon l'opération "assistingSSPIPRoutingAddress" demandée par l'entité SCF;
- 2) l'entité SSF attend la fin de la connexion temporaire.

A la réception de cette opération dans l'automate SSF-FSM "Attente des instructions", l'entité SSP doit exécuter les actions suivantes:

- réinitialiser le temporisateur T_{SSF} ;
NOTE – Ce terme "facultatif" signifie que le temporisateur d'application T_{SSF} est initialisé facultativement. Qu'il soit ou non utilisé dépend de l'exploitant de réseau. Mais, il doit être synchronisé avec $T_{SCF-SSF}$ dans l'entité SCSM.
- acheminer l'appel vers l'entité SSF ou SRF assistante en utilisant "assistingSSPIPRoutingAddress";
- l'automate SSF-FSM passe à l'état "Attente de la fin de la connexion temporaire" (e7).

3.3.20.3.2 Traitement des erreurs

Jusqu'à ce que l'établissement de la connexion ait été accepté (se référer à la Recommandation Q.71) par l'entité SSF/SRF assistante, toutes les indications d'incident reçues en provenance du réseau sur l'établissement ETC seront indiquées à l'entité SCF comme erreur ETC "ETCFailed" (par exemple occupé, encombrement). Il convient de noter que le temporisateur pour ETC a une durée d'application plus longue que le temps maximal autorisé pour les procédures de signalisation destinées à accepter la connexion.

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.21 Procédure EventNotificationCharging (chargement de notification d'événement)

3.3.21.1 Description générale

Cette opération est utilisée par l'entité SSF pour rendre compte à l'entité SCF de l'occurrence d'un type particulier d'événement de taxation comme le demande l'entité SCF en utilisant l'opération "RequestNotificationChargingEvent". Cette opération fait appel aux options afin de prendre en charge les interactions applicables à la taxation. (Il convient de se référer à l'Appendice II.4/Q.1214 "Scénarios de taxation".)

Etant donné que plusieurs événements de taxation peuvent se produire pendant une configuration de connexion, il est possible d'invoquer l'opération "EventNotificationCharging" plusieurs fois. Pour chaque configuration de connexion, une opération "EventNotificationCharging" peut être utilisée plusieurs fois.

3.3.21.1.1 Paramètres

- eventTypeCharging (Type d'événement de taxation):
ce paramètre indique le type d'événement de taxation qui s'est produit. Son contenu dépend de l'exploitant de réseau, sous la forme d'impulsions de taxation ou de messages de taxation.
- eventSpecificInformationCharging:
ce paramètre contient des informations relatives à la taxation particulière. Son contenu est spécifique à l'exploitant de réseau.
- legID (Elément d'identification du demi-appel):
cet élément indique l'identification du demi-appel auquel s'applique l'événement de taxation.
- monitorMode (Mode de surveillance):
ce paramètre indique comment s'effectue le compte rendu de l'événement de taxation. Quand le mode "monitorMode" est interrompu, l'événement est déclaré comme une demande. Si le mode "monitorMode" est "notifyAndContinue", l'événement est déclaré comme une notification. Le mode de surveillance "transparent" ne s'applique pas à l'opération "EventNotificationCharging".

3.3.21.2 Entité d'appel (SSF)

3.3.21.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une relation de commande existe entre les entités SCF et SSF;
- 2) un événement de taxation qui est demandé par l'entité SCF a été détecté.

Postcondition de l'entité SSF:

- aucune transition d'état de l'entité FSM.

L'automate SSF-FSM est dans un état quelconque, à l'exception de l'état "Repos". Cette opération est appelée si un événement de taxation qui a été demandé par l'entité SCF a été détecté. L'événement de taxation détecté peut être provoqué soit:

- a) par une autre instance de logique de traitement de service SLPI; ou
- b) par un autre autocommutateur.

Quelle que soit la cause de l'événement de taxation, l'entité SSF effectue une des actions suivantes lorsque se produit l'événement de taxation (selon le mode "monitorMode" correspondant):

Interrupted (Interrompu):

- Il convient de notifier à l'entité SCF l'événement de taxation en utilisant l'opération "EventNotificationCharging": ne pas traiter l'événement, mais l'éliminer. Cependant, le traitement de l'appel et de la taxation en cours ne sera pas interrompu dans l'entité SSF.

NotifyAndContinue (Notifier et continuer):

- Il convient de notifier à l'entité SCF l'événement de taxation en utilisant l'opération "EventNotificationCharging" et de continuer le traitement de l'événement ou de la signalisation.

3.3.21.2.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.21.3 Entité de réponse (SCF)

3.3.21.3.1 Procédure normale

Précondition de l'entité SCF:

- une opération "RequestNotificationChargingEvent" a été envoyée à la demande d'un programme SLPI et le programme SLPI prévoit une opération "EventNotificationCharging" en provenance de l'entité SSF.

Postcondition de l'entité SCF:

- aucune transition d'état de l'entité FSM.

A la réception de cette opération, le programme SLPI qui attend cette notification peut continuer. Si le mode de surveillance correspondant a été fixé par le programme SLPI jusqu'à "Interrupted", le programme SLPI prépare des instructions pour l'entité SSF, le cas échéant.

3.3.21.3.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.22 Procédure EventReportBCSM

3.3.22.1 Description générale

Cette opération est utilisée pour notifier à l'entité SCF un événement relatif à un appel qui a été préalablement demandé par l'entité SCF dans une opération "RequestReportBCSMEvent". La surveillance de plus d'un événement pourrait être demandée par une opération "RequestReportBCSMEvent", mais chacun de ces événements demandés est indiqué de façon distincte dans une opération "EventReportBCSM".

3.3.22.1.1 Paramètres

- eventTypeBCSM (Type d'événement du modèle BCSM):
ce paramètre spécifie le type d'événement qui est indiqué.
- bcsmEventCorrelationID:
identificateur utilisé par l'entité SCF pour la corrélation avec une opération antérieure.
- eventSpecificInformationBCSM (Information spécifique à l'événement du modèle BCSM):
ce paramètre indique les informations relatives à l'appel qui sont spécifiques à l'événement.
Pour "CollectedInfo", ce paramètre contiendra le numéro "calledPartyNumber".
Pour "AnalysedInfo", ce paramètre contiendra le numéro "calledPartyNumber".
Pour "RouteSelectFailure", ce paramètre contiendra la cause d'incident "FailureCause" (Cause d'incident), le cas échéant.
Pour O- ou T-Busy (Occupation du demandé au départ ou à l'arrivée), il contiendra la cause d'occupation "BusyCause", le cas échéant.
Pour O- ou T-NoAnswer (Non-réponse au départ ou à l'arrivée), il sera vide.
Pour O- ou T-Answer (Réponse au départ ou à l'arrivée), il sera vide.
Pour O- ou T-MidCall, il contiendra "connectTime", le cas échéant.
Pour O- ou T-Disconnect, il contiendra la "releaseCause" et/ou "connectTime" (temps de connexion), le cas échéant.
Le temps de connexion, le cas échéant, indique la durée entre l'indication de réponse reçue en provenance du demandé et la libération de la connexion en unités de 100 ms.

- **legID** (élément d'identification de demi-appel):
ce paramètre indique pour quel correspondant dans l'appel l'événement est indiqué. L'entité SSF utilisera l'option "ReceivingSideID" (identificateur de la partie réceptrice) uniquement.
 - **receivingSideID** (identificateur de la partie réceptrice):
on suppose les valeurs suivantes pour "legID" (élément d'identification du demi-appel):
 - i) "legID" = 1 indique le correspondant qui était présent au moment de l'opération "Point de détection initial" ou de l'opération spécifique au point de détection (en cas de déclenchement de demi-appel, le correspondant provoquant le déclenchement), ou l'entité correspondante qui a été créée par l'opération "InitiateCallAttempt".
 - ii) "legID" = 2 indique que l'entité correspondante a été créée par une opération "Connect" (Connexion), ou en cas de déclenchement de demi-appel, le correspondant ne provoquant pas le déclenchement.
Si l'indicateur n'est pas inclus, on prendra les cas suivants par défaut:
 - iii) legID = 1 pour les événements CollectedInfo, AnalysedInformation, O-Abandon et T-Abandon.
 - iv) legID = 2 pour les événements RouteSelectFailure, O-CalledPartyBusy, O-NoAnswer, O-Answer, T-Busy, T-NoAnswer et T-Answer.
Le paramètre "legID" devra toujours être inclus pour les événements O-MidCall, O-Disconnect, T-MidCall et T-Disconnect.
- **miscCallInfo** (Informations diverses sur l'appel):
ce paramètre indique les informations relatives au point de détection.
 - **messageType**:
ce paramètre indique si le message est une demande, c'est-à-dire s'il résulte d'un événement "RequestReportBCSMEvent" avec "monitorMode" = interrompu, ou d'une notification, c'est-à-dire s'il provient d'un événement "RequestReportBCSMEvent" avec "monitorMode" = "notifyAndContinue".
 - **attribution du point de détection**:
ce paramètre sera omis pour cette opération.

3.3.22.2 Entité d'appel (SSF)

3.3.22.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) l'automate SSF-FSM est dans l'état "Surveillance"; ou
l'automate SSF-FSM peut être dans l'état "Attente des instructions" si le point de détection O/TDisconnect est armé et satisfait; ou
l'automate SSF-FSM peut être dans n'importe quel état si le point de détection O/TAbandon est armé et satisfait;
- 2) le modèle BCSM s'oriente vers un point EDP qui est armé.

Postconditions de l'entité SSF:

- 1) l'automate SSF-FSM reste dans l'état "Surveillance" si le type de message était la notification et s'il y a d'autres points EDP armés ou si une opération du type "CallInformationReport" ou "ApplyChargingReport" est demandée;
- 2) l'automate SSF-FSM passe à l'état "Repos", si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée;
- 3) l'automate SSF-FSM passe à l'état "Attente des instructions" si le type de message était la demande. Le traitement d'appel est interrompu.

Pour certains éléments de service, il est nécessaire que la même instance du modèle O-BCSM soit réutilisée. Des exemples sont des reprises de numérotation.

La décision de réutiliser la même instance du modèle O-BCSM peut uniquement être prise par l'entité SCF après que certains points EDP-R armés aient été signalés.

Les points EDP-R considérés sont:

- RouteSelectFailure (Echec de sélection d'acheminement);
- O_CalledPartyBusy (Occupation du demandé au départ);
- O_NoAnswer (Non-réponse au départ);
- O_Disconnect (Déconnexion au départ).

Si les critères d'un point EDP-R sont satisfaits, ce qui provoque la libération du demi-appel, tous les points EDP se rapportant à ce demi-appel sont désarmés et l'événement est signalé par l'intermédiaire d'une opération "EventReportBCSM". Pour permettre de réutiliser la même instance du modèle O-BCSM, le modèle BCSM doit conserver l'enregistrement de tous les paramètres de signalisation relatifs à l'appel (par exemple "callingPartyNumber", "callingPartyCategory") jusqu'à ce que l'instance du modèle BCSM soit libérée.

Dans le cas où le point EDP-R respectif est satisfait (voir la liste ci-dessus "O-Disconnect" uniquement pour l'identificateur du demi-appel = 2), le demi-appel A sera maintenu, le demi-appel B sera libéré et l'entité SCF est informée par l'intermédiaire de "EventReportBCSM".

Si le correspondant A interrompt l'appel, l'appel est libéré qu'un point de détection ait été ou non armé.

3.3.22.2.2 Traitement des erreurs

Dans le cas où le type de message est la demande, à la fin de la temporisation T_{SSF} et avant de recevoir une quelconque opération, l'entité SSF interrompt l'interaction avec l'entité SCF et la fin de traitement d'appel est effectuée, par exemple par une annonce terminale.

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.22.3 Entité de réponse (SCF)

3.3.22.3.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre les entités SSF et SCF;
- 2) l'automate SCSM-FSM est dans l'état "Préparer les instructions SSF", sous-état "Attente de la notification ou de la demande".

Postconditions de l'entité SCF:

- 1) l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande", si le type de message était la notification et s'il n'y a plus d'autres points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM passe à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM passe à l'état "Préparer les instructions SSF" si le type de message était la demande;
- 2) l'événement est indiqué à un programme SLPI, en tenant compte de l'identificateur de dialogue. L'entité SCF préparera les instructions SSF ou SRF selon l'instance SLPI.

3.3.22.3.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.23 Procédure "FurnishChargingInformation" (Fournir les informations de taxation)

3.3.23.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de produire, d'enregistrer un graphique d'appel ou d'inclure certaines informations dans le graphique d'appel par défaut. Le graphique des appels enregistrés est prévu pour la taxation (en différé) de l'appel. Il est possible que l'opération FurnishChargingInformation (FCI) soit invoquée plusieurs fois. Une opération FCI pourrait être appliquée au début de l'appel pour demander qu'on commence à produire les graphiques d'appel. En outre, l'opération FCI peut également être appliquée à la fin de l'appel ou de la configuration de connexion (par exemple pour des reprises de numérotation). Dans ce cas, l'opération FCI est également appliquée pour inclure les informations relatives à la taxation dans le graphique des appels qui avait démarré au début de l'appel. Quand

d'autres opérations FCI sont utilisées, il est recommandé d'armer un point EDP-R (indiquant la fin de l'appel ou de la configuration de connexion) pour pouvoir appliquer l'opération FCI avant la fin de la production du graphique d'appel. Les scénarios de taxation sur lesquels repose cette opération sont: 2.2, 2.3 et 2.4 (se référer à l'Appendice II/Q.1214 "Scénarios de taxation").

3.3.23.1.1 Paramètres

- FCIBillingChargingCharacteristics (Caractéristiques de facturation/taxation de l'opération "fournir les informations de taxation"):

ce paramètre indique les caractéristiques de facturation et/ou de taxation. Son contenu est spécifique à l'exploitant de réseau. Selon le scénario de facturation appliqué, les éléments d'information suivants peuvent être inclus (il convient de se référer à l'Appendice II/Q.1214 "Scénarios de taxation"):

- complete charging record (effectuer le graphique d'enregistrement) (scénario 2.2);
- charge party (correspondant taxé) (scénario 2.3);
- charge level (niveau de taxation) (scénario 2.3);
- charge items (éléments de taxation) (scénario 2.3);
- correlationID (identificateur de corrélation) (scénario 2.4).

3.3.23.2 Entité d'appel (SCF)

3.3.23.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre l'entité SCF et l'entité SSF;
- 2) une instance SLPI a déterminé qu'une opération "FurnishChargingInformation" doit être envoyée par l'entité SCF.

Postconditions de l'entité SCF:

- 1) aucune transition d'état de l'entité FSM;
- 2) l'exécution par l'instance SLPI peut se poursuivre.

L'automate SCSM-FSM est dans l'état "Préparer les instructions SSF" ou dans l'état "Mise en file d'attente dans l'entité FSM". Cette opération est invoquée par l'entité SCF si une instance SLPI aboutit à une demande de création d'un graphique d'appel vers l'entité SSF ou afin d'inclure certaines informations de facturation ou de taxation dans le graphique des éléments d'appel par défaut. Dans le cas des appels mis en file d'attente, cette opération peut contenir des informations relatives au début des appels mis en file d'attente ou de la durée de mise en file d'attente des appels à des fins d'enregistrement des événements d'appel. Cela ne provoque pas de transition d'état dans l'automate SCSM-FSM.

3.3.23.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.23.3 Entité de réponse (SSF)

3.3.23.3.1 Procédure normale

Préconditions de l'entité SSF:

- automate SSF-FSM: état c, "Attente des instructions"; ou
automate SSF-FSM: état d, "Attente de la fin de l'interaction avec l'utilisateur"; ou
automate SSF-FSM: état e, "Attente de la fin de la connexion temporaire"; ou
automate SSF-FSM: état f, "Surveillance"; ou
automate SSF-FSM d'assistance/de transfert de tâche, état b, "Attente des instructions".

Postcondition de l'entité SSF:

- aucune transition d'état de l'entité FSM.

A la réception de cette opération, l'entité SSF effectue des actions pour créer un graphique d'appel selon le scénario de taxation différée qui est applicable en utilisant les éléments d'information qui sont inclus dans l'opération:

- enregistre le graphique complet de l'appel qui est inclus dans l'opération;
- crée et enregistre un graphique d'appel selon les informations (correspondant à la taxation, au niveau de taxation, aux éléments de taxation);
- inclut les informations reçues "identification de corrélation" dans le registre d'appel par défaut qui est produit et enregistré par défaut dans l'entité SSF.

Au moyen d'un paramètre dans l'opération "FurnishChargingInformation", l'entité SCF peut déclencher la fonction de comptage de la "taxation par impulsions" de l'entité SSF.

Dans ce cas, l'entité SSF produira des impulsions de taxation selon le niveau de taxation applicable, les comptabilisera et les enregistrera.

L'entité SSF enregistre les données relatives à la taxation comme par exemple la durée d'appel, l'heure de début ou l'heure de la fin d'appel. En outre, l'entité SSF enregistre d'autres données, si cela est nécessaire.

Le niveau de taxation peut être déterminé par:

- a) l'entité SCF; ou
- b) l'entité SSF; ou
- c) un autocommutateur suivant; ou
- d) la fonction de post-traitement.

Si a) est applicable, le niveau de taxation est inclus dans l'opération "FurnishChargingInformation".

Si b) est applicable, l'entité SSF devra déterminer le niveau de taxation à partir des paramètres correspondants contenus dans l'opération.

Si c) est applicable, soit l'opération "FurnishChargingInformation" contient les paramètres correspondants indiquant que le niveau de taxation doit être déterminé dans un autocommutateur suivant, soit l'entité SSF détecte durant la détermination du niveau de taxation à partir des paramètres fournis que le niveau de taxation doit être déterminé dans un autocommutateur suivant.

L'entité SSF peut soit comptabiliser les impulsions reçues soit convertir en impulsions tous les messages de taxation reçus en provenance de l'extrémité B. Dans les deux cas, le cumul des impulsions est inclus quand le graphique d'appel RI est préparé ou est omis.

3.3.23.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.24 Procédure HoldCallInNetwork (Mise en garde du réseau)

3.3.24.1 Description générale

Cette opération est utilisée pour permettre la mise d'un appel en file d'attente pendant la phase d'établissement (par exemple exécuter la procédure d'appel sur occupation, l'appel étant mis en file d'attente jusqu'à ce que le destinataire devienne libre).

3.3.24.1.1 Paramètres

- holdcause ("Cause de maintien de l'appel"):
ce paramètre est facultatif. Quand il est utilisé, il est spécifique au réseau et indique la cause de maintien de l'appel.

3.3.24.2 Entité d'appel (SCF)

3.3.24.2.1 Procédure normale

Précondition de l'entité SCF:

- l'entité SCSM est dans l'état 2.2.1 "Préparer les instructions SSF", et l'événement (e2.2.3) Busy_Line/Trunk a eu lieu.

Postcondition de l'entité SCF:

- l'entité SCSM va dans l'état 2.2.2 "Mise en file d'attente".

3.3.24.2.2 Traitement des erreurs

Il n'y a pas de traitement spécifique des erreurs associé à cette opération. Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.24.3 Entité de réponse (SSF)

3.3.24.3.1 Procédure normale

Précondition de l'entité SSF:

- l'automate SSF-FSM est dans l'état c, "Attente des instructions".

Postcondition de l'entité SSF:

- l'automate SSF-FSM reste dans le même état.

3.3.24.3.2 Traitement des erreurs

Il n'y a pas de traitement d'erreurs spécifique à l'opération. Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.25 Procédure "InitialDP" (Point de détection initial)

3.3.25.1 Description générale

Cette opération est envoyée par l'entité SSF après détection d'un point TDP-R dans le modèle BCSM, afin de demander à l'entité SCF des instructions pour achever l'appel.

3.3.25.1.1 Paramètres

- serviceKey:
ce paramètre indique sans ambiguïté à l'entité SCF le service de réseau intelligent demandé. Il est utilisé pour indiquer l'application/au programme de logique de service approprié à l'intérieur de l'entité SCF (et non pas un adressage SCP).
- calledPartyNumber:
ce paramètre contient le numéro utilisé pour identifier le demandé dans le sens en avant, voir donc la Recommandation Q.762.
- callingPartyNumber:
voir la Recommandation Q.762. Information de signalisation du numéro du demandeur.
- callingPartysCategory:
voir la Recommandation Q.762. Information de signalisation de la catégorie de demandeur.
- cGEncountered:
voir la Recommandation Q.1290.
- iPSSPCapabilities:
voir la Recommandation Q.1290.
- iPAavailable:
voir la Recommandation Q.1290.
- locationNumber (Numéro d'emplacement):
ce paramètre est utilisé pour transmettre l'adresse de zone géographique pour les services de mobilité, voir la Recommandation Q.762. Il est utilisé quand "callingPartyNumber" ne contient pas d'informations sur l'emplacement géographique du demandeur (par exemple acheminement dépendant de l'origine quand le demandeur est un abonné mobile).

- originalCalledPartyID:
voir la Recommandation Q.762 Original Called Number signalling information. Informations de signalisation du numéro demandé initial.
- triggerType:
ce paramètre indique à l'entité SCF l'événement particulier qui a provoqué la détection d'un état valide de déclenchement.
- highlayerCompatibility (Compatibilité de couches supérieures):
ce paramètre indique le type de compatibilité de couches supérieures qui sera utilisé pour déterminer le téléservice RNIS d'un terminal connecté au RNIS. Pour le codage, le DSS 1 (Q.931) est utilisé. La compatibilité de couches supérieures peut également être transportée par le service utilisateur ISUP à l'intérieur du paramètre ATP (voir la Recommandation Q.763).
- serviceInteractionIndicators (Indications de dialogues de service):
ce paramètre contient des indications envoyées à partir de l'entité SSF vers l'entité SCF pour commander les services de réseau dans les autocommutateurs de départ et d'arrivée.
- additionalCallingPartyNumber (Numéro complémentaire du demandeur):
le numéro du demandeur fourni par le système de signalisation à l'accès du demandeur, par exemple fourni par un autocommutateur privé.
- forwardCallIndicators:
ce paramètre indique si l'appel devra être traité comme un appel national ou international. Il indique également les capacités de signalisation de l'accès au réseau, la connexion de réseau précédente et les capacités de signalisation préférentielles de la connexion de réseau suivante. Les capacités d'accès de réseau n'indiquent pas le type de terminal. Par exemple, un autocommutateur PBX de réseau privé aura un accès de type RNIS, mais le terminal de l'utilisateur derrière le PBX pourra être du type RNIS ou non-RNIS.
- bearerCapability:
ce paramètre indique le type de la connexion de la capacité du support ou les spécifications du moyen de transmission vers l'utilisateur. On choisira en tant qu'option dans le réseau l'un des deux paramètres à utiliser:
 - bearerCap:
ce paramètre contient la valeur du paramètre Capacité du Support du DSS 1 (Q.931) au cas où l'entité SSF se trouve au niveau de l'autocommutateur local, ou bien il contient la valeur du paramètre d'information du service utilisateur ISUP (Q.763) au cas où l'entité SSF se trouve au niveau du commutateur de transit.

Le paramètre "bearerCapability" (capacité du support) ne devra être inclus dans l'opération "InitialDP" que dans le cas où le paramètre Capacité Support du DSS 1 ou le paramètre d'information du service utilisateur ISUP est disponible dans l'entité SSP.

Si deux valeurs de capacité support sont disponibles dans l'entité SSF ou si l'information de service utilisateur et l'information de service utilisateur Premier choix sont disponibles dans l'entité SSF, le "bearerCap" devra contenir la valeur préférentielle de la capacité du support dans l'ordre respectif de la valeur du paramètre Premier choix de la spécification du moyen de transmission.
 - tmr (paramètre de spécification du moyen de transmission):
Le tmr est codé comme paramètre de spécification du moyen de transmission du sous-système ISUP selon la Recommandation Q.763.

Si deux valeurs relatives à la spécification du moyen de transmission sont disponibles dans l'entité SSF ou si la spécification du moyen de transmission et la spécification du moyen de transmission Premier choix sont disponibles dans l'entité SSF, le "bearerCap" devra contenir la valeur de la spécification du moyen de transmission préférentiel (la valeur du paramètre Premier choix de la spécification du moyen de transmission).
- eventTypeBCSM (Type d'événement du modèle BCSM):
ce paramètre indique l'événement de point de détection armé du modèle BCSM, résultant de l'opération "point de détection initial".

- `redirectingPartyID`:
ce paramètre indique le dernier numéro d'Annuaire à partir duquel l'appel a été réacheminé.
- `redirectionInformation`:
voir la Recommandation Q.763 Informations de signalisation du transfert d'appel.

3.3.25.2 Entité d'appel (SSF)

3.3.25.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) un événement remplissant les critères pour l'exécution du point de détection a été détecté;
- 2) l'espacement des appels et la surcharge de la signalisation sémaphore ne s'appliquent pas à l'appel, et l'appel ne doit pas être filtré.

Postconditions de l'entité SSF:

- 1) une relation de commande a été établie si le point de détection était armé comme point TDP-R, la machine FSM de l'entité SSF se place dans l'état "Attente des instructions";
- 2) la machine FSM de l'entité SSF reste dans l'état "Idle" si le point de détection était armé comme point TDP-N.

Suite à une détection de déclenchement (les critères attribués au point de détection étant satisfaits) qui s'applique à un point TDP armé dans le modèle BCSM, causée par une tentative d'appel initial, l'entité SSF vérifie si l'espacement des appels, la surcharge SS n° 7 ou le filtrage du service ne s'appliquent pas au segment d'appel considéré.

Si ces conditions sont satisfaites, l'opération "Point de détection initial" est invoquée par l'entité SSF. L'adresse de l'entité SCF vers laquelle l'opération "point de détection initial" doit être envoyée est déterminée à partir des données qui s'appliquent au déclenchement. L'entité SSF fournit autant de paramètres que possible. Dans certains cas, certains paramètres doivent être disponibles (tels que `callingPartyNumber` ou `callingPartyCategory`). Cela doit être traité de façon appropriée par l'entité SSF dans sa table de déclenchement (pour savoir que de tels paramètres sont nécessaires pour certaines conditions de déclenchement) et pour effectuer les actions nécessaires pour obtenir ces paramètres s'ils ne sont pas disponibles (par exemple si une signalisation non sémaphore est utilisée, il peut être possible de demander la `callingPartyCategory` à partir d'un autocommutateur précédent).

Dans le cas contraire, la commande d'appel est renvoyée vers le réseau sous-jacent.

Si le point de détection était armé comme point TDP-R, une relation de commande est établie vers l'entité SCF. Le temporisateur d'application T_{SSF} de l'entité SSF est chargé quand l'entité SSF envoie l'opération "Point de détection initial" pour demander des instructions en provenance de l'entité SCF. On l'utilise pour empêcher un temps d'interruption d'appel excessif.

3.3.25.2.2 Traitement des erreurs

Si l'entité SCF de destination n'est pas accessible, alors le traitement de fin d'appel est effectué.

A la fin de la temporisation T_{SSF} , avant de recevoir toute opération, l'entité SSF interrompt l'interaction avec l'entité SCF et le traitement de fin d'appel est effectué (par exemple acheminement vers une annonce de fin d'appel).

Si le demandeur interrompt l'appel après l'envoi de l'opération "Point de détection initial", l'entité SSF interrompt la relation de commande au moyen d'une interruption vers le protocole TCAP. Il convient de noter que le TCAP attendra jusqu'à ce que le premier message de réponse en provenance de l'entité SCF ait été reçu avant qu'il n'envoie une interruption vers l'entité SCF (voir également 3.2).

Le traitement générique des erreurs relatives aux opérations est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.25.3 Entité de réponse (SCF)

3.3.25.3.1 Procédure normale

Précondition de l'entité SCF:

- aucune.

Postcondition de l'entité SCF:

- une instance SLPI a été invoquée.

A la réception d'une opération "InitialDP", l'entité SCSM passe de l'état "Repos" à l'état "Préparer les instructions SSF", une relation de commande est créée avec l'entité SSF connexe. Une instance de programme de logique de traitement de service (SLPI) est appelée pour accéder à l'opération "point de détection initial" à partir du paramètre "serviceKey". Au moyen de cette relation de commande, l'entité SCF peut influencer le traitement d'appel de base selon la logique de traitement de service invoquée.

Les actions devant être exécutées dans l'instance SLPI dépendent des paramètres transmis par l'intermédiaire de cette opération et de l'instance SLPI elle-même, c'est-à-dire le service RI demandé lui-même.

3.3.25.3.2 Traitement des erreurs

Si l'opération "point de détection initial" est rejetée, l'entité SCSM reste à l'état "Repos". La fonction de maintenance est informée et aucune instance SLPI n'est invoquée.

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.26 Procédure "InitiateCallAttempt"

3.3.26.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de créer un nouvel appel vers un correspondant en utilisant les informations d'adresse fournies par l'entité SCF (par exemple appel de réveil); un point EDP-R doit être armé sur réponse et sur tous les événements d'échec d'appel, afin que l'entité SCF traite cet appel de façon appropriée quand l'un de ces événements surviendra.

3.3.26.1.1 Paramètres

- destinationRoutingAddress:
ce paramètre contient le numéro du demandé vers lequel l'appel doit être acheminé. L'adresse destinationRoutingAddress peut contenir un numéro de demandé uniquement pour cette opération. Le codage de ce paramètre est défini dans la Recommandation Q.763.
- alertingPattern (séquence d'alerte):
voir la Recommandation Q.1290. Elle ne s'applique que si la signalisation de réseau exploite ce paramètre ou si l'entité SSF est l'autocommutateur local de rattachement de l'abonné.
- iSDNAccessRelatedInformation:
transmet les mêmes informations que le paramètre de transport d'accès du sous-système utilisateur ISUP dans la Recommandation Q.762.
- travellingClassMark:
l'entité SCF utilise le paramètre travellingClassMark pour fournir des informations essentielles de sélection d'itinéraire vers l'entité SSF. L'entité SSF utilise ces informations pour coder le message ISUP IAM sortant, le codage et les correspondances de ce paramètre étant spécifiques à l'exploitant de réseau.
- serviceInteractionIndicators (Indicateurs d'interaction de service):
ce paramètre contient des indicateurs envoyés de l'entité SCF vers l'entité SSF pour commander les services de réseau vers les autocommutateurs de départ et d'arrivée.
- callingPartyNumber:
ce paramètre identifie le numéro qui doit être considéré comme le demandeur pour l'appel créé. Si ce paramètre n'est pas envoyé par l'entité SCF, l'entité SSF peut fournir une valeur par défaut attribuée selon le réseau.

3.3.26.2 Entité d'appel (SCF)

3.3.26.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une instance SLPI a été invoquée, aucune relation de commande n'existe entre les entités SCF et SSF;
- 2) une instance SLPI a déterminé qu'une opération "InitiateCallAttempt" devrait être envoyée vers l'entité SCF;
- 3) l'automate SCSM-FSM est dans l'état "Idle".

Postconditions de l'entité SCF:

- 1) une relation de commande est établie entre l'entité SCF et l'entité SSF;
- 2) l'automate SCSM-FSM est dans l'état "Préparer les instructions SSF";
- 3) l'exécution de l'instance SLPI se poursuit.

L'automate SCSM-FSM va à l'état "Préparer les instructions SSF" quand la logique de traitement de service appelle cette opération. Afin de permettre l'établissement d'une relation de commande entre l'entité SCF et l'entité SSF et de permettre à l'entité SCF de contrôler de façon appropriée l'appel établi, l'instance SLPI doit surveiller l'événement ou les événements du modèle BCSM indiquant le résultat de l'établissement de l'appel. Cela comprend l'un des points de détection 3 ou 4, 5, 6 et 7. Toutes les autres instructions Non-Call_Processing_Instructions peuvent être également envoyées. L'opération "InitiateCallAttempt" crée une instance de modèle BCSM dans l'entité SSF mais l'entité SSF interrompt le traitement d'appel de ce modèle BCSM. L'instance SLPI devra envoyer une opération "Continue" pour demander à l'entité SSF d'acheminer l'appel vers la destination spécifiée. L'automate SCSM-FSM devra procéder comme cela est spécifié dans la procédure de l'opération "Continue".

La procédure décrite ci-dessus fera partie intégrante de l'établissement de la relation de commande, c'est-à-dire que les opérations "jusqu'à" et incluant l'opération "Continue" devront être envoyées ensemble dans le même message à l'entité SSF. L'entité SCF devra faire démarrer un temporisateur de réponse T_{SCF} quand l'opération "InitiateCallAttempt" est envoyée. Le temporisateur de réponse devra superviser la confirmation du dialogue en provenance de l'entité SSF, la valeur de T_{SCF} devant être égale ou inférieure à celle du temporisateur de non-réponse du réseau.

3.3.26.2.2 Traitement des erreurs

A la fin de la temporisation T_{SCF} , l'entité SCF devra interrompre le dialogue, indiquer l'interruption aux fonctions de maintenance et informer l'instance de logique de traitement de service (SLPI) de l'échec de l'établissement du dialogue. L'automate SCSM-FSM va à l'état "Idle".

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2, les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.26.3 Entité de réponse (SSF)

3.3.26.3.1 Procédure normale

Précondition de l'entité SSF:

- aucune.

Postconditions de l'entité SSF:

- 1) un nouveau modèle BCSM au départ a été créé, le traitement d'appel étant suspendu au point de détection DP 1;
- 2) l'automate SSF-FSM est passé de l'état "Idle" à l'état "Attente des instructions".

A la réception de "InitiateCallAttempt", l'entité SSF crée un nouvel automate BCSM au départ et interrompt le traitement d'appel de ce modèle BCSM au point de détection DP 1. Toutes les opérations subséquentes sont traitées selon leurs procédures normales.

Les propriétés et les capacités, normalement reçues du demandeur ou associées à celui-ci, qui sont nécessaires à l'établissement de l'appel devront avoir une valeur par défaut attribuée selon le réseau. Si un numéro de demandeur est fourni par l'entité SCF, ces propriétés peuvent dépendre du numéro du demandeur qui a été reçu.

3.3.26.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.27 Procédure "ModifyEntry" (Modifier l'entrée)

3.3.27.1 Description générale

L'opération "ModifyEntry" (Modifier l'entrée) contenue dans X.500 est utilisée pour demander à l'entité SDF d'apporter une ou plusieurs modifications à des données de type objet. Les modifications concernent les attributs et leurs valeurs constituant l'objet. Le type de modifications qui doit être effectué est indiqué dans l'argument d'opération fourni par l'entité SCF. Les modifications ne s'appliquent pas aux valeurs des attributs utilisés pour identifier l'objet (c'est-à-dire nom de l'objet). Pour une description complète de l'opération ModifyEntry, voir 11.3/X.511.

3.3.27.1.1 Paramètres

Voir 11.3.2/X.511 et 11.3.3/X.511.

3.3.27.2 Entité d'appel (SCF)

3.3.27.2.1 Procédure normale

Précondition de l'entité SCF:

- SCSM: "Rattachement SDF" ou "Attente des demandes subséquentes".

Postcondition de l'entité SCF:

- SCSM: "Rattachement SDF".

Quand l'entité SCSM est dans l'état "Attente des demandes subséquentes" et qu'il existe un besoin de logique de traitement de service de modifier les informations vers l'entité SDF, un événement interne [(e2) Request_to_SDF] se produit. Avant que le processus d'application n'indique, avec un délimiteur (ou une fin de temporisation) que l'opération devrait être envoyée, l'entité SCSM reste dans l'état "Attente des demandes subséquentes" et l'opération n'est pas envoyée. L'opération est envoyée vers l'entité SDF dans un message contenant un argument de rattachement. L'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E5) Response_from_SDF_with_Bind ou (E4) Bind_Error] à l'opération de rattachement préalablement présentée à l'entité SDF provoque une transition de l'entité SCF à l'état "Rattachement SDF" ou à l'état "Repos". Une fois que l'entité SCSM est allée à l'état "Repos", l'opération ModifyEntry est abandonnée. Dans l'état "Rattachement SDF", la réponse à l'opération ModifyEntry [(E7) Response_from_SDF] provoque une transition de l'entité SCF vers le même état ("Rattachement SDF"). Cela peut être soit le résultat de l'opération ModifyEntry ou une erreur.

Quand le modèle SCSM est dans l'état "Rattachement SDF" et qu'il existe un besoin de logique de traitement de service de modifier les informations vers l'entité SDF, un événement interne se produit. Cet événement, appelé (e6) Request_to_SDF provoque une transition vers le même état "Rattachement SDF" et l'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E7) Response_from_SDF] à l'opération ModifyEntry préalablement présentée à l'entité SDF provoque une transition de l'entité SCF vers le même état "Rattachement SDF". La réponse en provenance de l'entité SDF peut être soit le résultat de l'opération ModifyEntry ou une erreur.

3.3.27.2.2 Traitement des erreurs

Le traitement générique des erreurs d'opération est décrit en 11.3.4/X.511 et 11.3.5/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 2.2.2.

3.3.27.3 Entité de réponse (SDF)

3.3.27.3.1 Procédure normale

Précondition de l'entité SDF:

- SDSM: "Rattachement SDF" ou "Rattachement en cours".

Postcondition de l'entité SDF:

- SDSM: "Rattachement SDF".

Quand l'entité SDF est dans l'état "Rattachement en cours", l'événement externe (E3) Request_from_SCF provoqué par la réception d'une opération "ModifyEntry" en provenance de l'entité SCF se produit. L'entité SDF ne traite pas l'opération avant qu'une opération "Bind" (Rattachement) ait été exécutée avec succès. Elle reste dans le même état.

Quand l'entité SCF est dans l'état "Rattachement SDF", l'événement externe (E7) Request_from_SCF provoqué par la réception d'une opération "ModifyEntry" en provenance de l'entité SCF se produit. L'entité SDF attend la réponse à l'opération.

A la réception de l'événement (E7) et avant la mise à jour des différents attributs spécifiés dans les paramètres de l'opération, l'entité SDF effectuera les actions suivantes:

- vérifier que l'objet auquel la demande permet d'accéder existe;
- vérifier que l'utilisateur au nom duquel la demande est présentée a suffisamment de droits d'accès pour modifier l'objet lors de chaque modification élémentaire contenue dans l'opération;
- vérifier que chaque attribut ou valeur pour laquelle une opération devrait être effectuée existe dans l'objet;
- vérifier que les modifications proposées sont compatibles avec la classe de l'objet ou avec la syntaxe de l'attribut.

Une fois que les actions spécifiées indiquées ci-dessus ont été effectuées avec succès pour toutes les modifications demandées dans l'opération, toutes les modifications se rapportant à un même attribut ont lieu dans l'ordre indiqué dans le paramètre "changements". Un résultat est renvoyé vers l'entité SCF. L'envoi du résultat correspond à l'événement (e6) Response_to_SCF.

3.3.27.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 11.3.4/X.511 et 11.3.5/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 2.2.2.

3.3.28 Procédure O_Answer (Réponse au départ)

3.3.28.1 Description générale

Cette opération est envoyée de l'entité SSF vers l'entité SCF au point de détection O_Answer (Réponse de départ), après avoir déclenché une condition valide de déclenchement, ou afin de rendre compte d'un événement demandé par l'opération RequestReportBCSMEvent.

3.3.28.1.1 Paramètres

- dPSpecificCommonParameters (Paramètres communs spécifiques au point de détection):
- callingPartyBusinessGroupID (Identificateur de groupe d'abonnés du demandeur):
voir la Recommandation Q.1290. L'entité SCF peut utiliser ce paramètre pour choisir des programmes de logique de traitement de service (SLP) basés sur le groupe et à des fins d'autorisation. Les exploitants de réseau peuvent spécifier que ce paramètre devrait être utilisé si leur réseau particulier a les informations disponibles.
- callingPartySubaddress (Sous-adresse du demandeur):
voir la Recommandation Q.931 "Sous-adresse du demandeur".
- callingFacilityGroup (Faisceau d'équipement du demandeur):
voir la Recommandation Q.1290.
- callingFacilityGroupMember (Elément du faisceau d'équipement du demandeur):
voir la Recommandation Q.1290.
- originalCalledPartyID (Identificateur de demandé initial):
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.
- redirectingPartyID (Identificateur de réachemineur):
ce paramètre (le cas échéant) est le numéro d'Annuaire du dernier réachemineur.
- redirectionInformation:
voir la Recommandation Q.763 Informations de signalisation de réacheminement.
- routeList (liste de réacheminement):
la liste de réacheminement représente la liste des itinéraires qui auraient été utilisés pour acheminer l'appel. Les exploitants de réseau peuvent spécifier que ce paramètre devrait être utilisé si leur réseau particulier dispose des informations nécessaires.
- travellingClassMark (Marque de classe de transport):
selon la définition de la Recommandation Q.1290.

3.3.28.2 Entité d'appel (SSF)

3.3.28.2.1 Procédure normale

Préconditions de l'entité SSF (TDP et EDP):

- 1) une tentative d'appel a eu lieu au départ;
- 2) une information a été reçue indiquant que l'appel a été accepté et le correspondant final a répondu;
- 3) l'espacement des appels ou le filtrage du service n'est pas applicable;
- 4) les critères de point de détection sont satisfaits;

- 5) pour un point TDP-R, il n'y a pas de relation de commande existante;
- 6) pour un point EDP-R, il y a une relation de commande existante et le point EDP O_Answer est armé;
- 7) pour un point EDP-N, il y a une relation de commande ou de surveillance existante et le point EDP O_Answer est armé.

Postconditions de l'entité SSF pour le point TDP:

- 1) pour un point TDP-R, le traitement d'appel de base a été interrompu au point de détection O_Answer, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel de base se poursuit au point de traitement d'appel O_Active, et aucune relation de commande n'a été établie.

Postconditions de l'entité SSF pour le point EDP:

- 1) l'automate SSF-FSM reste dans l'état "Surveillance" si le type de message était la notification et il y a d'autres points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée;
- 2) l'automate SSF-FSM passe à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée;
- 3) l'automate SSF-FSM passe à l'état "Attente des instructions" si le type de message était la demande. Le traitement de l'appel est interrompu.

3.3.28.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.28.3 Entité de réponse (SCF)

3.3.28.3.1 Procédure normale

Précondition de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours de traitement;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours de traitement.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute et met fin à l'opération.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et il y a d'autres points EDP armés ou une information de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se met à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM passe à l'état "Préparer les instructions SSF" si le type de message était la demande.

3.3.28.3.2 Traitement des erreurs

Le traitement générique des erreurs d'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.29 Procédure O_Disconnect

3.3.29.1 Description générale

Cette opération est envoyée par l'entité SSF vers l'entité SCF après avoir détecté un état de déclenchement valide au point de détection O_Disconnect, ou afin d'indiquer un événement demandé par RequestReportBCSMEvent. Il convient de se référer au 4.2.2.2/Q.1214 pour la sémantique additionnelle relative à la simulation d'appel par modèle.

3.3.29.1.1 Paramètres

- serviceAddressInformation (serviceKey, miscCallInfo, triggerType):
voir la Recommandation Q.1290. Le paramètre triggerType indique l'événement particulier qui a provoqué l'interruption de l'appel. Le paramètre miscCallInfo indique qu'une demande d'instructions a été présentée à l'entité SCF, de même que la catégorie de déclenchement (ligne, groupe, entreprise). Le paramètre serviceKey est utilisé pour identifier la logique de traitement de service devant être invoquée.
- bearerCapability:
ce paramètre indique à l'utilisateur le type de connexion de capacité support. Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage applicables au paramètre de capacité support.
- calledPartyNumber:
voir la Recommandation Q.762 pour les informations de signalisation du numéro de demandé. Ce paramètre est utilisé pour identifier le demandé dans le sens en avant.
- callingPartyNumber:
voir la Recommandation Q.762 "Information de signalisation du numéro de demandeur". Il convient de se référer au 6.4.4/Q.1214 relatif aux règles de codage du paramètre du numéro de demandeur.
- callingPartysCategory:
voir la Recommandation Q.762 "Information de signalisation de la catégorie de demandeur". Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre de catégorie de demandeur.
- iPSSPCapabilities:
voir la Recommandation Q.1290.
- iPAvailable:
voir la Recommandation Q.1290.
- iSDNAccessRelatedInformation:
ce paramètre contient des éléments d'information (pouvant être multiples) conformes à la Recommandation Q.931. Voir les informations de signalisation du paramètre Transport d'accès dans les Recommandations Q.762, Q.763 et Q.931. Il convient de se référer au 6.4.4/Q.1214 (Informations analysées) pour les règles de codage applicables aux informations relatives à l'accès RNIS.
- cGEncountered:
voir la Recommandation Q.1290.
- locationNumber:
voir la Recommandation Q.762 relative aux informations de signalisation du numéro d'emplacement. Ce paramètre est utilisé pour transporter l'adresse de zone géographique pour les services mobiles. Il est utilisé quand le "callingPartyNumber" ne contient pas d'informations sur l'emplacement géographique du demandeur (c'est-à-dire pour un acheminement dépendant de l'origine quand le demandeur est un abonné mobile).
- serviceProfileIdentifier:
voir l'Annexe A/Q.932. Voir 6.4.4/Q.1214 pour les règles de codage de serviceProfileIdentifier.
- terminalType:
voir la Recommandation Q.1290. Ce paramètre identifie le type de terminal, pour que l'entité SCF puisse spécifier à l'entité SRF le type approprié de capacité (reconnaissance de la parole, DTMF, capacité de visualisation, etc.).

- chargeNumber:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 pour les règles de codage du paramètre "chargeNumber".
- servingAreaID:
voir la Recommandation Q.1290.
- callingPartyBusinessGroupID:
voir la Recommandation Q.1290.
- callingPartySubaddress:
voir la Recommandation Q.931 "Sous-adresse du demandeur".
- callingFacilityGroup:
voir la Recommandation Q.1290.
- callingFacilityGroupMember:
voir la Recommandation Q.1290.
- releaseCause:
indique la cause de la déconnexion.
- routeList:
voir la Recommandation Q.1290.
- carrier:
voir la Recommandation Q.1290.
- connectTime:
indique la durée entre l'indication de réponse reçue en provenance du demandé et la libération de la connexion.

3.3.29.2 Entité d'appel (SSF)

3.3.29.2.1 Procédures normales

Préconditions de l'entité SSF:

- 1) une tentative d'appel au départ a eu lieu;
- 2) indication reçue en provenance du modèle BCSM terminal, selon laquelle l'appel est accepté et le correspondant final a répondu;
- 3) déconnecter l'indication reçue en provenance d'un correspondant initial, ou reçue du correspondant final par l'intermédiaire du modèle BCSM terminal;
- 4) pour un TDP, l'espacement des appels ou le filtrage du service n'est pas applicable;
- 5) les critères de point de détection ont été satisfaits;
- 6) pour un TDP-R ou un TDP-N, il n'y a pas de relation de commande existante;
- 7) pour un point EDP, il existe une relation de commande existante et le point EDP O_Disconnect est armé.

Postconditions de l'entité SSF:

- 1) pour un point TDP-R, le traitement d'appel de base a été interrompu au point de détection O_Disconnect et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel se poursuit au point de traitement O_Null & Authorize Termination Attempt, et aucune relation de commande n'a été établie;
- 3) pour un point EDP, la même procédure que pour EventReportBCSM s'applique (voir 3.3.22.2.1).

3.3.29.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.29.3 Entité de réponse (SCF)

3.3.29.3.1 Procédure normale

Précondition de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie, et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute et met fin au traitement.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" et il y a encore des points EDP armés, ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se place dans l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se place dans l'état "Préparer les instructions SSF" si le type de message était la demande.

3.3.29.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.30 Procédure O_No_Answer

3.3.30.1 Description générale

Cette opération est envoyée de l'entité SSF vers l'entité SCF au point de détection O_No_Answer, après la détection d'un état de déclenchement valide, ou pour signaler un événement demandé par RequestReportBCSMEvent.

3.3.30.1.1 Paramètres

- dPSpecificCommonParameters:
- callingPartyBusinessGroupID:
voir la Recommandation Q.1290. L'entité SCF peut utiliser ce paramètre pour choisir des programmes de logique de traitement de service (SLP) basés sur le groupe et à des fins d'autorisation. Les exploitants de réseau peuvent spécifier que ce paramètre devrait être utilisé au cas où leur réseau particulier aurait les informations disponibles.
- callingPartySubaddress:
voir la Recommandation Q.931 "Sous-adresse du demandeur".
- callingFacilityGroup (Faisceau d'équipement du demandeur):
voir la Recommandation Q.1290 "Sous-adresse du demandeur".
- callingFacilityGroupMember (Elément du faisceau d'équipement du demandeur):
voir la Recommandation Q.1290.
- originalCalledPartyID:
voir la Recommandation Q.762 Information de signalisation du numéro de demandé initial.

- préfixe:
défini dans la Recommandation Q.1290.
- redirectingPartyID:
ce paramètre, le cas échéant, est le numéro d'Annuaire du dernier réacheminneur.
- redirectionInformation:
voir la Recommandation Q.763 Informations de signalisation de réacheminement.
- routeList:
routeList représente l'itinéraire utilisé.
- travellingClassMark:
comme cela est défini dans la Recommandation Q.1290.
- exploitant:
comme cela est défini dans la Recommandation Q.1290.

3.3.30.2 Entité d'appel (SSF)

3.3.30.2.1 Procédure normale

Préconditions de l'entité SSF (TDP et EDP):

- 1) une tentative d'appel au départ a eu lieu;
- 2) indication reçue que le correspondant final n'a pas répondu dans l'intervalle de temps spécifié;
- 3) l'espacement des appels ou le filtrage du service n'est pas applicable;
- 4) les critères du point de détection ont été satisfaits (TDP ou EDP);
- 5) pour un point TDP-R, il n'y a pas de relation de commande existante;
- 6) pour un point EDP-R, il y a une relation de commande existante et le point EDP O_No_Answer est armé;
- 7) pour un point EDP-N, il y a une relation de commande ou de surveillance existante et le point EDP O_No_Answer est armé.

Postconditions de l'entité SSF pour le point TDP:

- 1) pour un point TDP-R, le traitement de base de l'appel a été interrompu au point de détection O_No_Answer, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement des valeurs par défaut a été fourni, et aucune relation de commande n'a été établie. L'utilisation du TDP-N à ce point de détection implique qu'il n'y a pas d'autre avertissement du demandé et que les traitements de non-réponse faisant partie de l'autocommutateur peuvent, le cas échéant, être invoqués.

Postconditions de l'entité SSF pour le point EDP:

- 1) l'automate SSF-FSM reste dans l'état "Surveillance" si le type de message était la notification et s'il y a d'autres points EDP armés ou une opération du type "CallInformationReport" ou "ApplyChargingReport" demandée;
- 2) l'automate SSF-FSM passe à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération du type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée;
- 3) l'automate SSF-FSM va dans l'état "Attente des instructions" si le type de message était la demande. Le traitement d'appel est interrompu.

3.3.30.2.2 Traitement des erreurs

Le traitement des erreurs pour les erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.30.3 Entité de réponse (SCF)

3.3.30.3.1 Procédure normale

Précondition de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours de traitement;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours de traitement.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie, et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction de l'entité SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute l'opération et y met fin.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et il y a encore des points EDP armés ou une opération du type "CallInformationReport" ou "ApplyChargingReport" demandée; ou

l'automate SCSM-FSM passe à l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou

l'automate SCSM-FSM passe à l'état "Préparer les instructions SSF" si le type de message était la demande.

3.3.30.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.31 Procédure OriginationAttemptAuthorized (Procédure d'autorisation des tentatives d'appel au départ)

3.3.31.1 Description générale

Cette opération est envoyée par l'entité SSF après que l'autorisation de la tentative d'appel a été donnée à partir du point de traitement PIC de la tentative d'appel au départ par l'opération O_Null & Authorize de l'automate BCSM et qu'un point TDP ait été détecté.

3.3.31.1.1 Paramètres

- serviceAddressInformation (triggerType, serviceKey, miscCallInfo):

le paramètre triggerType indique l'événement particulier qui a provoqué l'interruption de l'appel. Le paramètre miscCallInfo indique qu'une demande d'instructions a été présentée à l'entité SCF, de même que la catégorie de déclenchement (ligne, groupe, entreprise). Le paramètre serviceKey est utilisé pour identifier la logique de traitement de service devant être invoquée.

- bearerCapability:

voir la Recommandation Q.762 "Information du service utilisateur" et la Recommandation Q.931 "Elément d'information de la capacité support". La Recommandation fait référence au 6.4.4/Q.1214 pour les règles de codage applicables au paramètre de capacité support.

- calledPartyNumber:

voir la Recommandation Q.762 pour les informations de signalisation du numéro de demandé. Ce paramètre est utilisé pour identifier le demandé dans le sens en avant (il est par exemple utilisé pour le codage de l'élément d'information du numéro de demandé dans le protocole de signalisation du support).

- callingPartyNumber:
voir la Recommandation Q.762 "Information de signalisation du numéro de demandeur". Ce paramètre identifie le numéro d'Annuaire de la ligne de départ ou le numéro du demandeur reçu dans la signalisation du sous-système ISUP. Il convient de se référer au 6.4.4/Q.1214 relatif aux règles de codage du paramètre du numéro de demandeur.
- callingPartysCategory:
voir la Recommandation Q.762 "Information de signalisation de la catégorie de demandeur". Ce paramètre indique le type de demandeur (par exemple exploitant, publiphone, abonné ordinaire). Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre de catégorie de demandeur.
- iPSSPCapabilities:
voir la Recommandation Q.1290.
- iPAvailable:
voir la Recommandation Q.1290.
- iSDNAccessRelatedInformation:
transporte les mêmes informations que le paramètre de transport à l'accès de l'élément de protocole du sous-système ISUP dans la Recommandation Q.762. Ce paramètre contient des éléments d'information fournis à travers l'interface RNIS. Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage applicables aux informations relatives à l'accès RNIS.
- cGEncountered:
ce paramètre indique que l'appel a satisfait aux critères d'espacement d'appel.
- locationNumber:
voir la Recommandation Q.762 relative aux informations de signalisation du numéro d'emplacement. Ce paramètre est utilisé pour transporter l'adresse de zone géographique pour les services mobiles. Il est utilisé quand le "callingPartyNumber" ne contient pas d'informations sur l'emplacement géographique du demandeur (c'est-à-dire pour un acheminement dépendant de l'origine quand le demandeur est un abonné mobile).
- serviceProfileIdentifier:
voir l'Annexe A/Q.932. Ce paramètre permet au réseau RNIS de fournir l'identification et la sélection de terminaux spécifiques sur une interface usager-réseau configurée en multipoint afin d'assurer des profils multiples de services utilisateur quand les éléments d'information de la Recommandation Q.931 ne sont pas suffisants. Voir 6.4.4/Q.1214 pour les règles de codage de serviceProfileIdentifier.
- terminalType:
voir la Recommandation Q.1290. Identifie le type de terminal, pour que l'entité SCF puisse spécifier à l'entité SRF le type approprié de capacité (reconnaissance de la parole, DTMF, capacité de visualisation, etc.).
- chargeNumber:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 pour les règles de codage du paramètre "chargeNumber".
- servingAreaID:
voir la Recommandation Q.1290.
- dialledDigits:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 en ce qui concerne les règles de codage pour dialledDigits.
- callingPartyBusinessGroupID:
voir la Recommandation Q.1290.
- callingPartySubaddress:
voir la Recommandation Q.931 "Sous-adresse du demandeur".
- callingFacilityGroup:
voir la Recommandation Q.1290.

- callingFacilityGroupMember:
voir la Recommandation Q.1290.
- travellingClassMark:
voir la Recommandation Q.1290. Il convient de se référer au 6.4.4/Q.1214 applicable aux règles de codage du paramètre travellingClassMark.
- exploitant:
voir la Recommandation Q.1290.

3.3.31.2 Entité d'appel (SSF)

3.3.31.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) le traitement d'appel a progressé par l'intermédiaire du point de traitement de la tentative d'appel au départ O_Null & Authorize du modèle O-BCSM et la tentative d'appel au départ a été autorisée;
- 2) l'espacement des appels et la surcharge SS n° 7 ne sont pas applicables à l'appel, et l'appel ne doit pas être filtré;
- 3) un déclenchement est armé comme point TDP au point de détection Origination_Attempt_Authorized.

Postconditions de l'entité SSF:

- 1) pour un point TDP-R, une relation de commande a été établie et l'entité SSF attend des instructions en provenance de l'entité SCF;
- 2) pour un point TDP-N, le traitement d'appel continue au point de traitement d'appel Collect_Info.

3.3.31.2.2 Traitement des erreurs

Si l'entité SCF de destination n'est pas accessible, on termine la procédure de traitement d'appel (d'autres traitements font l'objet d'un complément d'étude). Quand le temporisateur T_{SSF} vient à expiration avant de recevoir toute opération, l'entité SSF interrompt l'interaction avec l'entité SCF et le traitement final de l'appel a lieu (par exemple acheminement vers une annonce terminale). Si le demandeur interrompt l'appel après l'envoi de l'opération OriginationAttemptAuthorized, l'entité SSF interrompt la relation de commande après que le premier message en provenance de l'entité SCF a été reçu: l'identificateur de transaction est conservé en mémoire jusqu'à l'expiration de la temporisation T_{SSF} .

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.31.3 Entité de réponse (SCF)

3.3.31.3.1 Procédure normale

Précondition de l'entité SCF:

- aucune relation de commande n'existe.

Postconditions de l'entité SCF:

- 1) une instance SLPI a été invoquée; une relation de commande est établie pour un point TDP-R;
- 2) pour un point TDP-N: aucune.

A la réception de l'opération OriginationAttemptAuthorized, l'entité SCSM passe de l'état "Repos" à l'état "Préparer les instructions SSF". Une relation de commande est créée avec l'entité SSF connexe. Une instance de programme de logique de traitement de service (SLPI) est invoquée pour le traitement de l'opération OriginationAttemptAuthorized. Au moyen de cette relation de commande, l'entité SCF peut influencer le traitement d'appel de base selon la logique de traitement de service invoquée. Les actions qui doivent être exécutées dans l'instance SLPI dépendent des paramètres transmis par l'intermédiaire de cette opération et l'instance SLPI (c'est-à-dire le service de réseau intelligent demandé lui-même).

3.3.31.3.2 Traitement des erreurs

Si l'opération OriginationAttemptAuthorized est rejetée, l'automate SCSM reste dans le même état. La fonction de maintenance est informée et aucune instance SLPI n'est invoquée. Le traitement générique des erreurs d'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.32 Procédure PlayAnnouncement

3.3.32.1 Description générale

Cette opération est utilisée pour l'interaction dans la bande avec un usager du réseau analogique ou pour l'interaction avec un utilisateur du RNIS.

3.3.32.1.1 Paramètres

- informationToSend:
ce paramètre indique une annonce, une tonalité ou une information à visualiser devant être envoyée à l'usager par l'entité SRF.
 - a) inbandInfo:
ce paramètre spécifie quelles informations dans la bande doivent être envoyées.
 - messageID:
ce paramètre indique le message ou les messages devant être envoyés parmi les messages suivants:
 - i) elementaryMessageID:
ce paramètre indique une annonce unique.
 - ii) text:
ce paramètre indique un texte devant être envoyé. Le texte sera transformé en informations dans la bande (parole) par l'entité SRF. Ce paramètre comprend deux sous-paramètres, messageContent (contenu de message) et attributs. Les attributs du texte peuvent comprendre des articles tels que le langage.
 - iii) elementaryMessageIDs:
ce paramètre spécifie une séquence d'annonces.
 - iv) variableMessage:
ce paramètre spécifie une annonce comprenant une ou plusieurs parties de longueur variable.
 - numberOfRepetitions (nombre de répétitions):
ce paramètre indique combien de fois au plus le message devra être envoyé à l'usager.
 - duration:
ce nombre indique pendant quelle durée maximale en secondes le message devra être joué/répété. Le ZÉRO indique une répétition sans fin.
 - interval:
ce paramètre indique l'intervalle de temps en secondes séparant les répétitions, c'est-à-dire le temps qui s'est écoulé entre la fin de l'annonce et le début de la répétition suivante. Ce paramètre peut uniquement être utilisé si le nombre de répétitions est > 1.
 - b) tone:
ce paramètre spécifie une tonalité devant être envoyée à l'usager.
 - toneID:
ce paramètre indique la tonalité devant être envoyée.
 - duration:
ce paramètre indique la durée en secondes de la tonalité devant être envoyée. Le ZÉRO indique une durée infinie.
 - c) displayInformation:
ce paramètre indique une chaîne de texte devant être envoyée à l'usager. Ces informations peuvent ne pas être reçues par l'usager du RTPC.

NOTE – Etant donné que les systèmes actuels de signalisation (DSS 1/ISUP) ne fournissent pas d'indication sur le fait qu'une information puisse ou non être visualisée par le terminal de l'usager, en cas d'interaction avec l'utilisateur du réseau RNIS, deux opérations consécutives "PlayAnnouncement" sont envoyées. La première contient les informations visualisées, la seconde les informations dans la bande devant être envoyées à l'usager. Etant donné que l'exécution des informations de visualisation par l'entité SRF devrait avoir une durée limitée, les informations dans la bande seront immédiatement envoyées par l'entité SRF vers l'usager, en séquence avec les informations visualisées.

- `disconnectFromIPForbidden`:
ce paramètre indique si une entité SRF devrait être ou non déconnectée de l'utilisateur quand toutes les informations ont été envoyées.
- `requestAnnouncementComplete`:
ce paramètre indique si une opération "SpecializedResourceReport" devra ou non être envoyée à l'entité SCF quand toutes les informations auront été envoyées.

3.3.32.2 Entité d'appel (SCF)

3.3.32.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) l'instance SLPI détecte que les informations devraient être envoyées à l'utilisateur;
- 2) une connexion entre l'utilisateur et une entité SRF doit être établie;
- 3) l'automate SCSM-FSM est dans l'état "Interaction avec l'utilisateur", sous-état "Attente de la réponse en provenance de l'entité SRF".

Postconditions de l'entité SCF:

- 1) si la valeur de "RequestAnnouncementComplete" a été mise sur VRAI, l'entité SCSM restera dans le sous-état "Attente de la réponse en provenance de l'entité SRF" et attendra l'opération "SpecializedResourceReport";
- 2) si la valeur de "RequestAnnouncementComplete" a été mise sur FAUX et si davantage d'informations doivent être envoyées ("DisconnectFromIPForbidden" étant mis sur VRAI), l'entité SCSM restera dans le sous-état "Attente de la réponse en provenance de l'entité SRF";
- 3) si la valeur "RequestAnnouncementComplete" a été mise sur FAUX et si des informations ne doivent plus être envoyées ("DisconnectFromForbidden" ayant été mis sur FAUX), l'entité SCSM se placera dans l'état "Préparer les instructions SSF".

3.3.32.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.32.3 Entité de réponse (SRF)

3.3.32.3.1 Procédure normale

Précondition de l'entité SRF:

- l'automate SRSM-FSM est dans l'état "Connecté", ou dans l'état "Interaction avec l'utilisateur" si l'entité SRF a reçu préalablement une opération en provenance de l'entité SCF.

Postconditions en provenance de l'entité SRF:

- 1) l'entité SRF envoie les informations à l'utilisateur comme cela est indiqué par "InformationToSend";
- 2) l'automate SRSM-FSM se place dans l'état "Interaction avec l'utilisateur" ou reste dans le même état;
- 3) si toutes les informations ont été envoyées et si le paramètre "RequestAnnouncementComplete" a été mis sur "VRAI", l'entité SRSM envoie une opération "SpecializedResourceReport" à l'entité SCF;
- 4) si toutes les informations ont été envoyées et si le paramètre "disconnectFromIPForbidden" a été mis sur "FAUX", l'entité SRSM déconnecte l'entité SRF de l'utilisateur.

L'annonce envoyée à l'utilisateur se termine dans les conditions suivantes:

- si aucun des paramètres "duration" (durée) ou "numberOfRepetitions" (nombre de répétitions) n'est spécifié, les conditions de fin d'annonce spécifique au réseau devront s'appliquer; ou
- si le paramètre "numberOfRepetitions" est spécifié quand toutes les répétitions auront été envoyées; ou
- si la durée est spécifiée, quand celle-ci est venue à expiration. L'annonce est répétée jusqu'à ce que cette condition soit satisfaite; ou
- si les paramètres "duration" et "numberOfRepetitions" sont spécifiés, lorsqu'une de ces deux conditions est satisfaite (quelle que soit celle qui vient en premier).

3.3.32.3.2 Traitement des erreurs

Si l'opération "Cancel" est reçue avant ou après le traitement de l'opération, celle-ci est immédiatement annulée et l'erreur "Canceled" est signalée à l'entité d'appel.

Le traitement générique des erreurs est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.33 Procédure PromptAndCollectUserInformation

3.3.33.1 Description générale

Cette opération est utilisée pour assurer l'interaction avec un correspondant afin de recueillir des informations.

3.3.33.1.1 Paramètres

a) collectedInfo:

i) collectedDigits:

- minimumNbOfDigits (Nombre minimal de chiffres):

si ce paramètre est absent, la valeur par défaut et définie comme étant égale à 1. Le paramètre "minimumNbOfDigits" (Nombre minimal de chiffres) spécifie le nombre minimal de chiffres valides qui doivent être recueillis.

- maximumNbOfDigits:

ce chiffre devrait toujours être présent et spécifie le nombre maximal de chiffres valides qui doivent être recueillis. La règle suivante s'applique: "maximumNbOfDigits" >= "minimumNbOfDigits".

- endOfReplyDigit (Chiffre de fin de réponse):

ce paramètre indique le chiffre utilisé pour signaler la fin de saisie.

Au cas où le "maximumNbOfDigits" = le "minimumNbOfDigits", le "EndOfReplyDigit" (chiffre de fin de réponse) pourrait exister, mais n'a pas d'autre signification. Ce paramètre peut comporter un ou deux chiffres.

Au cas où le "maximumNbOfDigits" > "minimumNbOfDigits", la règle suivante est applicable:

– si le paramètre "endOfReplyDigit" n'est pas présent, la fin de saisie est indiquée:

- 1) quand le temporisateur interchiffre vient à expiration; ou
- 2) quand le nombre de chiffres valides reçus équivaut au "maximumNbOfDigits",

– si le paramètre "endOfReplyDigit" n'est pas présent, la fin de saisie est indiquée:

- 1) quand le temporisateur interchiffre vient à expiration; ou
- 2) quand la fin du chiffre de fin de réponse est reçue; ou
- 3) quand le nombre de chiffres valides reçus équivaut au "maximumNbOfDigits".

Quand on atteint la fin de l'entrée, les chiffres recueillis sont envoyés de l'entité SRF vers l'entité SCF, y compris le chiffre "endOfReplyDigit" si ce dernier est reçu par l'entité SRF.

Au cas où le nombre de chiffres valides reçus est inférieur au "minimumNbOf Digits", le temporisateur interchiffre vient à expiration ou quand la fin du chiffre de fin d'envoi est reçu, l'entrée est spécifiée comme étant erronée.

– cancelDigit (chiffre d'annulation):

si ce paramètre est présent, le chiffre d'annulation peut être introduit par l'utilisateur afin de demander qu'une nouvelle tentative soit éventuellement effectuée. Tous les chiffres qui ont déjà été reçus par l'entité SRF sont supprimés et la même procédure "PromptAndCollectInformation" est de nouveau effectuée ; ainsi par exemple la même annonce pour demander des informations à l'utilisateur est représentée à l'utilisateur et les informations sont recueillies. Ce paramètre peut comprendre un ou deux chiffres.

Si ce paramètre n'est pas satisfait, l'utilisateur ne peut pas demander qu'une tentative de relance de l'opération soit effectuée.

- **startDigit (Premier chiffre ou chiffre de début):**
 si ce paramètre est présent, le chiffre de sélection de service indique le début du recueil des chiffres valides. Les chiffres qui sont obtenus par l'entité SRF avant que ce chiffre de sélection de service soit reçu, sont supprimés et ne sont pas considérés comme étant valides. Ce paramètre peut comprendre un ou deux chiffres.
 Si ce paramètre n'est pas satisfait, tous les chiffres reçus sont considérés comme étant valides.
- **firstDigitTimeout (Temporisation d'attente du premier chiffre):**
 si ce paramètre est présent, le premier chiffre devrait être reçu par l'entité SRF avant l'expiration du temporisateur d'attente du premier chiffre. Dans le cas où le premier chiffre ne serait pas reçu avant l'expiration du temporisateur de premier chiffre, l'entrée serait considérée comme étant erronée. Une fois que la réception du premier chiffre composé valide ou non valide a eu lieu, le temporisateur correspondant d'attente du premier chiffre est arrêté.
 Si ce paramètre n'est pas satisfait, l'entité SRF utilise une valeur par défaut (spécifique à l'exploitant de réseau) pour le temporisateur d'attente du premier chiffre dans lequel le premier chiffre composé valide ou non valide a été reçu.
 Si le paramètre "startDigit" (chiffre d'attente du premier chiffre) est satisfait, le temporisateur d'attente du premier chiffre est arrêté une fois que le chiffre d'attente du premier chiffre est reçu.
- **interDigitTimeOut (Temporisation interchiffre):**
 si ce paramètre est satisfait, tout chiffre subséquent valide ou non valide devrait être reçu par l'entité SRF avant que la temporisation interchiffre vienne à expiration. En conséquence, le temporisateur interchiffre est remis à zéro et redémarré.
 Au cas où un chiffre subséquent valide ou non valide n'est pas reçu avant que la temporisation interchiffre vienne à expiration et si le nombre de chiffres valides reçus est inférieur au "minimumNbOfDigits", la saisie est considérée comme accomplie.
 Au cas où un chiffre subséquent valide ou non valide n'est pas reçu avant que la temporisation interchiffre vienne à expiration et si le nombre de chiffres valides reçus est supérieur au "minimumNbOfDigits", et inférieur ou égal au "maximumNbOfDigits", la saisie est considérée comme accomplie.
 Si le chiffre "Temporisateur interchiffre" n'est pas satisfait, l'entité SRF utilise une valeur par défaut pour la durée interchiffre.
- **errorTreatment (Traitement des erreurs):**
 ce paramètre pris par défaut définit quelle action spécifique devrait être prise par l'entité SRF au cas où des conditions d'erreur se produiraient. La valeur par défaut est reportErrorToSCF.
- **interruptableAnnInd (Indicateur d'annonce interrompible):**
 ce paramètre est facultatif quand la valeur par défaut spécifiée est VRAI.
 Si ce paramètre est VRAI, l'annonce est interrompue après que le premier chiffre valide ou non valide ait été reçu par l'entité SRF. Si l'annonce est interrompue, une temporisation éventuelle d'attente du premier chiffre ne s'appliquera plus. Cependant, si l'annonce n'a pas été interrompue, une temporisation éventuelle d'attente du premier chiffre commencera après que l'annonce ait pris fin.
 Si ce paramètre est présent et explicitement mis sur FAUX, l'annonce ne sera pas interrompue après que le premier chiffre ait été reçu par l'entité SRF. Les chiffres reçus durant l'annonce sont supprimés et considérés comme étant non valides. Tous les autres paramètres spécifiés ("minimumNbOfDigits", "maximumNbOfDigits", "endOfReplyDigit", etc.) ne s'appliquent pas avant que l'annonce ait été achevée. L'éventuel temporisateur d'attente du premier chiffre est démarré après que l'annonce ait pris fin.
- **voiceInformation (Information relative à la parole):**
 ce paramètre est facultatif quand la valeur par défaut est spécifiée comme étant sur FAUX. Si le paramètre "voiceInformation" est mis sur FAUX, tous les chiffres valides ou non valides sont introduits par DTMF.
 Si ce paramètre est présent et explicitement mis sur VRAI, le demandeur doit fournir sous forme de parole toutes les informations valides ou non valides. L'entité SRF effectuera la reconnaissance vocale et traduira en chiffres les informations fournies. Un éventuel chiffre de fin de réponse devra également être fourni par la parole.

- voiceBack (Retour sous forme vocale):

ce paramètre est facultatif dans le cas où la valeur par défaut est spécifiée comme étant sur FAUX. Si le paramètre "Retour sous forme vocale" est FAUX, il n'y a pas de retour d'information vocale indiqué par l'entité SRF.

Si ce paramètre est présent et explicitement mis sur VRAI, l'annonce vocale des chiffres valides d'entrée reçus par l'entité SRF sera renvoyée au demandeur immédiatement après que la fin de l'entrée ait été reçue. L'annonce vocale des chiffres d'entrée non valides ne sera pas renvoyée au demandeur.

Une fin éventuelle de chiffre de fin de réponse n'est pas renvoyée sous forme vocale.

- b) disconnect FromIPForbidden:

ce paramètre indique si l'entité SRF devrait déclencher la déconnexion vers l'entité SSF-CCF après que l'interaction ait été achevée. Si le paramètre n'est pas présent ou mis sur VRAI, l'entité SRF ne devra pas déclencher la déconnexion.

- c) informationToSend:

ce paramètre indique une annonce, une tonalité ou des informations de visualisation devant être envoyées à l'utilisateur par l'entité SRF.

- i) inbandInfo:

ce paramètre spécifie les informations dans la bande devant être envoyées.

- messageID:

ce paramètre indique le message ou les messages devant être envoyés, parmi les messages suivants:

- elementary Message ID:

ce paramètre indique une annonce simple.

- text:

ce paramètre indique un texte devant être envoyé. Le texte devra être transformé en informations dans la bande (parole) par l'entité SRF. Ce paramètre comprend deux sous-paramètres, messageContent et attributs. Les attributs du texte peuvent comprendre des articles tels que le langage.

- elementaryMessageIDs:

ce paramètre spécifie une séquence d'annonces.

- variableMessage:

ce paramètre spécifie une annonce ayant une ou plusieurs parties variables.

- numberOfRepetitions:

ce paramètre indique combien de fois au maximum le message devra être envoyé à l'utilisateur.

- duration:

ce paramètre indique la durée maximale en secondes pendant laquelle le message sera joué/répété. Le ZÉRO indique une répétition sans fin.

- interval:

ce paramètre indique l'intervalle en secondes entre les répétitions, c'est-à-dire le temps entre la fin de l'annonce et le début de la répétition suivante. Ce paramètre peut uniquement être utilisé quand le nombre de répétitions est > 1.

- ii) tone:

ce paramètre spécifie une tonalité devant être envoyée à l'utilisateur.

- toneID:

ce paramètre indique la tonalité devant être envoyée.

- duration:

ce paramètre indique la durée en secondes de l'envoi de la tonalité. Le ZÉRO indique une durée illimitée.

iii) displayInformation:

ce paramètre indique une chaîne de texte devant être envoyée à l'utilisateur. Ces informations ne peuvent pas être reçues par un utilisateur du RTPC.

NOTE – Etant donné que les systèmes de signalisation actuels (DSS 1/ISUP) n'indiquent pas si une information peut être ou non visualisée par le terminal utilisateur, en cas d'interaction avec un usager du RNIS, le paramètre "displayInformation" n'est pas utilisé dans l'opération "PromptAndCollectUserInformation". A la place, une opération "PlayAnnouncement" contenant le paramètre "displayInformation" suivie d'une opération "PromptAndCollectUserInformation" contenant des informations dans la bande sont envoyées à l'utilisateur. Etant donné que l'exécution des informations visualisées par l'entité SRF aurait une durée limitée, les informations dans la bande seront envoyées immédiatement par l'entité SRF à l'utilisateur, en séquence avec les informations visualisées.

– digitsResponse (Réponse sous forme numérique):

ce paramètre contient les informations recueillies en provenance de l'utilisateur.

3.3.33.2 Entité de réponse (SCF)

3.3.33.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) l'instance SLPI détecte que les informations devraient être recueillies en provenance de l'utilisateur;
- 2) une connexion entre l'utilisateur et une entité SRF a été établie;
- 3) l'automate SCSM-FSM est dans l'état "Interaction avec l'utilisateur", sous-état "Attente de la réponse en provenance de l'entité SRF".

Postconditions de l'entité SCF:

- 1) les informations recueillies sont reçues en provenance de l'entité SRF en réponse à l'opération "PromptAndCollectUserInformation";
- 2) si la valeur du paramètre "disconnectFromIPForbidden" a été mise à FAUX, l'automate SCSM-FSM se mettra en position "Préparer les instructions SSF";
- 3) dans le cas contraire, l'automate SCSM-FSM restera dans le même état.

L'instance SLPI peut continuer l'exécution avant que la réponse soit reçue en provenance de l'opération "PromptAndCollectUserInformation", plus d'une opération pouvant être envoyée à l'entité SRF avant que la réponse ne soit reçue. Le paramètre "disconnectFromIPForbidden" peut uniquement être mis en position FAUX si l'opération "PromptAndCollectUserInformation" est la dernière opération envoyée en direction de l'entité SRF.

3.3.33.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.33.3 Entité de réponse (SRF)

3.3.33.3.1 Procédure normale

Précondition de l'entité SRF:

- l'automate SRSM-FSM est dans l'état "Connecté", ou dans l'état "Interaction avec l'utilisateur" si l'entité SRF a reçu préalablement une opération en provenance de l'entité SCF.

Postconditions de l'entité SRF:

- 1) l'entité SRF a envoyé les informations à l'utilisateur comme cela est indiqué par l'opération "informationToSend";
- 2) les informations recueillies par l'utilisateur sont envoyées à l'entité SCF comme RETOUR DU RÉSULTAT de l'opération "PromptAndCollectUserInformation";
- 3) si la valeur du paramètre "disconnectFromIPForbidden" a été mise à FAUX, l'entité SRF déclenche une déconnexion du canal support en direction de l'entité SSF et l'automate SRSM-FSM passe à l'état "Repos";
- 4) dans le cas contraire, l'automate SRSM-FSM passe à l'état "Interaction avec l'utilisateur" ou reste dans le même état.

L'annonce "Envoyer à l'utilisateur" se termine dans les conditions suivantes:

- si aucune des opérations "duration" ou "numberOfRepetitions" n'est spécifiée, alors les conditions de fin d'annonce qui sont spécifiques au réseau s'appliqueront; ou
- si "numberOfRepetitions" est spécifié, quand toutes les répétitions auront été envoyées, ou
- si la durée est spécifiée, quand la durée est venue à expiration. L'annonce se répète jusqu'à ce que cette condition soit satisfaite; ou
- si "duration" et "numberOfRepetitions" sont spécifiés, quand une de ces deux conditions est satisfaite (quelle que soit celle qui vient en premier).

Les conditions ci-dessus sont annulées si le paramètre "interruptableAnnInd" n'est pas mis à l'état FAUX et l'utilisateur a répondu par un chiffre pendant l'envoi de l'annonce. Dans ce cas, l'annonce prend fin immédiatement. Les procédures ci-dessus ne s'appliquent qu'aux informations dans la bande et les tonalités sont envoyées à l'utilisateur; pour "displayInformation" les conditions finales sont satisfaites dès l'envoi, c'est-à-dire qu'aucune interruption ne peut se produire.

Le paramètre "errorTreatment" spécifie la façon dont l'entité SRF devra traiter l'erreur. La valeur par défaut "reportErrorToSCF" signifie quelle erreur devra être signalée à l'entité SCF au moyen du retour d'erreur avec "ImproperCallerResponse". La valeur "help" indique qu'aucune erreur ne devra être indiquée à l'entité SCF, mais qu'une assistance devra être fournie à l'utilisateur sous forme d'une annonce d'incident dépendant du réseau (qui peut dépendre du contexte, c'est-à-dire de l'envoi du message). La valeur "repeatPrompt" indique qu'aucune erreur ne devra être indiquée à l'entité SCF, mais l'invite devra être répétée à l'utilisateur final. Les deux dernières procédures ne seront effectuées qu'une fois par opération "PromptAndCollectUserInformation".

NOTE – Traitement "endOfInput": la réception de toute condition "endOfInput" (par exemple endOfReplyDigit, cancelDigit, first DigitTimeout, interDigitTimeout) met immédiatement fin à l'entrée en cours. En d'autres termes, quand par exemple un endOfReplyDigit est reçu, la réception d'une condition cancelDigit subséquente ne sera plus traitée.

3.3.33.2 Traitement des erreurs

Si une opération Cancel est reçue avant ou pendant le traitement de l'opération, alors l'opération est immédiatement annulée et l'erreur "Annulé" est signalée à l'entité d'appel.

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

Si l'une quelconque de ces limites applicables aux paramètres est dépassée (par exemple minimumNbOfDigits > maximumNbOfDigits), alors une erreur d'opération s'est produite.

3.3.34 Procédure ReleaseCall (libération d'appel)

3.3.34.1 Description générale

Cette opération est utilisée pour rompre un appel existant par l'entité SCF à n'importe quelle phase de l'appel pour tous ceux qui participent à cet appel. Cette opération peut ne pas être envoyée à une entité SSF existante, sauf dans le cas de la procédure de transfert.

3.3.34.1.1 Paramètres

- Cause:
nombre indiquant à l'entité SSF la raison de libérer cet appel spécifique. Il peut être utilisé par l'entité SSF pour produire des tonalités spécifiques aux différents participants dans l'appel ou pour remplir le paramètre "cause" dans le message de libération.

3.3.34.2 Entité d'appel (SCF)

3.3.34.2.1 Procédure normale

Précondition de l'entité SCF:

- état 2.1, "Préparer les instructions SSF" ou état 2.3 "Attente de la notification ou de la demande".

Postcondition de l'entité SCF:

- état 1, "Repos", si aucune des opérations "CallInformationReport" ou "ApplyChargingReport" ne doit être reçue en provenance de l'entité SSF. Toutes les ressources (par exemple mise en file d'attente) relatives à l'appel sont libérées par l'entité SCF; ou

état 2.3, "Attente de la notification ou de la demande" si une opération "CallInformationReport" ou "ApplyChargingReport" doit encore être reçue à partir de l'entité SSF.

3.3.34.2.2 Traitement des erreurs

Le traitement des erreurs d'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.34.3 Entité de réponse (SSF)

3.3.34.3.1 Procédure normale

Précondition de l'entité SSF:

- état C, "Attente des instructions"; ou
- état F, "Surveillance" ou transfert de tâche en "Attente d'instructions".

Postcondition de l'entité SSF:

- "Idle", état a, après avoir envoyé une opération quelconque en cours de traitement "CallInformationReport". D'éventuels points EDP armés sont omis. Toutes les connexions et ressources relatives à l'appel sont libérées.

3.3.34.3.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.35 Procédure RemoveEntry (Retirer une entrée)

3.3.35.1 Description générale

L'opération "RemoveEntry" (Retirer une entrée) de la Recommandation X.500 est utilisée pour demander à une entité SDF de retirer une entrée feuille (soit une entrée objet ou une entrée alias) de l'arbre d'informations de l'annuaire (DIT). Pour une description complète de l'opération RemoveEntry, il convient de se référer au 11.2/X.511.

3.3.35.1.1 Paramètres

Voir 11.2.2/X.511 et 11.2.3/X.511.

3.3.35.2 Entité d'appel (SCF)

3.3.35.2.1 Procédure normale

Préconditions de l'entité SCF:

- SCSM: "Rattachement SDF" ou "Attente des demandes subséquentes".

Postconditions de l'entité SCF:

- SCSM: "Rattachement SDF"

Quand l'entité SCSM est dans l'état "Attente des demandes subséquentes" et qu'il existe un besoin de logique de traitement de service de retirer une entrée d'une entité SDF, un événement interne [(e2) Request_to_SDF] se produit. Avant que le processus d'application n'ait indiqué par un délimiteur (ou un temporisateur) que l'opération devrait être envoyée, l'entité SCSM reste dans l'état "Attente des demandes ultérieures" et l'opération n'est pas envoyée. L'opération est envoyée vers l'entité SDF dans un message contenant un argument de rattachement. L'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E5) Response_from_SDF_with_Bind ou (E4) Bind_Error] à l'opération de rattachement préalablement présentée à l'entité SDF provoque une transition de l'entité SCF vers l'état "Rattachement SDF" ou vers l'état "Repos". Quand l'entité SCSM s'est mise dans l'état "Repos", l'opération RemoveEntry est abandonnée. Dans l'état "Rattachement SDF", la réponse à l'opération RemoveEntry [(E7) Response_from_SDF] provoque une transition de l'entité SCF vers le même état ("Rattachement SDF"). Elle peut être soit le résultat de l'opération RemoveEntry, soit une erreur.

Quand l'entité SCSM est dans l'état "Rattachement SDF" et qu'un besoin de logique de traitement de service existe pour retirer une entrée de l'entité SDF, un événement interne se produit. Cet événement, appelé (e6) Request_to_SDF provoque une transition vers le même état "SDF Bound" et l'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E7) Response_from_SDF] vers l'opération RemoveEntry préalablement présentée à l'entité SDF provoque une transition de l'entité SCF vers le même état "Rattachement SDF". La réponse provenant de l'entité SDF peut être soit le résultat de l'opération RemoveEntry, soit une erreur.

3.3.35.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 11.2.4/X.511 et 11.2.5/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 2.2.2.

3.3.35.3 Entité de réponse (SDF)

3.3.35.3.1 Procédure normale

Préconditions de l'entité SDF:

- SDSM: "Rattachement SCF" ou "Rattachement en cours".

Postconditions de l'entité SDF:

- SDSM: "Rattachement SCF".

Quand l'entité SDF est dans l'état "Rattachement en cours", l'événement externe (E3) Request_from_SCF provoqué par la réception de l'opération "RemoveEntry" en provenance de l'entité SCF se produit. L'entité SDF n'effectue pas l'opération avant qu'une opération de rattachement n'ait été effectuée avec succès. Elle reste dans le même état.

Quand l'entité SDF est dans l'état "Rattachement SCF", l'événement externe (E7) Request_from_SCF provoqué par la réception d'une opération "RemoveEntry" provenant d'une entité SCF se produit. L'entité SDF attend la réponse apportée à l'opération.

A la réception de l'événement (E7) et avant de retirer l'article d'entrée, l'entité SDF effectue les actions suivantes:

- vérifier que l'objet devant être retiré existe et est une entrée feuille;
- vérifier que les droits d'accès afin de retirer l'entrée sont suffisants.

Une fois que les actions spécifiques indiquées ci-dessus sont effectuées avec succès, l'entrée est retirée de la base de données de l'entité SDF. Un résultat nul est renvoyé à l'entité SCF. L'envoi de ce résultat correspond à l'événement (e6) Response_to_SCF.

3.3.35.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 11.2.4/X.511 et 11.2.5/X.511, et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 2.2.2.

3.3.36 Procédure RequestCurrentStatusReport (Demande de compte rendu d'état)

3.3.36.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de surveiller l'état (occupation ou repos) d'une extrémité donnée.

3.3.36.1.1 Paramètres

- resourceID:
ce paramètre indique qu'une ressource physique d'extrémité qui est demandée par l'entité SCF doit être surveillée par l'entité SSF. Ce paramètre est l'un des identificateurs lineID, facilityGroupID, facilityGroupMemberID ou trunkGroupID.
- resourceStatus.

3.3.36.2 Entité d'appel (SCF)

3.3.36.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) l'instance SLPI a déterminé qu'une "demande de compte rendu d'état" doit être envoyée;
- 2) l'entité SCME est dans l'état "Compte rendu d'état Idle".

Postconditions de l'entité SCF:

- l'entité SCME est dans l'état "Attente du compte rendu d'état de l'entité SSF".

Quand l'instance SLPI demande la surveillance de l'état de la ressource physique d'extrémité, l'entité SCF envoie l'opération de demande de compte rendu d'état à l'unité SSF pour surveiller l'état d'une ressource spécifique d'extrémité. Alors, l'entité SCME passe à l'état "Attente du compte rendu d'état sur la réponse de l'entité SSF" à partir de l'état "compte rendu d'état Repos". L'entité SCME retourne à l'état "Compte rendu d'état Repos", quand l'entité SCF reçoit une réponse en provenance de l'entité SSF.

Une clarification sur l'usage de cette opération à l'intérieur ou à l'extérieur du contexte d'un appel fait l'objet d'un complément d'étude.

3.3.36.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.36.3 Entité de réponse (SSF)

3.3.36.3.1 Procédure normale

Préconditions de l'entité SSF:

- l'entité SSME est dans l'un des états suivants:
 - état ma: "IdleManagement";
 - état mb: "Traitement non associé à un appel" (Non_Call_Associated_Treatment).

Postconditions de l'entité SSF:

- l'entité SSME est dans l'état: "Traitement non associé à un appel".

A la réception de cette opération, l'entité SSF commence à surveiller l'état d'une ressource d'extrémité spécifique et l'entité SSF envoie le retour du résultat de cette opération vers l'entité SCF. Si une erreur s'est produite (par exemple l'entité SSF peut ne pas trouver une ressource d'extrémité spécifique), l'entité SSF renvoie vers l'entité SCF le retour d'erreur (ReturnError) de cette opération avec le type d'erreur approprié.

3.3.36.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.37 Procédure RequestFirstStatusMatchReport (Demande de premier compte rendu de correspondance)

NOTE – Pour la procédure RequestEveryStatusChangeReport, voir 3.3.55.

3.3.37.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de commencer à surveiller l'état d'une ressource d'extrémité donnée afin que celui-ci transite dans l'état spécifique (occupation ou Idle).

3.3.37.1.1 Paramètres

- resourceID:
 - cet identificateur indique que la ressource d'extrémité donnée qui est demandée par l'entité SCF doit être surveillée par l'entité SSF. Il est l'un des identificateurs lineID, facilityGroupID, facilityGroupMemberID, ou trunkGroupID.
- resourceStatus:
 - ce paramètre indique l'état qui est demandé afin de détecter une ressource d'extrémité par l'entité SCF.
- correlationID:
 - utilisé par l'entité SCF pour la corrélation avec une opération antérieure.
- monitorDuration:
 - ce paramètre indique la durée maximale de la surveillance dans l'entité SSF.

3.3.37.2 Entité d'appel (SCF)

3.3.37.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) l'instance SLPI a déterminé qu'une opération "demande de premier compte rendu de correspondance" doit être envoyée.
- 2) l'entité SCME est dans l'état "Compte rendu reçu à l'état Idle".

Postcondition de l'entité SCF:

- L'entité SCME est dans l'état "Attente de la réponse de l'entité SSF sur le compte rendu d'état".

Quand l'instance SLPI demande la surveillance de l'état de la ressource physique d'extrémité, l'entité SCF envoie l'opération de "demande de premier compte rendu de correspondance" à l'entité SSF pour demander à cette entité de commencer à surveiller une ressource d'extrémité donnée. Alors, l'entité SCME passe à l'état "Attente de la réponse de l'entité SSF sur le compte rendu d'état" à partir de l'état "compte rendu d'état Idle". Ensuite, dans le cas où l'entité SCF reçoit le retour d'erreur de cette opération, l'entité SCME retourne à l'état "Compte rendu d'état Idle".

L'entité SCME retourne à l'état "Compte rendu reçu d'état Idle", quand l'entité SCF reçoit une opération de compte rendu d'état en provenance de l'entité SSF.

Une clarification sur l'usage de cette opération à l'intérieur ou en dehors du contexte d'un appel est à l'étude.

3.3.37.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.37.3 Entité de réponse (SSF)

3.3.37.3.1 Procédure normale

Précondition de l'entité SSF:

- l'entité SSME est dans l'un des états suivants:
 - état ma: "IdleManagement";
 - état mb: "Traitement non associé à l'appel" (Non_Call_Associated_Treatment).

Postcondition de l'entité SSF:

- l'entité SSME est dans l'un des états suivants:
 - état ma: "IdleManagement";
 - état mb: "Traitement non associé à l'appel" (Non_Call_Associated_Treatment).

A la réception de cette opération, l'entité SSF commence à surveiller l'état d'une ressource d'extrémité donnée et l'entité SSF envoie le retour du résultat de cette opération vers l'entité SCF. Si une erreur s'est produite (par exemple l'entité SSF peut ne pas trouver une ressource d'extrémité donnée), l'entité SSF renvoie vers l'entité SCF le retour d'erreur (ReturnError) de cette opération avec le type d'erreur approprié.

L'entité SSF surveille en continu l'état de la ressource d'extrémité donnée jusqu'à ce que l'état transite vers l'état donné, ou jusqu'à la fin de la temporisation donnée qui est spécifiée par le paramètre monitorDuration. Si l'entité SSF détecte la transition d'état vers un état spécifique, elle envoie l'opération "compte rendu d'état" à l'entité SCF avec le paramètre monitorCondition mis à "statusReport". Si l'expiration de la temporisation s'est produite, l'entité SSF envoie l'opération "StatusReport" à l'entité SCF avec le paramètre monitorCondition mis sur "TimerExpired". Dans un autre cas, l'entité SSF reçoit l'opération "Cancel Status Report" provenant de l'entité SCF, l'entité SSF envoyant l'opération "StatusReport" à l'entité SCF avec le paramètre monitorCondition mis sur "Canceled" (Annulé). Une fois que l'entité SSF envoie cette opération, l'entité SSME doit transiter vers l'état "IdleManagement", à moins qu'il y ait d'autres traitements de l'opération "Non associé à l'appel", auquel cas l'entité SSF devrait rester dans l'état "Non_Call_Associated_Treatments" (Traitements non associés à l'appel).

3.3.37.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.38 Procédure RequestNotificationChargingEvent (Evénement de demande de notification de taxation)

3.3.38.1 Description générale

Cette opération est utilisée pour donner l'ordre à l'entité SSF de gérer les événements de taxation qui sont reçus par d'autres entités fonctionnelles, indépendamment de la commande de l'instance de logique de traitement de service. Cette opération prend en charge les options visant à assurer les interactions concernant la taxation (se référer à l'Appendice II.4/Q.1214 "Scénarios de taxation"). Etant donné que plusieurs configurations de connexion peuvent être établies pendant un appel, il existe une possibilité que l'opération "RequestNotificationChargingEvent" soit invoquée plusieurs fois. Pour chaque configuration de connexion, un événement de taxation peut être utilisé plusieurs fois.

3.3.38.1.1 Paramètres

- SequenceofChargingEvent:
 - ce paramètre contient une liste des événements de taxation et des types correspondants de surveillance, de même que les demi-appels correspondants. Pour chaque élément dans la liste, les éléments suivants d'information sont inclus:
 - eventTypeCharging:
 - ce sous-paramètre indique le type d'événement de taxation. Son contenu est spécifique à l'exploitant de réseau et peut être des "impulsions de taxation" ou des "messages de taxation".
 - monitorMode:
 - ce sous-paramètre indique le monitorMode applicable au sous-paramètre correspondant "eventTypeCharging". Le mode "Monitor" (surveillance) peut être "interrompu"; "notifyAndContinue" ou "transparent".
 - legID (identificateur de demi-appel):
 - ce sous-paramètre indique l'identificateur de demi-appel du sous-paramètre de type de taxation de l'événement correspondant.

3.3.38.2 Entité d'appel (SCF)

3.3.38.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre l'entité SCF et l'entité SSF;
- 2) une instance SLPI a déterminé qu'un "RequestNotificationChargingEvent" devait être envoyé par l'entité SCF.

Postconditions de l'entité SCF:

- 1) pas de transition d'état de l'automate FSM.
- 2) l'exécution de l'instance SLPI peut se poursuivre.

L'automate SCSM-FSM est dans l'état "Préparer l'instruction SSF" ou dans l'état "Mis en file d'attente de l'entité FSM". Cette opération est invoquée par l'entité SCF si une instance SLPI aboutit à donner l'instruction à l'entité SSF d'assurer les interactions relatives à la taxation. Cela ne provoque pas de transition d'état de l'automate SCSM-FSM.

3.3.38.2.2 Traitement des erreurs

Le traitement générique des erreurs est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.38.3 Entité de réponse (SSF)

3.3.38.3.1 Procédure normale

Préconditions de l'entité SSF:

- SSF-FSM: état c: "Attente des instructions"; ou
- SSF-FSM: état d: "Attente de la fin de l'interaction avec l'utilisateur"; ou

SSF-FSM: état e: "Attente de la fin de la connexion temporaire"; ou
SSF-FSM: état f: "Surveillance"; ou
entité SSF-FSM d'assistance/de transfert de tâche état b: "Attente des instructions".

Postcondition de l'entité SSF:

- aucune transition d'état de l'automate FSM.

A la réception de cette opération, l'entité SSF exécute des actions afin d'assurer les interactions concernant la taxation selon les éléments d'information inclus dans l'opération. L'événement de taxation demandé peut être provoqué par:

- a) une autre instance SLPI; ou
- b) un autre autocommutateur.

Indépendamment de ce qui a provoqué l'événement de taxation, l'entité SSF exécute une des actions suivantes lorsque l'événement de taxation se produit (selon le monitorMode correspondant):

- **Interrupted:**
il convient de notifier à l'entité SCF l'événement de taxation en utilisant "EventNotificationCharging", de ne pas procéder au traitement de l'événement et de ne pas propager le signal. Cependant, l'appel et le traitement existant de la taxation ne seront pas interrompus dans l'entité SSF.
- **NotifyAndContinue:**
il convient de notifier à l'entité SCF l'événement de taxation en utilisant "EventNotificationCharging", et de continuer le traitement de l'événement ou du signal sans attendre les instructions de l'entité SCF (traitées comme point EDP-N pour les événements BCSM).
- **Transparent:**
ne pas notifier l'événement à l'entité SCF. Cela met fin à la surveillance d'un événement de taxation préalablement demandé.

Les événements de taxation demandés sont surveillés jusqu'à ce qu'ils soit mis fin à ces événements par un mode de surveillance transparent (ou dans le cas d'événements de taxation) jusqu'à la fin de la configuration de connexion.

Dans le cas où des opérations multiples de type "RequestNotificationChargingEvent" sont reçues pour la même configuration de connexion avec le même identificateur "eventTypeCharging" et "legID", seul le dernier "monitorMode" reçu s'appliquera.

3.3.38.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.39 Procédure RequestReportBCSMEvent

3.3.39.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de surveiller un événement associé à l'appel (par exemple des événements BCSM tels qu'occupation ou non réponse), puis renvoyer une notification à l'entité SCF quand l'événement est détecté.

NOTE – Si l'opération RequestReportBCSMEvent demande que le point de détection actuel à partir duquel le traitement d'appel a été suspendu soit armé, la prochaine occurrence du point de détection rencontré pendant le traitement du modèle BCSM sera détectée (c'est-à-dire pas le point de détection actuel à partir duquel l'appel a été suspendu).

3.3.39.1.1 Paramètres

- bcsmEvents:
ce paramètre spécifie l'événement ou les événements dont un compte rendu est demandé.
 - a) eventTypeBCSM:
ce paramètre spécifie le type d'événement dont un compte rendu est demandé. Les valeurs de origAttemptAuthorized et de termAttemptAuthorized ne sont pas valides pour le paramètre eventTypeBCSM.

b) monitorMode:

ce paramètre indique la façon dont l'événement devrait être indiqué. Quand le mode "monitorMode" est interrompu, l'événement sera considéré comme étant une demande, si le "monitorMode" est "notifyAndContinue", l'événement sera considéré comme étant une notification, si le "monitorMode" est transparent, l'événement ne devra pas être signalé.

c) legID (élément d'identification de demi-appel):

ce paramètre précise pour quel correspondant de l'appel l'événement est indiqué. L'entité SCF utilisera uniquement l'option "sendingSideID" (identificateur de la partie émettrice).

- sendingSideID (identificateur de la partie émettrice):

on suppose les valeurs suivantes pour "legID" (élément d'identification du demi-appel):

"legID" = 1 indique le correspondant qui était présent au moment de l'opération "Point de détection initial" ou de l'opération spécifique au point de détection" (en cas de déclenchement de demi-appel, le correspondant provoquant le déclenchement), ou l'entité correspondante qui a été créée par l'opération "InitiateCallAttempt".

"legID" = 2 indique que l'entité correspondante a été créée par une opération "Connect" (Connexion), ou en cas de déclenchement de demi-appel, le correspondant ne provoquant pas le déclenchement.

En cas de non inclusion, on prendra les cas suivants par défaut:

legID = 1 pour les événements CollectedInfo, AnalysedInformation, O-Abandon et T-Abandon.

legID = 2 pour les événements RouteSelectFailure, O-CalledPartyBusy, O-NoAnswer, O-Answer, T-Busy, T-NoAnswer et T-Answer.

Le paramètre "legID" devra toujours être inclus pour les événements O-MidCall, O-Disconnect, T-MidCall et T-Disconnect.

d) dPSpecificCriteria:

ce paramètre indique les informations spécifiques au point EDP devant être armé.

- numberOfDigits:

ce paramètre indique le nombre de chiffres devant être recueilli par l'entité SSF pour l'événement CollectedInfo. Si le nombre de chiffres indiqués est recueilli, l'entité SSF indiquera l'événement à l'entité SCF.

- applicationTimer:

ce paramètre indique le temporisateur d'application pour l'événement NoAnswer. Si l'utilisateur ne répond à l'appel à l'intérieur du temps imparti, l'entité SSF devra signaler l'événement à l'entité SCF. Il est prévu que la durée de cette temporisation soit plus courte que celle de la temporisation de non-réponse du réseau.

e) bcsmeventCorrelationID:

cet identificateur est utilisé par l'entité SCF pour la corrélation avec une opération antérieure.

3.3.39.2 Entité d'appel (SCF)

3.3.39.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre l'entité SCF et l'entité SSF;
- 2) une instance SLPI a décidé qu'une demande de compte rendu d'événement est nécessaire;
- 3) l'automate SCSM-FSM est dans l'état approprié pour envoyer "RequestReportBCSMEvent".

Postconditions de l'entité SCF:

- 1) l'automate SCSM-FSM reste dans le même état;
- 2) l'exécution de l'instance SLPI se poursuit;
- 3) si tous les points EDP ont été désarmés et si aucune opération quelle soit du type CallInformationReport ou ApplyChargingReport n'est en cours, il est mis fin à la relation de commande avec l'entité SSF. Si aucune autre relation ne se poursuit, l'entité SCSM retournera à l'état "Repos".

3.3.39.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.39.3 Entité de réponse (SSF)

3.3.39.3.1 Procédure normale

Précondition de l'entité SSF:

- l'automate SSF-FSM est dans l'état "Attente des instructions" ou dans l'état "Surveillance".

Postconditions de l'entité SSF:

- 1) les points EDP demandés ont été armés comme cela est indiqué;
- 2) les événements préalablement demandés sont surveillés jusqu'à leur terme par un mode de surveillance transparent, et ce jusqu'à la fin de l'appel, jusqu'à ce que les points EDP soient détectés ou jusqu'à ce que le demi-appel correspondant soit libéré;
- 3) l'automate SSF-FSM reste dans le même état;
- 4) si tous les points EDP ont été désarmés et si aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'a été demandée, l'automate SSF-FSM va à l'état "Repos".

3.3.39.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits dans en 3.4.

3.3.40 Procédure ResetTimer (Remise à zéro du temporisateur)

3.3.40.1 Description générale

Cette opération de classe 2 est utilisée par l'entité SCF pour remettre à zéro le temporisateur d'application T_{SSF} , afin d'éviter le délai d'attente dans l'entité SSF.

3.3.40.1.1 Paramètres

- timerID (Identificateur de temporisation):
ce paramètre a une valeur par défaut identifiant la temporisation T_{SSF} .
- timerValue (Valeur de temporisation):
ce paramètre a spécifié la valeur à laquelle la temporisation T_{SSF} devait être initialisée.

3.3.40.2 Entité d'appel (SCF)

3.3.40.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre l'entité SCF et l'entité SSF;
- 2) une instance SLPI a déterminé par la fin de temporisation de garde $T_{SCF-SSF}$ que l'opération de remise à zéro du temporisateur doit être demandée pour éviter la temporisation T_{SSF} dans l'entité SSF.

Postcondition de l'entité SCF:

- l'instance SLPI a remis à zéro la temporisation de garde $T_{SCF-SSF}$ dans l'entité SSF.

3.3.40.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.40.3 Entité de réponse (SSF)

3.3.40.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) le traitement d'appel de base a été interrompu au point de détection;

- 3) l'automate SSF-FSM est dans l'état "Attente d'instructions" ou "Attente de fin d'interaction avec l'utilisateur" ou "Attente de fin de connexion temporaire" ou si l'automate SSF-FSM d'assistance/de transfert de tâche est dans l'état "Attente des instructions" ou dans l'état "Attente de la fin de l'interaction avec l'utilisateur".

NOTE – Que le temporisateur T_{SSF} se trouve ou non dans l'état "Attente de la fin de l'interaction avec l'utilisateur" ou dans l'état "Attente de la fin de la connexion temporaire" dépend de l'exploitant de réseau.

Postconditions de l'entité SSF:

- 1) le temporisateur T_{SSF} a été remis à zéro;
- 2) l'automate SSF-FSM reste dans le même état.

3.3.40.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.41 Procédure RouteSelectFailure

3.3.41.1 Description générale

Cette opération est envoyée par l'entité SSF vers l'entité SCF après détection d'une condition valide de déclenchement au point de détection RouteSelectFailure, ou pour signaler un événement demandé par RequestReportBCSMEvent. Il convient de se référer au 4.2.2.2/Q.1214 pour la sémantique additionnelle relative à la simulation d'appel par modèle.

3.3.41.1.1 Paramètres

- serviceAddressInformation (serviceKey, miscCallInfo, triggerType):
voir la Recommandation Q.1290. Le paramètre triggerType indique l'événement particulier qui a provoqué l'interruption de l'appel. Le paramètre miscCallInfo indique qu'une demande d'instructions a été présentée à l'entité SCF, de même que la catégorie de déclenchement (ligne, groupe, entreprise). Le paramètre serviceKey est utilisé pour identifier la logique de traitement de service devant être invoquée.
- bearerCapability:
ce paramètre indique à l'utilisateur le type de connexion de capacité du support. Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage applicables au paramètre de capacité du support.
- calledPartyNumber:
voir la Recommandation Q.762 pour les informations de signalisation du numéro de demandé. Ce paramètre est utilisé pour identifier le demandé dans le sens en avant.
- callingPartyNumber:
voir la Recommandation Q.762 "Information de signalisation du numéro de demandeur". Il convient de se référer au 6.4.4/Q.1214 relative aux règles de codage du paramètre du numéro de demandeur.
- callingPartysCategory:
voir la Recommandation Q.762 "Information de signalisation de la catégorie de demandeur". Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre de catégorie de demandeur.
- iPSSPCapabilities:
voir la Recommandation Q.1290.
- iPAavailable:
voir la Recommandation Q.1290.
- iSDNAccessRelatedInformation:
ce paramètre contient des éléments d'information (pouvant être multiples) conformément à la Recommandation Q.931. Voir les informations de signalisation du paramètre AccessTransport dans les Recommandations Q.762, Q.763 et Q.931. Il convient de se référer au 6.4.4/Q.1214 (Informations analysées) pour les règles de codage applicables aux informations relatives à l'accès RNIS.
- cGEncountered:
voir la Recommandation Q.1290.

- locationNumber:
voir la Recommandation Q.762 relative aux informations de signalisation du numéro d'emplacement. Ce paramètre est utilisé pour transporter l'adresse de zone géographique pour les services mobiles. Il est utilisé quand le "callingPartyNumber" ne contient pas d'informations sur l'emplacement géographique du demandeur (c'est-à-dire pour un acheminement dépendant de l'origine quand le demandeur est un abonné mobile).
- serviceProfileIdentifier:
voir l'Annexe A/Q.932. Voir 6.4.4/Q.1214 pour les règles de codage de serviceProfileIdentifier.
- terminalType:
voir la Recommandation Q.1290. Identifie le type de terminal, pour que l'entité SCF puisse spécifier à l'entité SRF le type approprié de capacité (reconnaissance de la parole, DTMF, capacité de visualisation, etc.).
- chargeNumber:
voir la Recommandation Q.1290.
- servingAreaID:
voir la Recommandation Q.1290.
- dialledDigits:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 applicable aux règles de codage.
- callingPartyBusinessGroupID:
voir la Recommandation Q.1290.
- callingPartySubaddress:
voir la Recommandation Q.931.
- callingFacilityGroup:
voir la Recommandation Q.1290.
- callingFacilityGroupMember:
voir la Recommandation Q.1290.
- failureCause:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 applicable aux règles de codage.
- originalCalledPartyID:
voir la Recommandation Q.762. Informations de signalisation du demandé initial.
- préfixe:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 applicable aux règles de codage.
- redirectingPartyID:
ce paramètre (le cas échéant) est le numéro d'annuaire du dernier réachemineur.
- redirectionInformation:
voir la Recommandation Q.763 Informations de signalisation de réacheminement.
- routeList:
voir la Recommandation Q.1290.
- travellingClassMark:
voir la Recommandation Q.1290.
- carrier (exploitant):
voir la Recommandation Q.1290.

3.3.41.2 Entité d'appel (SSF)

3.3.41.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) le numéro du demandé est disponible et le type d'adresse est déterminé;
- 3) l'espacement des appels ou le filtrage du service ne s'applique pas au segment d'appel;
- 4) les critères de point de détection sont satisfaits;
- 5) pour un point TDP-R, il n'y a pas de relation de commande existante influençant le segment d'appel.

Postconditions de l'entité SSF:

- 1) pour un point TDP-R, le traitement d'appel de base a été interrompu au point de détection `Route_Select_Failure`, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel de base se poursuit au point `0_Exception`, et aucune relation de commande n'est établie;
- 3) pour un point EDP, comme pour la procédure `EventReportBCSM` (voir 3.3.22.2.1).

3.3.41.2.2 Traitement des erreurs

Le traitement générique pour les erreurs relatives à l'opération est décrit au 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.41.3 Entité de réponse (SCF)

3.3.41.3.1 Procédure normale

Préconditions de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie, et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute le traitement et y met fin.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et s'il y a encore des points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM va à l'état "Repos" si le type de message était la notification et s'il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandé; ou
l'automate SCSM-FSM va à l'état "Préparer les instructions SSF" si le type de message était la demande.

3.3.41.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.42 Procédure Search (Recherche)

3.3.42.1 Description générale

L'opération Search (Recherche) contenue dans la Recommandation X.500 est utilisée pour rechercher une section dans l'arborescence de l'Annuaire (DIT) de résident SDF pour des entrées intéressantes et de renvoyer les informations sélectionnées à partir de ces entrées. Pour une description complète de l'opération "Search", il convient de se référer au 10.2/X.511.

3.3.42.1.1 Paramètres

Voir 10.2.2/X.511 et 10.2.3/X.511.

3.3.42.2 Entité d'appel (SCF)

3.3.42.2.1 Procédure normale

Précondition de l'entité SCF:

- SCSM: "Rattachement SDF" ou "Attente des demandes subséquentes".

Postcondition de l'entité SCF:

- SCSM: "Rattachement SDF"

Quand l'entité SCSM est dans l'état "Attente des demandes subséquentes" et qu'il existe un besoin de logique de traitement de service de rechercher et/ou de lire des informations de l'entité SDF, un événement interne [(e2) Request_to_SDF] se produit. Avant que le processus d'application n'ait indiqué par un délimiteur que l'opération devrait être envoyée, l'entité SCSM reste dans l'état "Attente des demandes subséquentes" et l'opération n'est pas envoyée. L'opération est envoyée vers l'entité SDF dans un message contenant un argument de rattachement. L'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E5) Response_from_SDF_with_Bind ou (E4) Bind_Error] à l'opération de rattachement préalablement présentée à l'entité SDF provoque une transition de l'entité SCF à l'état "Rattachement SDF" ou à l'état "Repos". Une fois que l'entité SCSM est allée à l'état "Repos", l'opération Search est abandonnée. Dans l'état "Rattachement SDF", la réponse de l'opération Search [(E7) Response_from_SDF] provoque une transition de l'entité SCF vers le même état ("Rattachement SDF"). Cela peut être le résultat de l'opération Search ou une erreur.

Quand l'entité SCSM est dans l'état "Rattachement SDF" et qu'il existe un besoin de logique de traitement de service de rechercher et/ou lire des informations de l'entité SDF, un événement interne se produit. Cet événement, appelé (e6) Request_to_SDF provoque une transition vers le même état "Rattachement SDF" et l'entité SCSM attend la réponse en provenance de l'entité SDF. La réception de la réponse [(E7) Response_from_SDF] à l'opération Search préalablement présentée à l'entité SDF provoque une transition de l'entité SCF vers le même état "Rattachement SDF". La réponse en provenance de l'entité SDF peut être le résultat de l'opération Search ou une erreur.

3.3.42.2.2 Traitement des erreurs

Le traitement générique des erreurs d'opération est décrit en 10.2.4/X.511 et 10.2.5/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 2.2.2.

3.3.42.3 Entité de réponse (SDF)

3.3.42.3.1 Procédure normale

Préconditions de l'entité SDF:

- SDSM: "Rattachement SCF" ou "Rattachement en cours".

Postconditions de l'entité SDF:

- SDSM: "Rattachement SCF".

Quand l'entité SDF est dans l'état "Rattachement en cours", l'événement externe (E3) Request_from_SCF provoqué par la réception d'une opération "Search" en provenance de l'entité SCF se produit. L'entité SDF ne traite pas l'opération avant qu'une opération "Bind" (Rattachement) ait été exécutée avec succès. Elle reste dans le même état.

Quand l'entité SDF est dans l'état "Rattachement SCF", l'événement externe (E7) Request_from_SCF provoqué par la réception d'une opération "Search" en provenance de l'entité SCF se produit. L'entité SDF attend la réponse à l'opération.

A la réception de l'événement (E7) et avant la mise à jour des différents attributs spécifiés dans les paramètres de l'opération, l'entité SDF effectuera les actions suivantes:

- vérifier que les objets auxquels la demande permet d'accéder existent;
- vérifier que l'utilisateur au nom duquel la demande est présentée a suffisamment de droits d'accès pour modifier les objets et les attributs auxquels on accède pendant l'exécution de l'opération;
- vérifier que les attributs sur lesquels une opération devrait être effectuée existent dans l'objet.

Après que les actions spécifiées indiquées ci-dessus aient été exécutées avec succès, l'entité SDF renverra vers l'entité SCF tous les attributs possibles qui satisfont aux critères d'accès à l'information. L'envoi des résultats correspond à l'événement (e6) Response_to_SCF.

3.3.42.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit au 10.2.4/X.511 et 10.2.5/X.511 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 2.2.2.

3.3.43 Procédure SelectFacility

3.3.43.1 Description générale

Cette opération est envoyée par l'entité SCF vers l'entité SSF afin de demander à l'entité SSF d'effectuer les actions de traitement terminal d'appel de base afin de sélectionner une ligne terminale si elle est libre, ou de sélectionner une ligne au repos à partir d'un groupe de faisceau de recherche multiligne, ou de sélectionner le cas échéant un faisceau de circuits libre à partir d'un groupe de faisceaux. Si aucune ligne ni aucun faisceau n'est disponible, l'entité SSF déclare que l'équipement terminal est occupé. Il convient de se référer au 4.2.2.2/Q.1214, pour la sémantique additionnelle relative à la simulation d'appel par modèle.

3.3.43.1.1 Paramètres

- alertingPattern (séquence d'alerte):
voir la Recommandation Q.1290. Cela ne s'applique que si la signalisation du réseau accepte ce paramètre ou si l'entité SSF se trouve dans le centre local de rattachement de l'abonné.
- destinationNumberRoutingAddress:
ce paramètre contient le numéro du demandé vers lequel l'appel doit être acheminé. Le codage de ce paramètre est défini dans la Recommandation Q.763.
- iSDNAccessRelatedInformation:
ce paramètre contient des éléments d'information (pouvant être multiples) conformes à la Recommandation Q.931. Voir les informations de signalisation du paramètre de transport à l'accès dans les Recommandations Q.762, Q.763 et Q.931. Il convient de se référer au 6.4.4/Q.1214 (Informations analysées) pour les règles de codage applicables aux informations relatives à l'accès RNIS.
- calledFacilityGroup:
voir la Recommandation Q.1290.
- calledFacilityGroupMember:
voir la Recommandation Q.1290.
- originalCalledPartyID:
voir la Recommandation Q.762. Informations de signalisation du numéro de demandé initial.

3.3.43.2 Entité d'appel (SCF)

3.3.43.2.1 Procédure normale

Précondition de l'entité SCF:

- 1) une relation de commande existe entre une entité SCF et une entité SSF;
- 2) une instance SLPI a déterminé qu'une opération "SelectFacility" doit être envoyée par l'entité SCF.

Postcondition de l'entité SCF:

- l'exécution par l'instance SLPI peut se poursuivre.

Dans l'état "Préparer les instructions SSF" de l'automate SCSM-FSM, cette opération est invoquée par une entité SCF si la logique de traitement de service a pour résultat de demander à une entité SSF d'acheminer un appel vers une destination spécifique et de continuer le traitement de cet appel jusqu'au point de traitement d'appel `Select_Facility_and_Present_Call`. Si aucune surveillance d'événement n'a été demandée et si aucun compte rendu (`CallInformationReport` et `ApplyChargingReport`) n'a été demandé dans une opération préalable, l'automate SCSM-FSM passe à l'état "Repos". Sinon, si une surveillance d'événement a été demandée ou si un compte rendu quelconque (`CallInformationReport` et `ApplyChargingReport`) a été demandé, l'automate SCSM-FSM passe à l'état "Attente de notification ou de demande".

3.3.43.2 Traitement des erreurs

Si des messages de refus ou d'erreur sont reçus, l'entité SCSM informe l'instance SLPI du message d'erreur et reste dans l'état "Préparer les instructions SSF".

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.43.3 Entité de réponse (SSF)

3.3.43.3.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) le traitement d'appel de base a été interrompu au point de détection;
- 3) l'entité SSF attend des instructions.

Postconditions de l'entité SSF:

- 1) l'entité SSF effectue l'action de traitement d'appel afin d'acheminer l'appel vers la destination spécifiée et applique la séquence d'alerte appropriée;
- 2) dans l'automate T-BCSM, le traitement d'appel reprend au point de traitement d'appel `Select_Facility_and_Present_Call`.

3.3.43.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.44 Procédure `SelectRoute`

3.3.44.1 Description générale

Cette opération demande à l'entité SSF d'effectuer les actions de traitement d'appel de base au départ pour déterminer les informations d'acheminement et pour sélectionner l'acheminement d'un appel, soit à partir des informations d'appel disponibles dans l'entité SSF ou des informations fournies par l'entité SCF. Sur la base des informations d'adresse fournies, l'entité SSF essaie de choisir une voie d'acheminement de premier choix pour l'appel. Si l'itinéraire est occupé, l'entité SSF essaie de choisir un autre itinéraire. L'entité SSF peut échouer dans sa tentative de choix d'une voie d'acheminement pour l'appel si toutes les voies d'acheminement sont occupées.

3.3.44.1.1 Paramètres

- `destinationRoutingAddress`:
représente une liste de numéros de demandés (par itinéraires de premier choix et par itinéraires de débordement).
- `alertingPattern`:
voir la Recommandation Q.1290. Elle ne s'applique que si la signalisation de réseau exploite ce paramètre ou si l'entité SSF est l'autocommutateur local de rattachement de l'abonné.
- `correlationID`:
utilisé par l'entité SCF pour associer l'opération "AssistRequestInstructions" en provenance de l'entité SSF assistante avec la demande en provenance de l'entité de déclenchement SSF. L'identificateur "CorrelationID" est utilisé dans le contexte d'une procédure de transfert de tâche et seulement si l'identificateur id n'est pas intégré dans l'adresse "destinationRoutingAddress".

- iSDNAccessRelatedInformation:
transporte les mêmes informations que le paramètre d'Accès Transport de l'élément du sous-protocole ISUP décrit dans la Recommandation Q.762.
- originalCalledPartyID:
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.
- routeList:
voir la Recommandation Q.1290.
- scfID:
utilisé par une entité SSF assistante pour identifier vers quelle entité SCF l'adresse "destinationRoutingAddress" devrait être envoyée.
- travellingClassMark:
voir la Recommandation Q.1290.
- carrier:
voir la Recommandation Q.1290. Dans ce message, le champ de sélection d'exploitant est zéro (00000000) et l'identificateur d'exploitant indique l'exploitant devant utiliser l'appel.

3.3.44.2 Entité d'appel (SCF)

3.3.44.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une tentative d'appel a eu lieu au départ;
- 2) l'autorisation/la possibilité d'effectuer un appel sortant a été vérifiée;
- 3) les informations de destination sont disponibles dans l'entité SSF ou fournies par l'entité SCF;
- 4) le traitement de base de l'appel a été interrompu à un des points de détection suivants:
 - Origination_Attempt_Authorized;
 - Collected_Info;
 - Analysed_Info;
 - Route_Select_Failure;
 - O_Called_Party_Busy;
 - O_No_Answer;
 - O_Disconnect (déconnexion du demandé uniquement),
- 5) une relation de commande a été établie et l'instance SLPI effectue le traitement de la demande entrante.

Postconditions de l'entité SCF:

- 1) l'exécution de l'instance SLPI se termine si aucune surveillance n'est demandée;
- 2) l'exécution de l'instance SLPI est interrompue pendant la surveillance de l'événement, si la surveillance est demandée.

Le message SelectRoute demande à l'entité SSF/CCF de reprendre le traitement d'origine de l'appel en tenant compte de l'adresse et des informations d'acheminement d'appel fournies dans les paramètres du message. Le traitement d'appel reprend au point de traitement d'appel ROUTING & ALERTING.

3.3.44.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.44.3 Entité de réponse (SSF)

3.3.44.3.1 Procédure normale

Préconditions de l'entité SSF:

- une demande de point TDP ou EDP a été invoquée.

Postconditions de l'entité SSF:

- l'entité SSF effectue les actions de traitement d'appel afin de sélectionner la voie d'acheminement.

L'entité SSF/CCF reprendra le traitement d'appel au point de traitement d'appel ROUTING & ALERTING et procédera selon la description de l'automate BCSM pour essayer d'achever l'appel par la voie d'acheminement désignée.

Les spécifications supplémentaires suivantes s'appliquent également au message SelectRoute:

- le choix de la voie d'acheminement/d'exploitant comme cela est indiqué dans SelectRoute doit s'effectuer jusqu'à ce qu'un service complémentaire libre soit trouvé ou jusqu'à ce que le traitement de l'appel rencontre un dispositif de filtrage bloquant l'appel (spécifique à l'exploitant de réseau);
- si une voie d'acheminement exige un TCM, la valeur du paramètre TCM devra être utilisée. Si un TCM est exigé et si le paramètre TCM n'est pas fourni dans SelectRoute, l'entité SSF/CCF déduira le TCM selon les exigences qui s'appliquent en supposant qu'il n'y ait pas d'implication du RI dans l'appel.

SelectRoute et O_Called_Party_Busy, O_No_Answer:

- quand l'entité SSF/CCF reçoit le message SelectRoute en réponse au message O_Called_Party_Busy, l'entité SSF/CCF entreprendra les actions suivantes:
 - 1) l'entité SSF/CCF devra libérer toutes les ressources qui ont été utilisées pour traiter l'appel entre les points de traitement d'appel ANALYSE_INFORMATION et ROUTING & ALERTING;
 - 2) l'entité SSF/CCF devra reprendre le traitement d'appel au point de traitement ROUTING & ALERTING, et traiter le message comme cela est décrit dans cette description de procédure;
 - 3) si l'accès au départ est DSS 1, l'entité SSF/CCF ne devra pas envoyer un autre message. (L'interfonctionnement DSS 1 du sous-protocole INAP fait l'objet d'un complément d'étude),
- quand le message SelectRoute est reçu en réponse à un message de demande de point TDP O_No_Answer, l'entité SSF/CCF effectuera les actions suivantes:
 - 1) l'entité SSF/CCF, si elle engendre une tonalité audible au demandeur, devra arrêter cette tonalité;
 - 2) l'entité SSF/CCF libérera toutes les ressources qui ont été utilisées pour traiter l'appel entre les points de traitement d'appel ANALYSE_INFORMATION et ROUTING & ALERTING;
 - 3) l'entité SSF/CCF reprendra le traitement d'appel au point de traitement d'appel ROUTING & ALERTING, et traitera le message comme cela est décrit dans cette description de procédure;
 - 4) si l'accès de départ est DSS 1, l'entité SSF/CCF ne devra pas envoyer un autre message CALL PROCEEDING. (L'interfonctionnement DSS 1 du sous-protocole INAP fait l'objet d'un complément d'étude),
- quand le message SelectRoute est reçu en réponse à un message de demande de point EDP O_No_Answer, l'entité SSF/CCF effectuera les actions suivantes:
 - 1) l'entité SSF/CCF, si elle engendre une tonalité audible au demandeur, devra arrêter cette tonalité;
 - 2) l'entité SSF/CCF reprendra le traitement de l'appel au point de traitement ROUTING & ALERTING, et traitera le message comme cela est décrit dans la présente Recommandation;
 - 3) si l'accès d'origine est DSS 1, l'entité SSF/CCF ne devra pas envoyer un autre message CALL PROCEEDING (L'interfonctionnement INAP-DSS 1 fait l'objet d'un complément d'étude).

SelectRoute avec OriginalCalledPartyID:

- quand l'entité SSF/CCF reçoit un message SelectRoute qui contient le paramètre OriginalCalledParty ID, l'entité SSF/CCF devra établir des correspondances par exemple vers le paramètre OriginalCalledPartyID reçu (l'interfonctionnement du sous-système ISUP fait l'objet d'un complément d'étude):
 - 1) l'élément d'information OriginalCalledNumber dans l'élément d'information de service complémentaire si l'entité SSF/CCF n'a pas encore de valeur pour OriginalCalledNumber et si l'entité SSF/CCF achemine l'appel vers une ligne du réseau RNIS (pour la signalisation RNIS de l'accès de base et de l'accès primaire);
 - 2) le paramètre OriginalCalledNumber dans le message d'adresse initial si l'entité SSF/CCF n'a pas encore de valeur pour le paramètre OriginalCalledNumber et si l'entité SSF/CCF achemine l'appel vers une ligne de réseau sémaphore (pour la signalisation du sous-système utilisateur du réseau RNIS).

3.3.44.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.45 Procédure SendChargingInformation

3.3.45.1 Description générale

Cette opération est utilisée pour donner l'ordre à l'entité SSF d'envoyer l'information de taxation par l'intermédiaire de l'entité SCF. L'envoi des informations de taxation peut être effectué soit par l'envoi des impulsions de taxation, soit par une signalisation ou de façon interne, si l'entité SSF se trouve dans l'autocommutateur local. Dans l'autocommutateur local, soit un compteur de taxation est mis à jour soit un graphique normal d'appel créé. Il existe une possibilité que l'opération SendChargingInformation (SCI) soit invoquée à plusieurs reprises. Le scénario de taxation à la base de cette opération est 3.2 (il convient de se référer à l'Appendice II/Q.1214 "Scénarios de taxation").

NOTE – L'interfonctionnement entre l'entité SSF et le RTPC est spécifique à l'exploitant de réseau. Cette opération comporte beaucoup d'interactions entre le RTPC et le RI.

3.3.45.1.1 Paramètres

- sCIBillingChargingCharacteristics (Caractéristiques de facturation/taxation):
ce paramètre indique des caractéristiques de facturation et/ou de taxation). Son contenu est spécifique à l'exploitant de réseau. Selon le scénario de taxation applicable, les éléments suivants d'information peuvent être inclus (il convient de se référer à l'Appendice II/Q.1214 "Scénarios de taxation"):
 - charge level (niveau de taxation) (scénario 3.2);
 - chargePulses (impulsions de taxation);
 - chargeMessages (messages de taxation);
- partyToCharge (Correspondant devant être taxé):
ce paramètre indique si les informations de taxation doivent être envoyées.

3.3.45.2 Entité d'appel (SCF)

3.3.45.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) une relation de commande existe entre l'entité SCF et l'entité SSF;
- 2) une instance SLPI a déterminé qu'une opération "SendChargingInformation" doit être envoyée par l'entité SCF;

Postconditions de l'entité SCF:

- 1) pas de transition d'état de l'entité FSM;
- 2) l'exécution par l'instance SLPI peut se poursuivre.

L'automate SCSM-FSM est dans l'état "Préparer l'instruction SSF" ou dans l'état "Mise en file d'attente dans l'entité FSM". La procédure SendChargingInformation sera invoquée par l'entité SCF pour demander l'exécution conformément aux demandes d'informations de taxation présentées par l'instance SLPI. Le cas échéant, cette information sera renvoyée en arrière sur le circuit d'appel.

Cela ne provoque pas de transition d'état dans l'automate SCSM-FSM.

3.3.45.2.2 Traitement des erreurs

Le traitement générique des erreurs d'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.45.3 Entité de réponse (SSF)

3.3.45.3.1 Procédure normale

Préconditions de l'entité SSF:

- SSF-FSM: état c: "Attente des instructions"; ou
SSF-FSM: état d: "Attente de la fin de l'interaction avec l'utilisateur"; ou
SSF-FSM: état e: "Attente de la fin de la connexion temporaire"; ou

SSF-FSM: état f: "Surveillance"; ou

Entité SSF-FSM d'assistance/de transfert de tâche état b: "Attente des instructions".

Postcondition de l'entité SSF:

- aucune transition d'état de l'entité FSM.

A la réception de cette opération, l'entité SSF effectue des actions destinées à envoyer les informations de taxation. L'envoi des informations de taxation peut être effectué soit sous forme d'impulsions de taxation, soit par l'envoi d'une signalisation ou de façon interne, si l'entité SSF se trouve dans l'autocommutateur local. Dans l'autocommutateur local, le compteur de taxation peut être enregistré ou un graphique standard d'appel produit. L'interfonctionnement entre l'entité SSF et le RTPC est spécifique à l'exploitant de réseau. Cette opération a de nombreuses interactions entre le réseau RTPC et le réseau intelligent.

Par exemple, lors de l'envoi d'une opération "SendChargingInformation", l'entité SCF donne l'ordre à l'entité SSF de déclencher les fonctions de taxation RTPC/RNIS selon les informations données sur le niveau de taxation à appliquer.

Le niveau de taxation peut être déterminé par l'une des fonctions suivantes:

- a) l'entité SCF; ou
- b) l'entité SSF; ou
- c) la fonction de taxation dans un autocommutateur suivant.

Dans le cas où l'entité SCF a déterminé le niveau de taxation, l'opération "SendChargingInformation" contient le niveau de taxation devant être appliqué.

Dans le cas où l'entité SSF détermine le niveau de taxation, l'opération "SendChargingInformation" contient les paramètres destinés à déterminer le niveau de taxation.

Si le niveau de taxation a été déterminé par le réseau intelligent (l'entité SCF ou SSF), l'entité SSF indique le niveau de taxation devant être appliqué aux fonctions de taxation du RTPC/RNIS [cas a) et b)].

Dans le cas c), le niveau de taxation est déterminé dans un autocommutateur suivant. L'opération "SendChargingInformation" contient soit les paramètres correspondants indiquant ce fait ou l'entité SSF détecte pendant la tentative de sélection du niveau de taxation à partir des paramètres fournis par l'entité SCF que le niveau de taxation doit être déterminé dans un autocommutateur suivant. Sur la base des capacités RTPC/RNIS déjà existantes, l'entité SSF fournit les fonctions de taxation RTPC/RNIS avec les informations nécessaires et les messages de taxation vers l'arrière seront transférés vers le bas du trajet de circuit quand l'entité SCF le permettra (ce qui sera produit par un autocommutateur suivant, par exemple une passerelle internationale entre les réseaux).

Dans le scénario décrit ci-dessus, la taxation/la facturation est exécutée au moyen des mécanismes existants du RTPC/RNIS et commandée par le réseau intelligent.

Cela signifie que la détermination de la méthode de taxation - en ligne ou en différé - et la taxation des articles auront lieu dans le réseau de base, comme le déclenchement de la taxation et l'enregistrement de la taxation.

3.3.45.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits au 3.4.

3.3.46 Procédure ServiceFilteringResponse

3.3.46.1 Description générale

Cette opération est utilisée pour rendre compte des valeurs de compteurs spécifiées dans une opération envoyée préalablement "ActivateServiceFiltering" en direction de l'entité SCF.

3.3.46.1.1 Paramètres

- countersValue:
ce paramètre contient le décompte des appels filtrés durant la période de filtrage. C'est une liste des identifications de compteur et des valeurs afférentes.
- filteringCriteria:
ce paramètre est utilisé pour traiter la logique de traitement de service concernée dans l'entité SCF.

- responseCondition:

ce paramètre est utilisé pour identifier la raison de l'envoi de la réponse ServiceFilteringResponse.

intermediateResponse indique que le filtrage de service est actif, qu'un appel a été reçu et que le compteur d'intervalle de temps s'est arrêté; ou

que le filtrage de service est actif et que la valeur de seuil, "numberOfCalls", est atteinte.

lastResponse indique que la durée est expirée et que le filtrage de service est arrêté, ou que le moment de l'arrêt est atteint et que le filtrage de service est arrêté.

3.3.46.2 Entité d'appel (SSF)

3.3.46.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) le filtrage du service est en cours et l'intervalle de temps est arrivé à expiration et un appel est reçu; ou
- 2) le filtrage du service est en cours et la valeur de seuil est atteinte; ou
- 3) le filtrage du service s'est achevé (expiration de la durée ou heure d'arrêt du filtrage); ou
- 4) l'opération "ActivateServiceFiltering" est reçue et rencontre une entité de filtrage du service activée.

Postcondition de l'entité SSF:

- le filtrage du service se poursuit ou se termine selon la durée.

L'entité SSF envoie l'opération "ServiceFilteringResponse" à l'entité SCF. Le paramètre "filteringCriteria" est fourni pour permettre l'adressage de la logique de traitement de service concernée au niveau de l'entité SCF.

Avant que l'opération "ServiceFilteringResponse" ne soit envoyée, on vérifiera si les critères d'espacement d'appel sont satisfaits. Dans ce cas, la réponse "ServiceFilteringResponse" n'est pas envoyée et le comptage se poursuit sans remise à zéro des compteurs. La dernière réponse "ServiceFilteringResponse" (heure d'arrêt fixée ou expiration de la durée de filtrage) est envoyée sans vérifier les critères de séquençement des appels.

Après l'envoi de la réponse "ServiceFilteringResponse", les compteurs de filtrage du service sont remis à zéro.

Si le filtrage du service se poursuit après l'envoi de "ServiceFilteringResponse" (c'est-à-dire l'expiration de l'intervalle de temps), l'automate SSME-FSM reste dans l'état "Traitement non associé à l'appel".

Si le filtrage du service s'est arrêté après l'envoi de "ServiceFilteringResponse" (c'est-à-dire l'expiration de l'intervalle de temps ou heure d'arrêt de filtrage atteinte), l'automate SSME-FSM reste dans l'état "Idle Management". Toutes les ressources affectées sont libérées, c'est-à-dire que l'automate SSME-FSM est également libéré.

3.3.46.2.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.46.3 Entité de réponse (SCF)

3.3.46.3.1 Procédure normale

Préconditions de l'entité SCF:

- 1) le filtrage du service est en cours;
- 2) l'entité SCME est dans l'état "Attente de la réponse de filtrage du service".

Postcondition de l'entité SCF:

- l'entité SCME achemine les valeurs de compteur reçues vers l'instance SLPI.

L'opération est traitée par la partie de la machine FSM de Filtrage du service de l'entité de gestion de la machine SCF (SCME). L'entité SCME transmet les valeurs de compteur reçues à l'instance SLPI où elles sont ajoutées à des valeurs de compteur qui ont été préalablement reçues.

Le paramètre "filteringCriteria" comme cela est indiqué dans le "ServiceFilteringResponse" est utilisé pour accéder à l'entité SCME et l'instance de traitement de logique de service concernée.

La machine FSM de filtrage du service de l'entité SCME reste à l'état "Attente de la réponse de filtrage du service de l'entité SSF" jusqu'à ce que la durée du filtrage du service interne dans l'instance SLPI soit atteinte. Puis, l'instance SLPI informe l'entité SCME de l'expiration de la temporisation. A présent, l'entité SCME passe à l'état "Filtrage du service au Repos".

3.3.46.3.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.47 Procédure SpecializedResourceReport

3.3.47.1 Description générale

Cette opération est utilisée comme réponse à l'opération "PlayAnnouncement" quand l'indication de fin d'annonce est donnée.

3.3.47.1.1 Paramètres

- aucun.

3.3.47.2 Entité d'appel (SRF)

3.3.47.2.1 Procédure normale

Préconditions de l'entité SRF:

- 1) l'automate SRSF-FSM est dans l'état "Interaction avec l'utilisateur";
- 2) une opération "PlayAnnouncement" pour laquelle le paramètre "RequestAnnouncementComplete" a été mis sur VRAI est en cours d'exécution;
- 3) toutes les informations ont été envoyées vers l'usager.

Postconditions de l'entité SRF:

- 1) l'automate SRSF-FSM reste dans le même état;
- 2) si le paramètre "DisconnectFromIPForbidden" était sur FAUX, le modèle SRSF déclenche une séquence de déconnexion de canal support vers l'entité SSF en utilisant le système de signalisation de canal support approprié, après avoir envoyé l'opération "SpecializedResourceReport" vers l'entité SCF. L'automate SRSF-FSM va à l'état "Repos".

3.3.47.2.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.47.3 Entité de réponse (SCF)

3.3.47.3.1 Procédure normale

Précondition de l'entité SCF:

- l'automate SCSF-FSM est dans l'état "Interaction avec l'utilisateur", sous-état "Attente de la réponse en provenance de l'entité SRF".

Postconditions de l'entité SCF:

- 1) l'automate SCSF-FSM est dans l'état "Interaction avec l'utilisateur", sous-état "Attente de la réponse en provenance de l'entité SRF";
- 2) si l'opération "SpecializedResourceReport" concerne une opération "PlayAnnouncement" pour laquelle la libération d'appel par l'entité SRF est permise, l'automate SCSF-FSM va à l'état "Préparer les instructions SSF".

3.3.47.3.2 Traitement des erreurs

Le traitement des erreurs relatives à l'opération n'est pas applicable, en raison de l'opération de classe 4.

3.3.48 Procédure StatusReport (Compte rendu d'état)

3.3.48.1 Description générale

Cette opération est utilisée pour notifier à l'entité SCF le résultat de la surveillance qui est demandé par l'opération "RequestFirstStatusMatchReport" ou "RequestEveryStatusChangeReport" à l'entité SCF.

3.3.48.1.1 Paramètres

- resourceStatus (Etat des ressources):
ce paramètre indique dans quel état se trouve une ressource physique d'extrémité pour laquelle l'entité SCF demande la surveillance.
- CorrelationID:
utilisé par l'entité SCF pour la corrélation avec une opération précédente.
- ResourceID:
- ReportCondition:
spécifie la raison pour laquelle cette opération est présentée (compte rendu d'état normal ou expiration de la durée de surveillance).

3.3.48.2 Entité d'appel (SSF)

3.3.48.2.1 Procédure normale

Précondition de l'entité SSF:

- l'entité SSME est dans l'état "Non_Call_Associated_Treatments" (Traitements non associés à l'appel).

Postcondition de l'entité SSF:

- l'entité SSME est dans l'un des états suivants:
état ma : "IdleManagement"
état mb: "Non_Call_Associated_Treatments" (Traitements non associés à l'appel).

L'entité SSF envoie l'opération "StatusReport" à l'entité SCF, quand les événements suivants se produisent.

- l'entité SSF trouve le changement d'état de la ressource dans un état spécifique;
- l'entité SSF a reçu l'opération "CancelStatusReportRequest" en provenance de l'entité SCF;
- l'expiration de la temporisation s'est produite dans l'entité SSF.

Après que l'entité SSF ait envoyé cette opération, l'entité SSME se place dans l'état "Idle Management" (Gestion inactive) à moins qu'il y ait d'autres traitements de l'opération "Non-Call-Associated", auquel cas l'entité SSF resterait dans "Non_Call_AssociatedTreatments" (Traitements non associés à un appel).

La clarification sur l'usage de cette opération à l'intérieur ou à l'extérieur du contexte d'un appel est à l'étude.

3.3.48.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des opérations sont décrits en 3.4.

3.3.48.3 Entité de réponse (SCF)

3.3.48.3.1 Procédure normale

Préconditions de l'entité SCF:

- l'entité SCME est dans l'état "Attente du compte rendu d'état relatif à la réponse de l'entité SSF".

Postconditions de l'entité SCF:

- 1) l'entité SCME est dans l'état "Compte rendu d'état Repos";
- 2) l'entité SCF notifie à l'instance SLPI le résultat de la surveillance des ressources dans l'entité SSF.

A la réception de cette opération, l'instance SLPI qui attend cette opération continuera.

3.3.48.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.49 Procédure TAnswer

3.3.49.1 Description générale

Cette opération est envoyée de l'entité SSF vers l'entité SCF au point de détection T_Answer, après avoir détecté un état valide de déclenchement, ou afin d'indiquer un événement détecté par RequestReportBCSMEvent.

3.3.49.1.1 Paramètres

- dPSpecificCommonParameters:
 - calledPartyBusinessGroupID:
voir la Recommandation Q.1290.
 - calledPartySubaddress:
voir la Recommandation Q.931.
 - calledFacilityGroup:
voir la Recommandation Q.1290.
 - calledFacilityGroupMember:
voir la Recommandation Q.1290.

3.3.49.2 Entité d'appel (SSF)

3.3.49.2.1 Procédure normale

Préconditions de l'entité SSF (TDP):

- 1) appel entrant reçu d'un modèle BCSM de départ;
- 2) l'appel a été accepté et le correspondant final a répondu;
- 3) pour un point TDP, le séquençement des appels ou le filtrage du service ne s'applique pas au segment d'appel;
- 4) les critères du point de détection ont été satisfaits;
- 5) Pour un point TDP-R, il n'y a pas de relation de commande existante influençant le segment d'appel.

Préconditions de l'entité SSF (EDP):

- 1) pour un point EDP-R, il y a une relation de commande existante et le point EDP T_Answer est armé;
- 2) pour un point EDP-N, il y a une relation de surveillance ou de commande existante et le point EDP T_Answer est armé.

Postconditions de l'entité SSF (TDP):

- 1) pour un point TDP-R, le traitement d'appel de base a été interrompu au point de détection T_Answer, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel de base est effectué au point de traitement d'appel T_Answer et aucune relation de commande n'a été établie.

Postconditions de l'entité SSF (EDP):

- 1) l'automate SSF-FSM reste dans l'état "Surveillance" si le type de message était la notification et il y a encore des points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée;
- 2) l'automate SSF-FSM se place dans l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés, aucune opération "CallInformationReport" ou "ApplyChargingReport" n'étant demandée;
- 3) l'automate SSF-FSM se place dans l'état "Attente des instructions" si le type de message était la demande. Le traitement d'appel est interrompu.

3.3.49.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.49.3 Entité de réponse (SCF)

Préconditions de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de surveillance existante est en place et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute et met fin au traitement.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM est dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et il y a encore des points EDP armés ou une information de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se place à l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM se place dans l'état "Préparation des instructions SSF" si le type de message était la demande.

3.3.49.3.1 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.50 Procédure TBusy

3.3.50.1 Description générale

Cette opération est envoyée par l'entité SSF vers l'entité SCF après détection d'une condition valide de déclenchement au point de détection TBusy, ou pour rendre compte d'un événement demandé. Il convient de se référer au 4.2.2.2/Q.1214 pour la sémantique additionnelle relative à la simulation d'appel par modèle.

3.3.50.1.1 Paramètres

- serviceAddressInformation (serviceKey, miscCallInfo, triggerType):
voir la Recommandation Q.1290. Le paramètre triggerType indique l'événement particulier qui a provoqué l'interruption de l'appel. Le paramètre miscCallInfo indique qu'une demande d'instructions a été présentée à l'entité SCF, de même que la catégorie de déclenchement (ligne, groupe, entreprise). Le paramètre serviceKey est utilisé pour identifier la logique de traitement de service devant être invoquée.
- bearerCapability:
ce paramètre indique à l'utilisateur le type de connexion de capacité support. Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage applicables au paramètre de capacité support.
- calledPartyNumber:
voir la Recommandation Q.762 pour les informations de signalisation du numéro de demandé. Ce paramètre est utilisé pour identifier le demandé dans le sens en avant.

- callingPartyNumber:
voir la Recommandation Q.762 "Information de signalisation du numéro de demandeur". Il convient de se référer au 6.4.4/Q.1214 relative aux règles de codage du paramètre du numéro de demandeur.
- callingPartysCategory:
voir la Recommandation Q.762 "Information de signalisation de la catégorie de demandeur". Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre de catégorie de demandeur.
- iPSSPCapabilities:
voir la Recommandation Q.1290.
- iPAvailable:
voir la Recommandation Q.1290.
- iSDNAccessRelatedInformation:
ce paramètre contient des éléments d'information (pouvant être multiples) conformes à la Recommandation Q.931. Voir les informations de signalisation du paramètre d'Accès au Transport dans les Recommandations Q.762, Q.763 et Q.931. Il convient de se référer au 6.4.4/Q.1214 (Informations analysées) pour les règles de codage applicables aux informations relatives à l'accès au RNIS.
- cGEncountered:
voir la Recommandation Q.1290.
- locationNumber:
voir la Recommandation Q.762 relative aux informations de signalisation du numéro d'emplacement. Ce paramètre est utilisé pour transporter l'adresse de zone géographique pour les services mobiles. Il est utilisé quand le "callingPartyNumber" ne contient pas d'informations sur l'emplacement géographique du demandeur (c'est-à-dire pour un acheminement dépendant de l'origine quand le demandeur est un abonné mobile).
- serviceProfileIdentifier:
voir l'Annexe A/Q.932. Voir 6.4.4/Q.1214 pour les règles de codage de serviceProfileIdentifier.
- terminalType:
voir la Recommandation Q.1290. Identifie le type de terminal, pour que l'entité SCF puisse spécifier à l'entité SRF, le type approprié de capacité (reconnaissance de la parole, DTMF, capacité de visualisation, etc.).
- chargeNumber:
voir la Recommandation Q.1290.
- servingArea ID:
voir la Recommandation Q.1290.
- busyCause:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 relative aux règles de codage.
- calledPartyBusinessPartyGroupID:
voir la Recommandation Q.1290.
- calledPartySubaddress:
voir la Recommandation Q.931.
- originalCalledPartyID:
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.
- redirectingPartyID
Ce paramètre (s'il est disponible) est le numéro d'annuaire du dernier réachemineur.
- redirectionInformation:
voir la Recommandation Q.763 Informations de signalisation du réacheminement.

- routeList:
voir la Recommandation Q.1290.
- travellingClassMark:
voir la Recommandation Q.1290.

3.3.50.2 Entité d'appel (SSF)

3.3.50.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) une tentative d'appel au départ a eu lieu;
- 2) le numéro de demandé est disponible et la nature de l'adresse déterminée;
- 3) l'espacement des appels ou le filtrage du service ne s'appliquent pas au segment d'appel;
- 4) les critères du point de détection ont été satisfaits;
- 5) pour un point TDP-R, il n'y a pas de relation de commande existante influençant le segment d'appel.

Postconditions de l'entité SSF:

- 1) pour un point TDP-R, le traitement de base de l'appel a été interrompu au point de détection T_Busy, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel de base passe par le point T_Exception et aucune relation de commande n'a été établie;
- 3) pour un point EDP, on utilise la même procédure que pour EventReportBCSM (voir § 3.3.22.2.1).

L'entité SSF a suffisamment d'informations disponibles associées au segment d'appel final. L'entité SSF devra détecter le point T_Busy quand le réseau déterminera que l'accès usager terminal est occupé. Les critères permettant la détection de l'occupation de l'accès usager déterminé par le réseau dépend du type d'accès terminal et des services auxquels l'utilisateur a souscrit (ligne d'accès analogique, DSS1, groupe de faisceau de recherche multiligne, etc.).

3.3.50.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.50.3 Entité de réponse (SCF)

3.3.50.3.1 Procédure normale

Préconditions de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie, et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute et met fin au traitement.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de demande était la notification et il y a encore des points EDP armés, ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se place dans l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM se place dans l'état "Préparer les instructions SSF" si le type de message était la demande.

3.3.50.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.51 Procédure TDisconnect

3.5.51.1 Description générale

Cette opération est envoyée par l'entité SSF vers l'entité SCF après qu'une condition valide de déclenchement ait été détectée au point de détection T_Disconnect ou afin de signaler un événement indiqué par RequestReport BCSMEvent. Il convient de se référer au 4.2.2.2/Q.1214 pour la sémantique additionnelle relative à la simulation d'appel par modèle.

3.3.51.1.1 Paramètres

- serviceAddressInformation (serviceKey, miscCallInfo, triggerType):
voir la Recommandation Q.1290. Le paramètre triggerType indique l'événement particulier qui a provoqué l'interruption de l'appel. Le paramètre miscCallInfo indique qu'une demande d'instructions ou une notification a été présentée à l'entité SCF, de même que la catégorie de déclenchement (ligne, groupe, entreprise). Le paramètre serviceKey est utilisé pour identifier la logique de traitement de service devant être invoquée.
- bearerCapability:
ce paramètre indique à l'utilisateur le type de connexion de capacité support. Se référer au 6.4.4/Q.1214 pour les règles de codage applicables au paramètre de capacité support.
- calledPartyNumber:
voir la Recommandation Q.762 pour les informations de signalisation du numéro de demandé. Ce paramètre est utilisé pour identifier le demandé dans le sens en avant.
- callingPartyNumber:
Voir la Recommandation Q.762 "Information de signalisation du numéro de demandeur". Il convient de se référer au 6.4.4/Q.1214 relative aux règles de codage du paramètre du numéro de demandeur.
- callingPartysCategory:
voir la Recommandation Q.762 "Information de signalisation de la catégorie de demandeur". Il convient de se référer au 6.4.4/Q.1214 pour les règles de codage du paramètre de catégorie de demandeur.
- iPSSPCapabilities:
voir la Recommandation Q.1290.
- iPAvailable:
voir la Recommandation Q.1290.
- iSDNAccessRelatedInformation:
ce paramètre contient des éléments d'information (pouvant être multiples) selon la Recommandation Q.931. Voir les informations de signalisation du paramètre du Transport à l'accès. Il convient de se référer au 6.4.4/Q.1214 (Informations analysées) pour les règles de codage applicables aux informations relatives à l'accès RNIS.
- cGEncountered:
voir la Recommandation Q.1290.

- locationNumber:
voir la Recommandation Q.762 relative aux informations de signalisation du numéro d'emplacement. Ce paramètre est utilisé pour transporter l'adresse de zone géographique pour les services mobiles. Il est utilisé quand le "callingPartyNumber" ne contient pas d'informations sur l'emplacement géographique du demandeur (c'est-à-dire pour un acheminement dépendant de l'origine quand le demandeur est un abonné mobile).
- serviceProfileIdentifier:
voir l'Annexe A/Q.932. Voir 6.4.4/Q.1214 pour les règles de codage de serviceProfileIdentifier.
- terminalType:
voir la Recommandation Q.1290. Identifie le type de terminal, pour que l'entité SCF puisse spécifier à l'entité SRF, le type approprié de capacité (reconnaissance de la parole, DTMF, capacité de visualisation, etc.).
- chargeNumber:
voir la Recommandation Q.1290. Voir 6.4.4/Q.1214 pour les règles de codage du paramètre "chargeNumber".
- servingArea ID:
voir la Recommandation Q.1290.
- calledPartyBusinessGroupID:
voir la Recommandation Q.1290.
- calledPartySubaddress:
voir la Recommandation Q.931 "Sous-adresse du demandeur".
- calledFacilityGroup:
voir la Recommandation Q.1290.
- calledFacilityGroupMember:
voir la Recommandation Q.1290.
- releaseCause:
indique la cause de la déconnexion.
- connectTime (Temps de connexion):
indique la durée entre l'indication de réponse reçue en provenance du demandé et la libération de la connexion.

3.3.51.2 Entité d'appel (SSF)

3.3.51.2.1 Procédures normales

Préconditions de l'entité SSF:

- 1) l'appel entrant est reçu en provenance de l'automate BCSM de départ;
- 2) l'appel a été accepté et le correspondant final a répondu;
- 3) l'indication de déconnexion reçue du correspondant final, ou reçue du correspondant de départ par l'intermédiaire du modèle BCSM de départ;
- 4) pour un point TDP, le séquençement des appels ou le filtrage du service n'est pas applicable;
- 5) les critères de point de détection ont été satisfaits;
- 6) pour un point TDP-R ou un point TDP-N, il n'y a pas de relation de commande existante;
- 7) pour un point EDP, il y a une relation de commande existante et le point EDP T_Disconnect est armé.

Postconditions de l'entité SSF:

- 1) pour un point TDP-R, le traitement d'appel de base a été interrompu au point de détection T_Disconnect, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel progresse vers le point de traitement T_Null & Authorize Termination Attempt, et aucune relation de commande n'a été établie;
- 3) pour un point EDP, la même procédure est applicable que pour EventReportBCSM (voir 3.3.22.2.1).

3.3.51.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.51.3 Entité de réponse (SCF)

3.3.51.3.1 Procédure normale

Préconditions de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours de traitement;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours de traitement.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est établie, et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute le traitement et y met fin.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM reste dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et il y a encore des points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se place dans l'état "Repos" si le type de message était la notification et s'il y a encore des points EDP armés, aucune opération qu'elle soit du type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM se place dans l'état "Préparer les instructions SSF" si le type de message était la demande.

3.3.51.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.52 Procédure TermAttemptAuthorized

3.3.52.1 Description générale

Cette opération est envoyée de l'entité SSF vers l'entité SCF au point de détection TermAttemptAuthorized, après la détection d'une condition valide de déclenchement, ou pour signaler un événement demandé par RequestReportBCSMEvent.

3.3.52.1.1 Paramètres

- dPSpecificCommonParameters:
- calledPartyBusinessGroupID:
voir la Recommandation Q.1290.
- calledPartySubaddress:
voir la Recommandation Q.931.
- callingPartyBusinessGroupID:
voir la Recommandation Q.1290.
- originalCalledPartyID:
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.

- `redirectingPartyID`:
ce paramètre contient le numéro d'annuaire du dernier réachemineur.
- `redirectionInformation`:
voir la Recommandation Q.763 Informations de signalisation du réacheminement.
- `routeList` (liste des acheminements):
`routeList` représente la liste des acheminements qui auraient dû être utilisés pour acheminer l'appel. Les exploitants de réseau peuvent spécifier que cet élément d'information devrait être utilisé si leur réseau particulier dispose des informations nécessaires.
- `travellingClassMark`:
voir la Recommandation Q.1290.

3.3.52.2 Entité d'appel

3.3.52.2.1 Procédure normale

Préconditions de l'entité SSF:

- 1) l'appel entrant est reçu;
- 2) l'autorisation d'acheminer l'appel vers des ressources d'extrémité spécifiées est vérifiée;
- 3) l'espacement des appels ou le filtrage du service ne s'appliquent pas;
- 4) les critères de point TDP ont été satisfaits;
- 5) pour un point TDP-R, il n'y a pas de relation de commande existante.

Postconditions de l'entité SSF:

- 1) pour un point TDP-R, un traitement d'appel de base a été interrompu au point de détection `Term_Attempt_Authorized`, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement d'appel de base est effectué au point de traitement `Select_Facility&Present Call`, et aucune relation de commande n'a été établie.

3.3.52.2.2 Traitement des erreurs

Si l'entité SCF de destination n'est pas accessible, alors le traitement final de l'appel est effectué (d'autres traitements font l'objet d'un complément d'étude). Si le demandeur interrompt l'appel après avoir envoyé l'opération `TermAttemptAuthorized`, l'entité SSF interrompt la relation de commande après que le premier message de réponse ait été reçu en provenance de l'entité SCF: l'identificateur de transaction est conservé en mémoire jusqu'à l'expiration de la temporisation `TSSF`.

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.52.3 Entité de réponse (SCF)

3.3.52.3.1 Procédure normale

Préconditions de l'entité SCF:

- 1) pour un point TDP, aucun;
- 2) le cas EDP ne s'applique pas.

Postconditions de l'entité SCF:

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une instruction SSF est en cours de préparation.

A la réception de l'opération `TermAttemptAuthorized`, le modèle SCSM passe de l'état "Repos" à l'état "Préparer les Instructions SSF". Une relation de commande est créée avec l'entité SSF connexe. Une instance de programme de logique de traitement de service (SLPI) est invoquée pour le traitement de l'opération `TermAttemptAuthorized`. Au moyen de cette relation de commande, l'entité SCF peut influencer le traitement d'appel de base en conformité avec la logique de traitement de service invoquée. Les actions devant être exécutées dans l'instance SLPI dépendent des paramètres transmis par l'intermédiaire de cette opération et l'instance SLPI (par exemple le service de réseau intelligent demandé lui-même).

3.3.52.3.2 Traitement des erreurs

Si l'opération TermAttemptAuthorized est rejetée, le modèle SCSM reste dans le même état. La fonction de maintenance est informée et aucune instance SLPI n'est invoquée. Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.53 Procédure TNoAnswer

3.3.53.1 Description générale

Cette opération est envoyée de l'entité SSF vers l'entité SCF après la détection d'une condition valide de déclenchement au point de détection T_No_Answer, ou pour signaler un événement demandé par RequestReport BCSMEvent.

Cette opération demande à SSF/CCF d'envoyer un message T_No_Answer de TDP-Request quand il rencontrera un déclenchement T_No_Answer.

3.3.53.1.1 Paramètres

- dPSpecificCommonParameters:
- calledPartyBusinessGroupID:
voir la Recommandation Q.1290.
- calledPartySubaddress:
voir la Recommandation Q.931.
- calledFacilityGroup:
voir la Recommandation Q.1290.
- calledFacilityGroupMember:
voir la Recommandation Q.1290.
- originalCalledPartyID (Identificateur de demandé initial):
voir la Recommandation Q.762 Informations de signalisation du numéro de demandé initial.
- redirectingPartyID:
contient le numéro d'annuaire du dernier réacheminéur.
- redirectionInformation:
voir la Recommandation Q.763 Informations de signalisation de réacheminement.
- travellingClassMark:
voir la Recommandation Q.1290.

3.3.53.2 Entité d'appel (SSF)

3.3.53.2.1 Procédure normale

Préconditions de l'entité SSF (TDP):

- 1) l'appel entrant a été reçu;
- 2) le correspondant final n'a pas répondu à l'intérieur d'une période de temps donnée;
- 3) l'espacement des appels ou le filtrage du service ne s'appliquent pas;
- 4) les critères de point de détection ont été satisfaits;
- 5) pour un point TDP-R, il n'y a pas de relation de commande existante.

Préconditions de l'entité SSF (EDP):

- 1) pour un point EDP-R, il y a une relation de commande existante et le point EDP T_No_Answer est armé;
- 2) pour un point EDP-N, il y a une relation de surveillance ou de commande existante et le point EDP_T_No_Answer est armé.

Postconditions de l'entité SSF (TDP):

- 1) pour un point TDP-R, le traitement d'appel de base a été interrompu au point de détection T_No_Answer, et une relation de commande a été établie;
- 2) pour un point TDP-N, le traitement par défaut des exceptions a été fourni, et aucune relation de commande n'a été établie.

Postconditions de l'entité SSF (EDP):

- 1) l'automate SSF-FSM reste dans l'état "Surveillance" si le type de message était la notification et il y a encore des points EDP armés ou une opération de type "CallInformationReport" ou "ApplyChargingReport" demandée;
- 2) l'automate SSF-FSM se place dans l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés, aucune opération "CallInformationReport" ou "ApplyChargingReport" n'étant demandée;
- 3) l'automate SSF-FSM se place dans l'état "Attente des instructions" si le type de message était la demande. Le traitement d'appel est interrompu.

3.3.53.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.53.3 Entité de réponse (SCF)

3.3.53.3.1 Procédure normale

Préconditions de l'entité SCF (TDP):

- aucune.

Préconditions de l'entité SCF (EDP):

- 1) pour un point EDP-R, dans l'entité SSF, une relation de commande existante est en place et une instance SLPI est en cours d'exécution;
- 2) pour un point EDP-N, une relation de surveillance existante est en place et une instance SLPI est en cours d'exécution.

Postconditions de l'entité SCF (TDP):

- 1) une instance SLPI a été invoquée;
- 2) pour un point TDP-R, une relation de commande est en place et une instance SLPI a été invoquée;
- 3) pour un point TDP-R, une instruction SSF est en cours de préparation;
- 4) pour un point TDP-N, aucune relation n'est établie. Une instance SLPI a été invoquée, elle exécute et met fin au traitement d'appel.

Postconditions de l'entité SCF (EDP):

- pour un point EDP, l'automate SCSM-FSM est dans le sous-état "Attente de la notification ou de la demande" si le type de message était la notification et il y a encore des points EDP armés ou une information de type "CallInformationReport" ou "ApplyChargingReport" demandée; ou
l'automate SCSM-FSM se place à l'état "Repos" si le type de message était la notification et il n'y a plus de points EDP armés, aucune opération de type "CallInformationReport" ou "ApplyChargingReport" n'étant demandée; ou
l'automate SCSM-FSM se place dans l'état "Préparation des instructions SSF" si le type de message était la demande.

3.3.53.3.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP qui sont utilisés pour rendre compte des erreurs l'opération sont décrits en 3.4.

3.3.54 Procédure Unbind (Détachement)

3.3.54.1 Description générale

L'opération de "Détachement" contenue dans la Recommandation X.500 est utilisée par l'entité SDF pour mettre fin à une association authentifiée entre une entité SCF et une entité SDF au nom de l'utilisateur. Pour une description complète de l'opération "Unbind", il convient de se référer au 8.2/X.511.

3.3.54.1.1 Paramètres

Aucun.

3.3.54.2 Entité d'appel (SCF)

3.3.54.2.1 Procédure normale

Préconditions de l'entité SCF:

- SCSM: "Rattachement SDF".

Postconditions de l'entité SCF:

- SCSM: "Idle".

Le modèle SCSM a préalablement déclenché une opération permettant le Rattachement à l'annuaire de l'entité SDF. Il se trouve dans l'état "Rattachement SDF". La logique de traitement de service détermine qu'il doit être mis fin à l'accès authentifié à l'entité SDF. Elle présente une opération "Détachement" [(e8) Unbind_Request] qui provoque le retour de l'entité SCSM vers l'état "Idle".

3.3.54.2.2 Traitement des erreurs

L'opération "Détachement" n'a pas d'erreurs relatives à l'opération.

3.3.54.3 Entité de réponse (SDF)

3.3.54.3.1 Procédure normale

Préconditions de l'entité SDF:

- SDSM: "Rattachement SCF".

Postconditions de l'entité SDF:

- SDSM: "Idle".

Une opération de rattachement a été préalablement présentée et l'entité SDSM se trouve dans l'état "Rattachement SDF" en attendant une demande en provenance de l'entité SCF et/ou en exécutant une opération. La réception de l'opération Unbind (Détachement) provoque une transition vers l'état "Idle" avec la transition (E5) Unbind_from_SCF.

3.3.54.3.2 Traitement des erreurs

L'opération "Détachement" n'a pas les erreurs relatives à l'opération.

3.3.55 RequestEveryStatusChangeReport procedure

3.3.55.1 Description générale

Cette opération est utilisée pour demander à l'entité SSF de commencer à surveiller tous les changements de l'état d'occupation/de repos d'une extrémité particulière.

3.3.55.1.1 Paramètres

- resourceID:
ce paramètre indique que la ressource physique terminale qui est demandée par l'entité SCF doit être surveillée par l'entité SSF. Ce paramètre est l'un des paramètres lineID, facilityGroupID, facilityGroupMemberID, ou trunkGroupID.
- correlation ID:
ce paramètre est utilisé par l'entité SCF pour associer le "StatusReport" provenant de l'entité SSF à la demande dans l'entité SCF.
- monitorDuration:
ce paramètre indique la durée maximale de surveillance dans l'entité SSF.

3.3.55.2 Entité d'appel (SCF)

3.3.55.2.1 Procédure normale

Préconditions de l'entité SCF:

- 1) l'occurrence de l'utilisation du script de la logique de service a déterminé qu'une opération "Request Every Status Change Report" doit être envoyée;
- 2) le modèle SCME est dans l'état "compte rendu d'état Idle".

Postcondition de l'entité SCF:

- le modèle SCME est dans l'état "Attente du compte rendu d'état relatif à la réponse de l'entité SSF".

Quand l'occurrence de l'utilisation du script de logique de service demande la surveillance de tout état d'occupation/de repos d'une ressource physique de terminaison, l'entité SCF envoie "Request Every Status Change Report" à l'entité SSF pour surveiller tout changement d'état d'une ressource d'extrémité particulière. Puis, l'entité SCME passe à l'état "Attente du compte rendu d'état de ressources de l'entité SSF" à partir de l'état "compte rendu d'état Idle". Ensuite, dans le cas où l'entité SCF reçoit le résultat d'erreur de cette opération, l'entité SCME reste dans le même état. Quand l'entité SCF reçoit le résultat d'erreur de cette opération, l'entité SCME retourne à l'état "Compte rendu d'état Idle".

Quand l'entité SCF reçoit une opération de compte rendu d'état provenant de l'entité SSF, l'entité SCME reste dans l'état "Attente du compte rendu d'état de ressources de l'entité SSF". Sinon, quand l'entité SCF reçoit une opération "compte rendu d'état" avec le paramètre monitorCondition mis sur "timerExpired" ou sur "canceled" à partir de l'entité SSF, l'entité SCME retourne à l'état "compte rendu d'état Idle".

La clarification sur l'usage de cette opération à l'intérieur ou à l'extérieur du contexte d'un appel est à l'étude.

3.3.55.2.2 Traitement des erreurs

Le traitement générique des erreurs relatives à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.3.55.3 Entité de réponse (SFF)

3.3.55.3.1 Procédure normale

Préconditions de l'entité SSF:

- l'entité SSME est dans l'un des états suivants:
 - état ma : "IdleManagement";
 - état mb: "Traitement non associé à un appel".

Postcondition de l'entité SSF:

- l'entité SSME est dans l'un des états suivants:
 - état ma : "IdleManagement";
 - état mb: "Traitement non associé à un appel".

A la réception de cette opération, l'entité SSF commence à surveiller tout changement dans l'état d'occupation/de repos d'une ressource d'extrémité particulière. Si une erreur s'est produite (par exemple l'entité SSF peut ne pas trouver de ressource d'extrémité spécifique, alors l'entité SSF renvoie à l'entité SCF ReturnError, le résultat d'erreur de cette opération, avec le type approprié d'erreur.

L'entité SSF surveille en continu tout changement dans l'état d'occupation/de repos d'une ressource particulière de terminaison jusqu'à ce qu'un temporisateur qui est spécifié par le paramètre monitorDuration vienne à expiration. A chaque fois que l'entité SSF rencontre un changement d'état, l'entité SSF envoie l'opération "compte rendu d'état" à l'entité SCF avec le paramètre monitorCondition mis à la valeur "statusReport". Après l'envoi de cette opération, l'entité SSF devrait rester dans l'état "Traitement non associé à un appel". Si la fin de temporisation se produit, l'entité SSF envoie l'opération "StatusReport" à l'entité SCF avec le paramètre monitorCondition mis à la valeur "timerExpired". Sinon, quand l'entité SSF reçoit une opération "Annuler le compte rendu d'état" provenant de l'entité SCF, l'entité SSF envoie l'opération "compte rendu d'état" à l'entité SCF avec le paramètre monitorCondition mis à la valeur "Canceled". Après l'envoi de cette opération par l'entité SSF, l'entité SSME devrait passer à l'état "IdleManagement", à moins qu'il y ait d'autres traitements de l'opération Non associé à un appel, auquel cas l'entité SSF devrait rester dans l'état "Non-Call AssociatedTreatment" (traitement non associé à un appel).

3.3.55.3.2 Traitement des erreurs

Le traitement générique des erreurs liées à l'opération est décrit en 3.2 et les services TCAP utilisés pour rendre compte des erreurs d'opération sont décrits en 3.4.

3.4 Services assurés par le protocole d'application TCAP

Ce paragraphe décrit les procédures et les primitives TCAP qui devront être utilisés pour la transmission des messages entre les entités SSF, SCF et SRF. En ce qui concerne l'interface SCF-SDF, se référer au 2.2.2.2.

3.4.1 Procédures normales

Ce paragraphe décrit les procédures et les primitives TCAP qui devront être utilisées pour transmettre des messages entre les entités SSF, SCF et SRF dans des conditions normales d'exploitation.

Le protocole INAP, comme utilisateur TC, utilise uniquement une capacité de dialogue structurée fournie par le protocole d'application TCAP. Les situations suivantes peuvent se produire quand un message est envoyé entre deux entités physiques:

- a) un dialogue devra être établi – L'utilisateur TC présente une primitive de demande TC-BEGIN;
- b) un dialogue devra être maintenu – L'utilisateur TC présente une primitive de demande TC-CONTINUE;
- c) un dialogue ne devra plus être maintenu – L'utilisateur TC présente une primitive de demande TC-END soit avec une fin normale, soit avec une fin sur entente préalable selon les conditions suivantes:
 - i) fin normale:
 - au cas où le dialogue est établi, les opérations, entraînant la fin de la relation, peuvent être transmises par l'entité fonctionnelle avec une primitive de demande TC-END (de base) au cas où l'entité fonctionnelle n'est pas intéressée par la réception de composantes ERREUR ou de composantes REJET pour ces opérations envoyées. Une fois que les ressources de dialogue de l'entité fonctionnelle ont été libérées, toute composante ERREUR ou REJET reçue pour ces opérations sera écartée par le TC comme cela est décrit dans la Recommandation Q.774;
 - au cas où le dialogue est établi et où l'entité fonctionnelle a reçu une opération, entraînant une fin de relation de dialogue, le dialogue n'a plus de raison de se poursuivre et il n'y a pas d'opération à envoyer, une primitive de demande TC-END (de base) avec des composantes zéro pouvant être envoyées à partir de l'entité fonctionnelle,
 - ii) fin sur entente préalable:
 - au cas où une entité est intéressée par des messages possibles d'ERREUR ou de REJET en réponse à l'envoi d'opérations entraînant la fin de la relation, il est mis fin au dialogue par la primitive de demande TC-END (fin sur entente préalable) après la fin de la temporisation de la dernière opération associée. L'entité de réception peut finir le dialogue avec une primitive de demande TC-END (fin sur entente préalable) après le traitement réussi de ces opérations (c'est-à-dire que la relation est terminée);
- d) en général, l'utilisation de la fin sur entente préalable sera limitée au cas de deux entités en communication clairement identifiables par le fait que l'entité homologue applique une fin sur entente préalable. Dans tous les autres cas, on utilisera la fin normale;
- e) un dialogue ne sera pas établi – Pour les opérations de classe 2 ou 4, seul l'envoi de TC utilisateur délivre une primitive de demande TC-BEGIN et met fin localement au dialogue après la fin de la temporisation de l'opération au moyen d'une fin sur entente préalable. A la réception d'une primitive d'indication TC-BEGIN, l'utilisateur de TC récepteur mettra fin localement au dialogue.

3.4.1.1 Messages SSF-vers-SCF

3.4.1.1.1 Messages relatifs à la machine FSM de l'entité SSF

Un dialogue sera établi quand la machine FSM de l'entité SSF passera de l'état **Idle** à l'état **Attente des instructions**. L'opération du protocole d'application INAP applicable, pouvant être l'opération InitialDP ou une des opérations spécifiques au point TDP-R, devra être transmise dans le même message.

Aucun dialogue ne sera établi quand la machine FSM de l'entité SSF sortira de l'état **Idle** et reviendra à l'état **Idle** sur détection du point TDP-N. L'opération INAP applicable, pouvant être l'opération InitialDP ou une des opérations spécifiques au point de détection pour TDP-N, devra être envoyée avec une primitive de demande TC-BEGIN et il sera localement mis fin au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable.

Pour toutes les autres opérations envoyées par la machine FSM de l'entité SSF, le dialogue sera maintenu sauf dans les cas suivants.

Quand la machine FSM de l'entité SSF effectue une transition d'état du cas sans erreur à l'état **Idle** et quand il y a une ou plusieurs opérations en cours et quand le dialogue du sous-protocole d'application TCAP est établi, il peut être mis fin au dialogue du sous-protocole d'application TCAP par une primitive TC-END avec une composante ou des composantes. Quand l'entité SSF envoie le dernier EventReportBCSM, ApplyChargingReport ou CallInformationReport, il peut être mis fin au dialogue à partir de l'entité SSF par une primitive de demande TC-END avec fin normale.

Dans le cas où il n'y a pas d'opération en cours et où le dialogue du sous-protocole d'application TCAP est établi, il peut être mis fin au dialogue du sous-protocole d'application TCAP par une primitive TC-END avec une composante zéro ou une fin sur entente préalable. Quand la machine FSM de l'entité SSF transite dans le cas sans erreur vers l'état **Idle** et quand il n'y a pas d'opération à envoyer, il est mis fin au dialogue au moyen d'une primitive de demande TC-END (de base) avec composante zéro, ou bien il est localement mis fin au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable.

L'entité SSF peut mettre fin à un dialogue avec une primitive de demande TC-END avec une composante zéro ou bien il peut être mis fin au dialogue sur entente préalable, selon que le dialogue du sous-protocole d'application TCAP ait ou non été établi; dans le cas où la libération d'appel est déclenchée par toute autre entité que l'entité SCF et où l'entité SSF n'a pas de demandes d'information d'appel en cours (ni de demandes en cours qui doivent être traitées de façon identique, voir Note 1 de 3.1.1.5), ni aucun point EDP armé pour notifier la libération d'appel à l'entité SCF (pour une autre méthode, voir 3.4.2.2).

Quand l'entité SSF a envoyé le dernier EventReportBCSM, ApplyChargingReport ou CallInformationReport, il peut être mis fin au dialogue à partir de l'entité SCF ou par une primitive de demande TC-END avec fin normale.

3.4.1.1.2 Messages relatifs à la machine FSM de l'entité SSF d'assistance/de transfert de tâche

Un dialogue devra être établi quand la machine FSM de l'entité SSF d'assistance/de transfert de tâche passera de l'état **Idle** à l'état **Attente des instructions**. L'opération AssistRequestInstructions sera transmise vers une primitive de demande TC-BEGIN.

Pour toutes les autres opérations envoyées à partir de la machine FSM de l'entité SSF d'assistance/de transfert de tâche, le dialogue sera maintenu sauf dans les cas suivants:

Quand la machine FSM de l'entité SSF effectue une transition d'état vers l'état **Idle**, quand il y a une opération ou plusieurs opérations en cours, et quand le dialogue du sous-protocole d'application TCAP est établi, il peut être mis fin au dialogue du sous-protocole d'application TCAP par une primitive TC-END avec composante(s). Quand l'entité SSF envoie la dernière opération ApplyChargingReport, il peut être mis fin au dialogue à partir de l'entité SSF par une primitive de demande TC-END avec fin normale.

Dans le cas où il n'y a pas d'opération en cours et où le dialogue du sous-protocole d'application TCAP est établi, il peut être mis fin au dialogue du sous-protocole d'application TCAP par une primitive TC-END avec une composante zéro ou il peut être mis fin à ce dialogue sur entente préalable. Quand la machine FSM de l'entité SSF transite dans le cas sans erreur vers l'état **Idle** et qu'aucune opération ne doit être envoyée, il est localement mis fin au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable.

Une fois que l'entité SSF a envoyé la dernière opération ApplyChargingReport, il peut être mis fin au dialogue provenant de l'entité SCF par une primitive de demande TC-END avec fin normale.

3.4.1.1.3 Messages relatifs à la machine FSM de l'entité SSME

Il conviendra de se conformer aux procédures suivantes:

- le dialogue devra être maintenu quand ActivityTestReturnResult (retour d'essai d'activité) sera envoyé;
- aucun dialogue ne sera établi lorsque l'opération ServiceFilteringResponse sera envoyée. L'opération est envoyée par une primitive de demande TC-BEGIN et il est mis fin au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable;
- un dialogue ne sera plus maintenu quand le résultat de l'opération ActivateServiceFiltering sera envoyé. Il sera mis fin au dialogue au moyen d'une primitive de demande TC-END avec fin normale, le résultat étant transmis avec la même demande;
- il est mis fin localement au dialogue au moyen d'une primitive de demande TC-END sur entente préalable, à la réception d'une primitive d'indication TC-BEGIN avec une opération CallGap;
- le dialogue sera maintenu quand l'opération RequestCurrentStatusReport, RequestEveryStatusChangeReport ou RequestFirstStatusMatchReport sera reçue à l'intérieur du contexte d'appel;

- le dialogue sera maintenu lors de l'envoi des opérations suivantes à l'intérieur du contexte d'appel, si les opérations ne sont pas les opérations finales:
 - retour du résultat de l'opération RequestCurrentStatusReport;
 - retour du résultat de RequestFirstStatusMatchReport ou opération RequestEveryStatusChangeReport; et
 - opération StatusReport en réponse à RequestFirstStatusMatchReport ou Opération RequestEveryStatusChangeReport,
- le dialogue ne sera plus maintenu lors de l'envoi des opérations suivantes à l'intérieur du contexte d'appel si les opérations sont des opérations finales:
 - retour du résultat de l'opération RequestCurrentStatusReport;
 - retour du résultat de RequestFirstStatusMatchReport ou opération RequestEveryStatusChangeReport; et
 - opération StatusReport en réponse à RequestFirstStatusMatchReport ou opération RequestEveryStatusChangeReport,

il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive de demande TC-END avec fin normale, une des opérations ci-dessus étant transmise avec la même demande;

si la durée de surveillance se termine pour l'opération RequestFirstStatusMatchReport ou RequestEveryStatusChangeReport qui a été reçue à l'intérieur du contexte d'appel et s'il n'est pas nécessaire de maintenir ce dialogue, il est mis fin à ce dialogue à partir de l'entité SSF au moyen d'une primitive de demande TC-END (de base) avec composante zéro;

- le dialogue sera établi si l'une des opérations RequestCurrentStatusReport, RequestEveryStatusChangeReport ou RequestFirstStatusMatchReport est reçue à l'extérieur du contexte d'appel;
- le dialogue ne sera plus maintenu si le retour du résultat de l'opération RequestCurrentStatusReport est envoyé à l'extérieur du contexte d'appel. Il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive de demande TC-END avec fin normale, le retour du résultat étant transmis avec la même demande;
- le dialogue sera maintenu si le retour du résultat de l'opération RequestFirstStatusMatchReport ou RequestEveryStatusChangeReport est envoyé à l'extérieur du contexte d'appel;
- le dialogue ne sera plus maintenu si l'opération StatusReport est envoyée en réponse à l'opération RequestFirstStatusMatchReport à l'extérieur du contexte d'appel. Il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive de demande TC-END avec fin normale, l'opération StatusReport étant transmise avec la même demande;
- le dialogue sera maintenu si l'opération StatusReport est envoyée en réponse à l'opération RequestEveryStatusChangeReport à l'extérieur du contexte d'appel;
- le dialogue ne sera plus maintenu si la durée de surveillance se termine pour RequestFirstStatusMatchReport ou pour RequestEveryStatusChangeReport qui a été reçue à l'extérieur du contexte d'appel. Il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive de demande (de base) TC-END avec composante zéro.

3.4.1.2 Messages SCF-vers-SSF

3.4.1.2.1 Messages relatifs à la machine FSM du modèle SCSM

Aucun dialogue ne sera établi si la machine FSM du modèle SCSM passe de l'état **Repos** à l'état **Repos** à la réception d'une opération InitialDP ou d'une des opérations spécifiques au point de détection pour TDP-N. L'opération est reçue avec une primitive d'indication TC-BEGIN et il est mis fin localement au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable.

Un dialogue sera établi quand la machine FSM du modèle SCSM passe de l'état **Repos** à l'état **Préparer les instructions SSF** lors de la réception d'une opération InitialDP pour un point TDP-R, une des opérations spécifiques au point TDP-R ou une opération AssistRequestInstructions.

Un dialogue devra être établi si la machine FSM du modèle SCSM passe de l'état **Repos** à l'état **Préparer les instructions SSF** lors de l'envoi d'une opération InitiateCallAttempt à l'entité SSF.

Pour des opérations subséquentes envoyées à partir de la machine FSM du modèle SCSM, le dialogue sera maintenu sauf dans les cas suivants, c'est-à-dire que toutes les autres opérations seront envoyées après qu'un dialogue ait été établi à partir de l'entité SSF (l'entité SCF a préalablement reçu une primitive d'indication TC-BEGIN avec une opération InitialDP, une des opérations spécifiques au point DP ou une opération AssistRequestInstructions).

Le dialogue ne sera plus maintenu si la condition de fin sur entente préalable est satisfaite dans l'entité SCF. Si l'entité SCF n'attend plus d'autre message que tout autre message REJET ou ERREUR éventuel pour les entités envoyées et si la dernière temporisation d'opération associée vient à expiration, il sera localement mis fin au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable. Sinon, l'envoi des opérations, conduisant à la fin de la relation au moyen d'une primitive de demande TC-END (fin normale) est possible.

3.4.1.2.2 Messages relatifs à la machine FSM de l'entité SCME

Les opérations envoyées à partir de la machine FSM de l'entité SCME seront présentées selon les procédures suivantes:

- le dialogue sera maintenu quand l'opération ActivityTest aura été envoyée;
- un dialogue ne sera pas établi quand une opération CallGap sera envoyée sans utiliser un dialogue associé au modèle SCSM. L'opération sera envoyée en utilisant une primitive de demande TC-BEGIN et il sera mis fin au dialogue sur entente préalable;
- pour l'envoi d'une ou de plusieurs opérations CallGap, la machine FSM de l'entité SCME peut utiliser une machine FSM associée au modèle SCSM qui a été déclenchée par une machine FSM de l'entité SSF (c'est-à-dire établie pour la transmission d'une opération InitialDP ou une des opérations spécifiques au point de détection). Le dialogue sera établi et l'opération CallGap ou toutes les opérations CallGap devra (devront) être envoyée(s) avec la première réponse de la machine FSM du modèle SCSM à l'opération InitialDP ou une des opérations spécifiques au point de détection;
- un dialogue sera établi quand l'opération ActivateServiceFiltering sera envoyée. Cette opération sera transmise avec une primitive de demande TC-BEGIN;
- il est localement mis fin au dialogue dès la réception d'une opération ServiceFilteringResponse en utilisant une primitive de demande TC-END avec fin sur entente préalable;
- le dialogue sera maintenu lorsque l'opération RequestCurrentStatusReport, RequestEveryStatusChangeReport ou RequestFirstStatusMatchReport est envoyée à l'intérieur du contexte d'appel;
- le dialogue sera maintenu lors de la réception des opérations suivantes à l'intérieur du contexte d'appel si les opérations ne sont pas finales:
 - retour du résultat de l'opération RequestCurrentStatusReport;
 - retour du résultat de RequestFirstStatusMatchReport ou opération RequestEveryStatusChangeReport; et
 - opération StatusReport en réponse à RequestFirstStatusMatchReport ou Opération RequestEveryStatusChangeReport,
- le dialogue ne sera plus maintenu lors de la réception des opérations suivantes à l'intérieur du contexte d'appel si les opérations sont finales:
 - retour de résultat de l'opération RequestCurrentStatusReport;
 - retour de résultat de RequestFirstStatusMatchReport ou opération RequestEveryStatusChangeReport; et
 - opération StatusReport en réponse à RequestFirstStatusMatchReport ou opération RequestEveryStatusChangeReport,

il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive d'indication TC-END avec fin normale, l'une des opérations ci-dessus étant transmise avec la même demande;

si la durée de surveillance se termine pour l'opération RequestFirstStatusMatchReport ou l'opération RequestEveryStatusChangeReport qui a été reçue à l'intérieur du contexte d'appel et s'il n'est pas nécessaire de maintenir le dialogue, il sera mis fin à ce dialogue à partir de l'entité SSF au moyen d'une primitive d'indication TC-END (fin normale) avec composante zéro;

- le dialogue sera établi si l'une des opérations RequestCurrentStatusReport, RequestEveryStatusChangeReport ou RequestFirstStatusMatchReport est envoyée à l'extérieur du contexte d'appel;
- le dialogue ne sera plus maintenu si le retour du résultat de l'opération RequestCurrentStatusReport est reçu à l'extérieur du contexte d'appel. Il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive d'indication TC-END avec fin normale, le retour du résultat étant transmis avec la même indication;
- le dialogue sera maintenu si le retour du résultat de l'opération RequestFirstStatusMatchReport ou RequestEveryStatusChangeReport est reçu à l'extérieur du contexte d'appel;
- le dialogue ne sera plus maintenu si l'opération StatusReport est reçue en réponse à l'opération RequestFirstStatusMatchReport à l'extérieur du contexte d'appel. Il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive d'indication TC-END avec fin normale, l'opération Status Report étant reçue avec la même indication;
- le dialogue sera maintenu si l'opération StatusReport est reçue en réponse à l'opération RequestEveryStatusChangeReport à l'extérieur du contexte d'appel;
- le dialogue ne sera plus maintenu si la durée de surveillance prend fin pour l'opération RequestFirstStatusMatchReport ou RequestEveryStatusChangeReport à l'extérieur du contexte d'appel. Il est mis fin au dialogue à partir de l'entité SSF au moyen d'une primitive d'indication TC-END (de base) avec composante zéro.

3.4.1.3 Messages de l'entité SCF-vers/à partir de-SRF

Un dialogue est établi quand l'entité SRF envoie une opération AssistRequestInstructions vers l'entité SCF. Pour toutes les autres opérations envoyées vers/à partir de l'entité SRF, le dialogue sera maintenu.

Au cas où il n'y a pas d'opération en cours et où le dialogue du sous-protocole d'application TCAP est établi, il peut être mis fin au dialogue du sous-protocole d'application TCAP par une primitive TC-END avec composante zéro. Quand le modèle SCSM transite du cas sans erreur à l'interaction utilisateur, et quand il n'y a pas d'opération à envoyer, il est mis fin au dialogue au moyen d'une primitive de demande TC-END (fin normale) avec zéro composantes.

Le dialogue ne sera plus maintenu si l'envoi de l'opération SRReport pour PlayAnnouncement avec déconnexion à partir de l'entité SRF est mis à la valeur VRAI ou si le retour du résultat de PromptAndCollectUserInformation avec déconnexion à partir de l'entité SRF est également mis à la valeur VRAI. Il est mis fin aux dialogues au moyen d'une primitive de demande TC-END avec fin normale, et l'une des opérations ci-dessus est transmise avec la même demande.

Indifféremment du fait que l'opération existe ou n'existe pas, la machine FSM de l'entité SRSM est informée de la déconnexion de la connexion support (en cas de déconnexion déclenchée par l'entité SCF ou d'interruption d'appel en provenance de la ligne appelante) le dialogue est établi; il est mis fin au dialogue au moyen de primitive de demande TC-END (appel normal) avec zéro composantes ou une primitive de demande TC-END (fin sur entente préalable).

Le dialogue ne sera plus maintenu quand la condition de fin sur entente préalable est satisfaite dans l'entité SRF. Quand la machine FSM du modèle SRSM est informée de la déconnexion de la connexion support et quand le dialogue du sous-protocole d'application TCAP n'est pas établi, il est mis localement fin au dialogue au moyen de la primitive TC-END avec fin sur entente préalable.

Quand l'entité SCF n'attend pas des messages autres que les messages REJET ou ERREUR éventuels pour les opérations envoyées et à l'expiration de la dernière temporisation d'opération associée, il est mis fin localement au dialogue au moyen d'une primitive de demande TC-END avec fin sur entente préalable. Sinon, l'envoi des opérations, conduisant à la fin de la relation, au moyen d'une primitive de demande TC-END (fin normale), est possible.

Dans le cas du relais, la relation SRF-SCF utilise le dialogue du sous-protocole d'application TCAP de l'association SSF-SCF. Ceci est possible parce que le début et la fin de la relation SRF-SCF sont inclus dans la relation SSF-SCF. Les informations SRF-SCF devront être échangées avec les primitives de demande TC-CONTINUE.

Dans le cas du relais SSF, la façon d'établir les correspondances avec la capacité ROSE du système de signalisation support entre l'entité SSF et l'entité SRF et les services attendus de ROSE nécessitent un complément d'étude.

3.4.2 Procédures anormales

Ce paragraphe décrit les procédures et les primitives TCAP qui doivent être utilisées pour rendre compte des situations anormales entre les entités SSF, SCF et SRF. Les cas d'erreur sont définis en 3.2.

Les primitives suivantes devront être utilisées pour rendre compte des situations anormales:

- les erreurs d'opération telles qu'elles sont définies dans le protocole d'application INAP sont indiquées par la primitive de demande TC-U-ERROR;
- le rejet d'une composante TCAP par l'utilisateur TC sera indiqué par une primitive de demande TC-U-REJECT;
- quand l'entité fonctionnelle détectant une erreur ou rejetant l'opération décide qu'il sera mis fin au dialogue TC, la primitive de demande TC-END (de base) avec erreur ou rejet peut être utilisée pour mettre fin au dialogue TC;
- quand l'entité SSF ou SRF détectant l'erreur ou rejetant l'opération reconnaît la possibilité de continuer le dialogue, la primitive de demande TC-CONTINUE avec erreur ou rejet peut être utilisée pour continuer le dialogue TC;
- un dialogue sera interrompu par l'utilisateur TC avec une primitive de demande TC-U-ABORT;
- lorsque le temporisateur d'application T_{SSF} ou T_{SRF} vient à expiration, il devra être mis fin au dialogue au moyen d'une primitive de demande TC-U-ABORT avec un motif d'interruption, que le dialogue ait été ou non établi.

Dans les situations anormales détectées par le protocole d'application TCAP, les mêmes règles devront s'appliquer à la réception de l'indication TC-R-REJECT, à la transmission de la demande TC-U-REJECT, à la transmission de l'indication TC-P-ABORT comme à la transmission d'une primitive de demande TC-U-ABORT.

Les règles suivantes devront s'appliquer pour mettre fin au dialogue du sous-protocole d'application TCAP dans des situations anormales:

- au cas où la condition d'interruption est détectée et où le dialogue du sous-protocole d'application TCAP est établi, il est mis fin au dialogue du sous-protocole d'application TCAP par la primitive TC-U-ABORT avec une cause d'interruption;
- au cas où la condition d'interruption est détectée et où le dialogue du sous-protocole d'application TCAP n'est pas établi, il est localement mis fin au dialogue du sous-protocole d'application TCAP avec la primitive TC-U-ABORT (dans un cas comme la temporisation d'application).

Dans des situations d'erreur, on n'utilisera pas la fin sur entente préalable pour mettre fin au dialogue TCAP. Au cas où une entité d'application rencontrerait une situation d'erreur, l'entité homologue ne sera pas explicitement avertie de l'erreur, si cela est possible. Si à partir de quelque entité que ce soit, l'erreur rencontrée nécessite qu'il soit mis fin à la relation, il sera mis fin au dialogue par l'intermédiaire d'une primitive de demande TC-END avec fin normale ou par l'intermédiaire d'une primitive de demande TC-U-ABORT, selon que n'importe quelle composante ERREUR ou REJET en cours doive ou non être envoyée.

Dans le cas où une entité reçoit une primitive d'indication TC-END et une fois que toutes les composantes ont été examinées, la machine FSM n'est pas à même de mettre un terme à la relation, une erreur interne appropriée devrait être indiquée.

Dans les cas où il doit être mis fin à un dialogue par l'entité déclencheuse avant que l'établissement ait été achevé (avant que la première primitive d'indication TC ait été reçue à partir de l'entité de réponse), l'utilisateur TC présentera une primitive de demande TC-END avec fin sur entente préalable ou une primitive de demande TC-U-ABORT. Le résultat de ces primitives ne sera que local, toute indication TC subséquente reçue pour ce dialogue étant traitée selon les procédures anormales comme cela est spécifié dans la Recommandation Q.774.

3.4.2.1 Messages de l'entité SCF-vers-l'association SSF/SRF

Compte tenu du fait que les entités SSF et SRF ne disposent pas de la logique pour traiter les cas d'erreur détectés sur l'interface entre l'entité SCF et l'association SSF/SRF, la règle suivante est applicable:

- les erreurs d'opération et le rejet des composantes TCAP seront transmis vers l'entité SSF et, respectivement, l'entité SRF avec une primitive de demande TC-END, fin normale.

Si, dans le cas où la procédure ci-dessus n'est pas respectée, une composante ERREUR ou REJET est reçue avec une primitive d'indication TC-CONTINUE, l'entité SSF et, respectivement, l'entité SRF interrompent le dialogue avec une primitive de demande TC-U-ABORT.

Dans le cas du relais SSF, la façon d'établir les correspondances pour les messages vers la capacité ROSE du système de signalisation support entre l'entité SSF et l'entité SRF, de même que les services attendus de ROSE nécessitent un complément d'étude.

3.4.2.2 Messages de l'association SSF/SRF vers l'entité SSF

Les erreurs d'opération et le rejet des composantes TCAP seront transmis vers l'entité SCF selon les règles suivantes:

- le dialogue sera maintenu quand le message précédent, qui contenait la composante d'erreur, a indiqué que le dialogue serait maintenu, c'est-à-dire que l'erreur ou le rejet serait transmis avec une primitive de demande TC-CONTINUE si la composante d'erreur a été reçue avec une primitive d'indication TC-CONTINUE;
- à la réception d'une composante ERREUR ou REJET, l'entité SCF décidera de la poursuite du traitement. Elle peut soit continuer le dialogue, y mettre fin explicitement ou l'interrompre;
- dans toutes les autres situations, le dialogue ne sera plus maintenu, c'est-à-dire que la composante ERREUR ou REJET sera transmise avec une primitive de demande TC-END, fin normale, si la composante d'erreur a été reçue avec une primitive d'indication TC-BEGIN;
- à l'expiration de la temporisation T_{SSF} ou T_{SRF} , il sera mis fin au dialogue au moyen de la primitive TC-U-ABORT avec une cause d'interruption, que le dialogue avec le sous-protocole d'application TCAP ait été ou non établi.

Si le traitement d'erreur dans l'association SSF/SRF conduit au cas où l'association SSF/SRF n'est pas à même de poursuivre le traitement des opérations SCF alors que le dialogue doit être maintenu, l'association SSF/SRF interrompra le dialogue par une primitive de demande TC-END avec fin normale ou une primitive de demande TC-U-ABORT, selon qu'une composante ERREUR ou REJET en cours doive ou non être envoyée.

L'entité SSF peut mettre fin à un dialogue au moyen d'une primitive de demande TC-U-ABORT, au cas où la libération d'appel est déclenchée par toute autre entité et que les entités SCF et SSF n'ont pas de demandes d'informations d'appel en cours (ou de demandes en cours qui devraient être traitées de la même façon, c'est-à-dire ni l'opération ApplyCharging ni aucun point EDP armé pour notifier à l'entité SCF la libération d'appel (pour toute autre méthode, voir 3.4.1.1.1).

Dans le cas du relais de l'entité SSF, la façon de mapper les messages sur la capacité ROSE du système de signalisation support entre l'entité SSF et l'entité SRF, ainsi que les services attendus de l'élément ROSE appellent un complément d'étude.

3.4.3 Etablissement du dialogue

L'établissement d'un dialogue INAP implique deux processus d'application comme cela est décrit en 0.4, l'une étant l'entité appelante de dialogue et l'autre le répondeur de dialogue.

La négociation du contexte d'application ne peut pas être prise en charge dans toutes les entités physiques et/ou tous les réseaux;

Cette procédure est engendrée par les signaux suivants:

- une primitive de demande TC-BEGIN à partir de l'entité appelante de dialogue;
- une primitive d'indication TC-BEGIN se produisant du côté de l'entité appelée;
- la première primitive d'indication TC-CONTINUE se produisant du côté de l'émission d'appel ou dans des conditions spécifiques:
 - une primitive d'indication TC-END du côté de l'émission d'appel;
 - une primitive d'indication TC-U-ABORT se produisant du côté de l'émission d'appel;
 - une primitive d'indication TC-P-ABORT se produisant du côté de l'émission d'appel.

3.4.3.1 Envoi d'une primitive de demande TC-BEGIN

Avant de présenter une primitive de demande TC-BEGIN, la règle SACF enregistrera le nom de l'AC et, le cas échéant, le paramètre d'information de l'utilisateur.

La règle SACF demandera l'appel des opérations connexes en utilisant le service TC-INVOKE. Voir 3.4.8 pour la description de la procédure d'appel.

Après le traitement de la dernière demande d'appel, la règle SACF présentera une primitive de demande TC-BEGIN.

L'entité appelante SACF attendra une primitive d'indication TC et ne présentera pas d'autres demandes, sauf une demande TC-U-ABORT ou une demande TC-END avec le paramètre de méthode de libération mis à "libération sur fin avec entente préalable".

Si aucune primitive d'indication TC n'est attendue parce qu'un dialogue ne doit pas être établi conformément aux règles comme cela est indiqué aux 3.4.1 et 3.4.2, la règle SACF attendra que la temporisation de la dernière opération TCAP associée vienne à expiration et présentera une demande TC-END avec le paramètre de méthode de libération mis à "libération sur fin avec entente préalable".

3.4.3.2 Réception d'une indication TC-BEGIN

A la réception d'une primitive d'indication TC-BEGIN, le répondeur SACF devra:

- analyser le nom du contexte d'application s'il est inclus dans la primitive. S'il est pris en charge, il convient de traiter toutes les autres primitives d'indication reçues du TC comme cela est décrit au 3.4.8;
- si aucun dialogue n'est établi selon les règles indiquées aux 3.4.1 et 3.4.2, la règle SACF attendra la dernière primitive d'indication du TC et présentera une demande TC-END avec le paramètre de méthode de libération mis à "libération sur fin avec entente préalable";
- si le nom du contexte d'application inclus dans la primitive n'est pas pris en charge, présenter une primitive de demande TC-U-ABORT. Si un autre contexte d'application peut se présenter, son nom est inclus dans la primitive de demande TC-U-ABORT.

Le fait qu'une négociation du contexte d'application soit ou non limitée à l'utilisation de la primitive TC-U-ABORT nécessite un complément d'étude.

3.4.3.3 Réception de la première primitive d'indication TC-CONTINUE

A la réception de la première primitive d'indication TC-CONTINUE pour un dialogue, la règle SACF vérifiera la valeur du paramètre "nom du contexte d'application". Si cette valeur correspond à celle utilisée dans la primitive de demande TC-BEGIN, la règle SACF traitera les primitives d'indication suivantes de traitement de composantes TC comme cela est décrit au 3.4.8; dans le cas contraire, elle présentera une primitive de demande TC-U-ABORT.

Le fait qu'une négociation du contexte d'application soit ou non limitée à l'utilisation de la primitive TC-U-ABORT nécessite un complément d'étude.

3.4.3.4 Réception d'une primitive d'indication TC-END

A la réception d'une primitive d'indication TC-END à l'état "dialogue initiated", la règle SACF vérifiera la valeur du paramètre du nom de contexte d'application. Si sa valeur correspond à celle utilisée dans la primitive de demande TC-BEGIN, la règle SACF traitera les primitives d'indication suivantes du traitement de composantes TC comme cela est décrit au 3.4.8.

3.4.3.5 Réception d'une primitive d'indication TC-U-ABORT

La réception d'une primitive d'indication TC-U-ABORT est décrite comme faisant partie de la procédure d'interruption de l'utilisateur (voir 3.4.6.2). Si la cause d'interruption est "application context name not supported" (nom de contexte d'application non pris en charge), l'entité de réponse peut proposer un autre nom de contexte d'application dans l'indication TC-U-ABORT. Si un autre contexte d'application est proposé, l'entité à la réception vérifiera ce nom et s'il peut être pris en charge, un nouveau dialogue peut être établi.

3.4.3.6 Réception d'une primitive d'indication TC-P-ABORT

La réception d'une primitive d'indication TC-P-ABORT est décrite comme faisant partie d'une procédure d'interruption du fournisseur (voir 3.4.7.1)

3.4.4 Continuité du dialogue

Une fois établi, on dit que le dialogue est en phase de continuité.

Les deux processus d'application peuvent demander le transfert de APDU du protocole d'application INAP jusqu'à ce que l'un d'eux demande qu'il soit mis fin au dialogue.

3.4.4.1 Envoi de l'entité

La règle SACF traitera toutes les primitives de demande de traitement de composantes comme cela est décrit au 3.4.8.

Après le traitement de la dernière primitive de demande de traitement de composantes, la règle SACF présentera une primitive de demande TC-CONTINUE.

3.4.4.2 Réception de l'entité

A la réception d'une primitive d'indication TC-CONTINUE, la règle SACF acceptera zéro, une ou plusieurs primitives d'indication de traitement de composantes TC et les traitera comme cela est décrit au 3.4.8.

3.4.5 Fin de dialogue

L'entité appelante de dialogue et l'entité appelé auront toutes deux la possibilité de demander la fin d'un dialogue une fois qu'il a été établi, pour autant qu'un autre dialogue ne doive pas être établi ou qu'il ne doive plus être poursuivi, comme cela est indiqué aux points 3.4.1 et 3.4.2.

La procédure de fin de dialogue est provoquée par les événements suivants:

- une primitive de demande TC-END;
- une primitive d'indication TC-END.

3.4.5.1 Envoi d'une demande TC-END

Quand le dialogue ne doit plus être maintenu, la règle SACF traitera toutes les primitives de demande de traitement de composantes comme cela est décrit au 3.4.8.

Après le traitement de la dernière primitive de demande de traitement de composantes (le cas échéant), la règle SACF présentera une primitive de demande TC-END avec le paramètre de méthode de libération mis sur "fin normale" ou "libération sur fin avec entente préalable", selon les règles énoncées aux 3.4.1 et 3.4.2.

Si aucun dialogue n'est établi, il convient de se référer aux 3.4.3.1 et 3.4.3.2.

3.4.5.2 Réception d'une primitive d'indication TC-END

A la réception d'une primitive d'indication TC-END, la règle SACF acceptera toutes les primitives d'indication de traitement de composantes et les traitera comme cela est décrit au 3.4.8.

Après avoir traité la dernière primitive de traitement de composantes, toutes les ressources liées au dialogue seront libérées.

3.4.6 Interruption de l'utilisateur

L'entité appelante et l'entité appelée auront toutes deux la possibilité d'interrompre un dialogue à n'importe quel moment.

La procédure d'interruption de l'utilisateur est provoquée par un des événements suivants:

- une primitive de demande TC-U-ABORT;
- une primitive d'indication TC-U-ABORT.

3.4.6.1 Envoi d'une primitive de demande TC-U-ABORT

Après la présentation d'une primitive de demande TC-U-ABORT, toutes les ressources liées au dialogue sont libérées.

3.4.6.2 Réception d'une primitive d'indication TC-U-ABORT

A la réception d'une primitive d'indication TC-U-ABORT, toutes les ressources relatives au dialogue sont libérées.

3.4.7 Interruption du service fournisseur

Le TC a la capacité d'interrompre un dialogue à la fois du côté entité appelante de dialogue et du côté répondeur de dialogue.

La procédure interruption du service fournisseur est provoquée par l'événement suivant:

- une primitive d'indication TC-P-ABORT.

3.4.7.1 Réception d'une indication TC-P-ABORT

A la réception d'une indication TC-P-ABORT, toutes les ressources afférentes au dialogue sont libérées.

3.4.8 Procédures pour les opérations INAP

Ce paragraphe décrit les procédures applicables aux opérations INAP.

3.4.8.1 Appel d'opération

La règle SACF permettra de préparer un argument d'opération à partir des paramètres reçus et de demander l'appel de l'opération associée en utilisant la procédure TC-INVOKE. Si un paramètre couplé ID est inséré dans la primitive, ceci indique un processus fils et implique que l'opération est liée à un processus père.

3.4.8.2 Réception de l'appel d'opération

A la réception d'une primitive d'indication TC-INVOKE, la règle SACF devra:

- si le code d'opération ne correspond pas à une opération prise en charge par le contexte d'application, demander le transfert d'une composante rejet en utilisant une primitive de demande TC-U-REJECT, avec le code problème approprié (opération non reconnue);
- si un identificateur couplé est inclus, exécuter les vérifications suivantes: si l'opération à laquelle il est fait référence par l'identificateur couplé ne permet pas les opérations liées, ou si le code d'opération ne correspond pas à une opération liée autorisée, ou si l'appel du processus père n'est pas actif, présenter une primitive de demande TC-U-REJECT avec le code problème approprié (réponse liée imprévue ou opération liée imprévue);
- si le type d'argument n'est pas défini pour l'opération, demander le transfert de la composante de rejet en utilisant la primitive de demande TC-U-REJECT avec le code problème (paramètre de type erroné);
- si l'opération ne peut pas être appelée parce que le dialogue relatif à INAP est sur le point d'être libéré, demander le transfert de la composante rejet en utilisant une primitive de demande TC-U-REJECT avec le code problème approprié (déclenchant la libération);
- si des ressources suffisantes liées à INAP ne sont pas disponibles pour exécuter l'opération demandée, demander le transfert de la composante de rejet en utilisant une primitive de demande TC-U-REJECT avec le code problème (limite de ressources);
- dans un autre cas, il convient d'accepter la primitive d'indication TC-INVOKE. Si l'opération doit être confirmée par l'utilisateur, la règle SACF attendra la réponse correspondante.

3.4.8.3 Réponse d'opération

Pour les opérations confirmées par l'utilisateur, la règle SACF devra:

- si aucune indication d'erreur n'est comprise dans la réponse à l'opération de classe 1 à 3, élaborer un élément d'information de sortie à partir des paramètres reçus et demander son transfert en utilisant le service TC-RESULT-L;
- si une indication d'erreur est incluse dans la réponse à une opération de classe 1 ou 2, élaborer un paramètre d'erreur à partir des paramètres reçus et de demander son transfert en utilisant une primitive de demande TC-U-ERROR.

3.4.8.4 Réception d'une réponse

3.4.8.4.1 Réception d'une indication TC-RESULT-NL

A la réception d'une indication TC-RESULT-NL, la règle SACF devra:

- demander le transfert d'une composante de rejet en utilisant une primitive de demande TC-U-REJECT, avec le code problème approprié (paramètre de type erroné).

3.4.8.4.2 Réception d'une indication TC-RESULT-L

A la réception d'une indication TC-RESULT-L, la règle SACF devra:

- si le type de paramètre de sortie n'est pas défini pour le résultat de cette opération, demander le transfert d'une composante rejet en utilisant une primitive de demande TC-U-REJECT, avec le code de problème approprié (paramètre de type erroné);
- dans un autre cas, accepter la primitive d'indication TC-RESULT-L.

3.4.8.4.3 Réception d'une indication TC-U-ERROR

A la réception d'une indication TC-U-ERROR, la règle SACF devra:

- si le code d'erreur n'est pas défini pour la règle SACF ou n'est pas associé à l'opération à laquelle il est fait référence par l'identificateur d'appel, demander le transfert d'une composante rejet en utilisant une primitive de demande TC-U-REJECT, avec le code problème approprié (erreur non reconnue ou erreur imprévue);
- si le type de paramètre d'erreur n'est pas celui défini pour cette erreur, demander le transfert d'une composante rejet en utilisant une primitive d'indication TC-U-REJECT, avec le code problème approprié (paramètre de type erroné);
- dans un autre cas, accepter la primitive d'indication TC-U-ERROR.

3.4.8.4.4 Réception d'une primitive d'indication TC-U-REJECT

A la réception d'une primitive d'indication TC-U-REJECT affectant une opération en cours, la règle SACF acceptera la primitive d'indication TC-U-REJECT.

3.4.8.4.5 Réception d'une primitive d'indication TC-L-REJECT

Cet événement survient quand le TC local détecte une erreur de protocole dans une composante entrante qui affecte une opération.

A la réception d'une indication TC-L-REJECT indiquant "renvoyer le résultat de l'incident, message d'erreur non prévu", la règle SACF informera le processus d'application.

A la réception d'une indication TC-REJECT indiquant "renvoyer l'erreur, message d'erreur non prévu", la règle SACF informera le processus d'application.

Noter que lorsque le code problème indique un problème d'ordre général, on considère que cet événement ne peut pas concerner une opération active même si l'identificateur invoque Id est indiqué par le TC. Ceci parce qu'il est incertain que cet identificateur fasse référence à une invocation locale ou distante. Le comportement de la règle SACF dans un tel cas est décrit au 3.4.8.5.3.

3.4.8.4.6 Réception d'une primitive d'indication TC-L-CANCEL

A la réception d'une primitive d'indication TC-L-CANCEL, la règle SACF doit:

- si l'opération associée est une opération de classe 1, informer le processus d'application;
- si l'opération associée est une opération de classe 2 et si des opérations non liées sont définies pour cette opération, omettre la primitive;
- si l'opération associée est une opération de classe 2 et a des opérations liées, mais aucune d'elles n'a été appelée, informer le processus d'application;
- si l'opération associée est une opération de classe 2 et une invocation d'opération liée a déjà été reçue en réponse à cette opération, omettre la primitive;
- si l'opération associée est une opération de classe 3, informer le processus d'application;
- si l'opération associée est une opération de classe 4, omettre la primitive.

3.4.8.5 Autres événements

Ce paragraphe décrit le comportement de la SACF à la réception d'une primitive d'indication de traitement de composante qui ne peut pas concerner une opération quelconque ou qui n'affecte pas une opération en cours.

3.4.8.5.1 Réception d'une primitive d'indication TC-U-REJECT

A la réception d'une primitive d'indication TC-U-REJECT qui n'affecte pas une opération active (c'est-à-dire indiquant un renvoi de résultat ou un problème de renvoi d'erreur), il incombe au processus d'application d'interrompre, de continuer ou de mettre fin au dialogue, s'il n'a pas été mis fin à ce dialogue par l'envoi du processus d'application selon les règles comme cela est défini au 3.4.2. Ceci s'applique également aux problèmes d'appel concernant l'opération liée de classe 4.

3.4.8.5.2 Réception d'une primitive d'indication TC-R-REJECT

A la réception d'une primitive d'indication TC-R-REJECT (c'est-à-dire quand une erreur de protocole a été détectée par l'entité homologue TC) ce qui n'affecte pas une opération active, il incombe au processus d'application d'interrompre, de continuer ou de mettre fin au dialogue, s'il n'a pas déjà été mis fin à ce dialogue par l'envoi du processus d'application selon les règles comme cela est indiqué au 3.4.2.

3.4.8.5.3 Réception d'une primitive d'indication TC-L-REJECT

A la réception d'une primitive d'indication TC-L-REJECT (c'est-à-dire quand une erreur de protocole a été détectée par l'entité TC locale), ce qui ne peut pas se rapporter à une opération active, il incombe au processus d'application de continuer ou de mettre fin au dialogue et de déclencher implicitement la transmission de la composante rejet ou d'interrompre le dialogue.

3.4.8.5.4 Réception d'une primitive d'indication TC-NOTICE

La règle SACF est informée qu'un message ne peut pas être présenté par la couche Réseau, cela ne pouvant se produire que si une option de renvoi a été fixée (voir 3.4.9.1.8). Il incombe au processus d'application de choisir de mettre ou non fin au dialogue ou d'essayer de relancer une opération.

3.4.9 Correspondances vers les services TC

3.4.9.1 Commande de dialogue

Le service TC-UNI n'est pas utilisé par INAP.

3.4.9.1.1 Adresse de destination

Ce paramètre est fixé par le processus d'application de déclenchement du dialogue, et peut être facultativement modifié par le dialogue de réponse dans le premier paramètre TC-CONTINUE arrière, mais ne sera pas vu par l'utilisateur appelant.

3.4.9.1.2 Adresse de départ

Ce paramètre est fixé par le processus d'application déclenchant le dialogue.

3.4.9.1.3 Identificateur de dialogue

La valeur de ce paramètre est associée à l'invocation INAP d'une façon dépendante de la réalisation.

3.4.9.1.4 Nom du contexte d'application

Le paramètre du nom de contexte d'application est fixé par la règle SACF comme cela est défini au 2.1.5 et au 2.2.2.5.3.

3.4.9.1.5 Information utilisateur

Ce paramètre peut être utilisé à la fois par le processus d'application appelant et appelé.

3.4.9.1.6 Composante existante

Ce paramètre est utilisé par la règle SACF comme cela est décrit dans la Recommandation Q.771.

3.4.9.1.7 Fin d'opération

La valeur du paramètre de méthode de libération de la primitive de demande TC-END est fixée par la règle SACF selon les règles définies aux 3.4.1 et 3.4.2.

3.4.9.1.8 Qualité de service

La qualité de service de primitive de demande TC est fixée par la règle SACF selon la valeur suivante:

- maintien en séquence demandé;
- option de retour – Ce paramètre étant fixé par la règle SACF d'une façon dépendante de la réalisation.

3.4.9.2 Procédures d'opération

3.4.9.2.1 Identificateur d'invocation

Ce paramètre est fixé par l'envoi du processus d'application.

3.4.9.2.2 Identificateur lié

Ce paramètre est fixé par l'envoi du processus d'application.

3.4.9.2.3 Identificateur de dialogue

La valeur de ce paramètre est associée à l'invocation INAP d'une façon dépendante de la réalisation.

3.4.9.2.4 Classe

La valeur de ce paramètre est fixée par la règle SACF selon le type de l'opération devant être invoquée selon 2.1.

3.4.9.2.5 Opération

Le code d'opération d'une primitive de demande TC-INVOKE est fixé par l'envoi du processus d'application comme cela est défini au 2.4.

La règle SACF fixera le code d'opération de la primitive TC-RESULT-L (le cas échéant) à la même valeur que celle reçue au moment de l'invocation.

3.4.9.2.6 Erreur

L'erreur de paramètre de la primitive de demande TC-U-ERROR est fixée par l'envoi du processus d'application comme cela est défini au 2.4.

3.4.9.2.7 Paramètres

Le paramètre d'argument des primitives TC-INVOKE est fixé par l'envoi du processus d'application comme cela est défini à l'article 2.

Le paramètre résultant des primitives TC-RESULT-L est fixé par l'envoi du processus d'application comme cela est défini à l'article 2.

Le paramètre résultant des primitives TC-U-ERROR est fixé par l'envoi du processus d'application comme cela est défini à l'article 2.

3.4.9.2.8 Expiration de la temporisation

La valeur de ce paramètre est fixée par la règle SACF selon le type d'opération invoquée.

3.4.9.2.9 Dernière composante

Ce paramètre est utilisé par la règle SACF comme cela est décrit dans la Recommandation Q.771.

3.4.9.2.10 Code de problème

Ce paramètre est utilisé par la règle SACF comme cela est décrit au 3.4.8.

3.4.9.2.11 Cause d'interruption

Ce paramètre est utilisé par la règle SACF, et les attributs et le codage sont spécifiés par l'exploitant de réseau.

Annexe A

Diagrammes SDL INAP

(Cet annexe fait partie intégrante de la présente Recommandation)

A.1 Introduction

Cette annexe contient des diagrammes SDL pour les machines FSM de l'entité SSF, l'entité SSF d'assistance/de transfert de tâche, l'entité SCF et l'entité SRF, comme cela est décrit dans le texte et les Figures 3.1.1.5, 3.1.1.6, 3.1.2 et 3.1.3, de même que les informations provenant de procédures détaillées d'opération. Les diagrammes SDL pour les états associés à l'entité SDF, à l'entité SCF et à l'entité SDF elle-même sont contenus dans l'Annexe B.

Les notes relatives aux diagrammes donnent si nécessaire d'autres détails. Les diagrammes SDL ont le même niveau de détail que le texte et les figures qui s'y rattachent, et ne contiennent rien qui ne fasse pas partie du corps de cette Recommandation à moins que cela n'ait été spécifiquement noté.

Les diagrammes SDL utilisés sont conformes aux dernières Recommandations relatives aux diagrammes SDL [Z.100 (1993)], mais aucun symbole qui n'est pas contenu dans Z.100 (1988) n'est utilisé.

En cas de compatibilité entre les diagrammes SDL et le texte, le texte sera considéré comme étant correct.

A.2 Diagrammes SDL

Le langage SDL dans ces diagrammes est très évolué. Il est incomplet du fait que les déclarations de données sont faites uniquement pour les temporisateurs et les signaux, mais les données transportées par les signaux ne sont pas déclarées.

Les noms utilisés pour les opérations sont exactement celles de la présente Recommandation et d'autres noms définis localement concernent en général des noms d'événement utilisés dans la présente Recommandation. Dans certains cas, des nouveaux noms ont cependant été introduits pour des signaux pour lesquels la présente Recommandation ne fournit pas de nom. Dans les cas où les noms d'opération ont été décomposés de façon à contenir un symbole, la technique SDL de la rupture des lignes avec un trait souligné suivant un espace est utilisé; cette combinaison est omise en langage SDL de sorte que A_B a le même nom que AB.

En général, la réception d'un signal (d'entrée) à partir d'une entité fonctionnelle à l'extérieur d'une entité fonctionnelle qui est en cours de modélisation nécessitera un certain traitement de la logique de service, mais le détail d'un tel traitement n'intéresse pas l'entité FSM (elle intéresse seulement les résultats). L'existence de ce traitement de logique de service a été indiqué par l'inclusion d'une procédure générale appelée Entrée de Procédé. Ailleurs, les tâches SDL sont utilisées pour indiquer le champ de traitement nécessaire.

Il convient de noter que le terme de «réinitialisation» s'applique en langage SDL à des moyens de temporisation destinés à arrêter la temporisation, et non pas à la faire redémarrer. «Initialiser» en langage SDL couvre à la fois le démarrage et le redémarrage d'un temporisateur.

```

/*
SSF SDLs for INAP
Based on Q.1218, Section 3.1.1.5, "SSF State Transition Diagram".
Version 2.0 January 1995
*/

```

```

Process_Input

```

```

/* Data declarations */
TIMER Tssf;

```

```

/* Timer Tssf can have four different values, as described in Q.1218, Section 3.1.1.5. These
are denoted:

value1 when entering State Waiting For Instructions after sending initiating IF to SCF
value2 when entering State Waiting For Instructions under any other condition
value3 when SSF receives HoldCallInNetwork operation
value4 when entering States Waiting For End of User Interaction or Waiting For End of
Temporary Connection (optional)

*/

```

```

/*
A locally defined procedure Process_Input is used to indicate analysis of an Input from an external source to
determine whether service logic processing (outside the SSF FSM) is required.

No details of possible processing in the procedure are given, as it is intended only to indicate that processing
may be required, not its exact nature.

*/

```

T1171770-95/d44

FIGURE A.1/Q.1218 (feuillet 1 de 17)
Diagramme SDL pour SSF-FSM

```

/* Signal definitions - First part
These signals to and from CCF are internal indications which are not defined in the IN
CS-1 Recommendations.

The names are therefore local names only.
*/
/* From CCF */
SIGNAL ForwardConnectionReleased, Abandon, Disconnect;
/* To CCF */
SIGNAL RestartProcessing, TerminateCall, ConnectSRF, RouteToDefault, SRFReleaseRequest;

```

```

/* Signal definitions - Second part.
These signals from internal (SSF) logic are internal indications which are not defined in the IN CS-1
Recommendations.

The names are therefore local names only.
*/
/* From SSF service logic */
SIGNAL TDP_R, TDP_N, EDP_N, EDP_R;
/* Local names defined for signals SSF to SCF */
SIGNAL TDPEvent; /* Indicates one of InitialDP or a DP specific operation */
SIGNAL EDPEvent; /* Indicates one of EventReportBCSM or a DP specific operation */

```

```

/* Signal definitions - Third part. Defined in IN CS-1 Recommendations. */
/* To SCF - The following are the DP specific operations from SSF to SCF */
SIGNAL TAnswer, TDisconnect, TermAttemptAuthorized, TmidCall, TNoAnswer, AnalysedInformation,
TCalledPartyBusy, CollectedInformation, OAnswer, OCalledPartyBusy, ODisconnect, OMidCall,
ONoAnswer, OriginationAttemptAuthorized, RouteSelectFailure;
/* Other signals SSF to SCF */
SIGNAL Initial DP, EventReport BCSM, ApplyChargingReport, EventNotificationCharging,
CallInformationReport;
/* From SCF */
SIGNAL InitiateCallAttempt, ConnectToResource, EstablishTemporaryConnection, HoldCallInNetwork,
ApplyCharging, CallInformationRequest, FurnishChargingInformation, RequestNotificationChargingEvent,
RequestReportBCSMEvent, ResetTimer, SendChargingInformation, AnalyseInformation,
CollectInformation, Connect, Continue, SelectFacility, SelectRoute, DisconnectForwardConnection,
ReleaseCall;

```

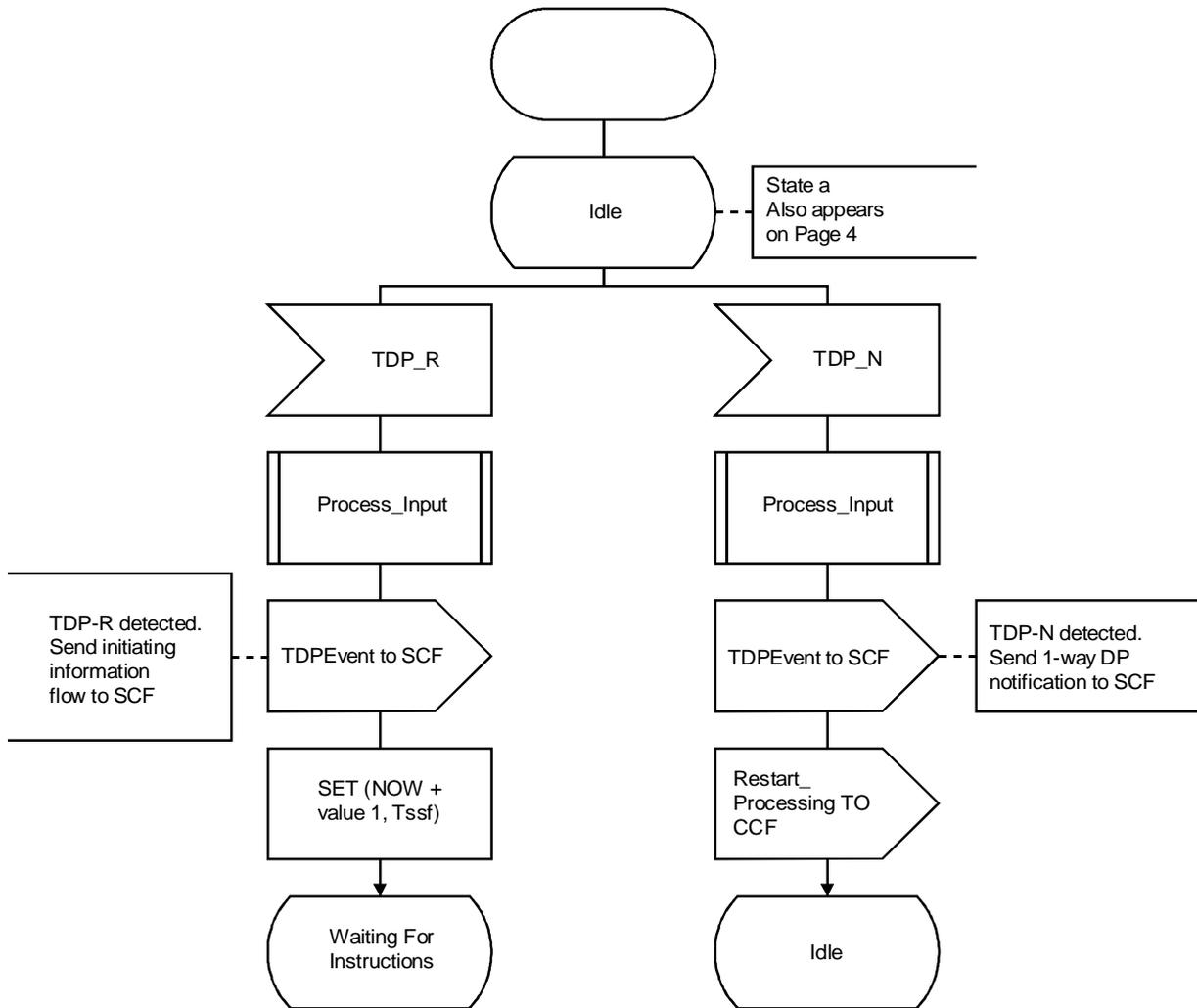
```

/* Signal definitions - Fourth part. Defined in IN CS-1 Recommendations. */
/* Relay from SCF to SRF */
/* NOTE - Cancel can also be used SCF to SSF */
SIGNAL Cancel, PlayAnnouncement, PromptAndCollectUserInformation;
/* Relay from SRF to SCF */
SIGNAL ReturnResult_from_PromptAndCollectUserInformation, SpecializedResourceReport;

```

FIGURE A.1/Q.1218 (feuillet 2 de 17)
Diagramme SDL pour SSF-FSM

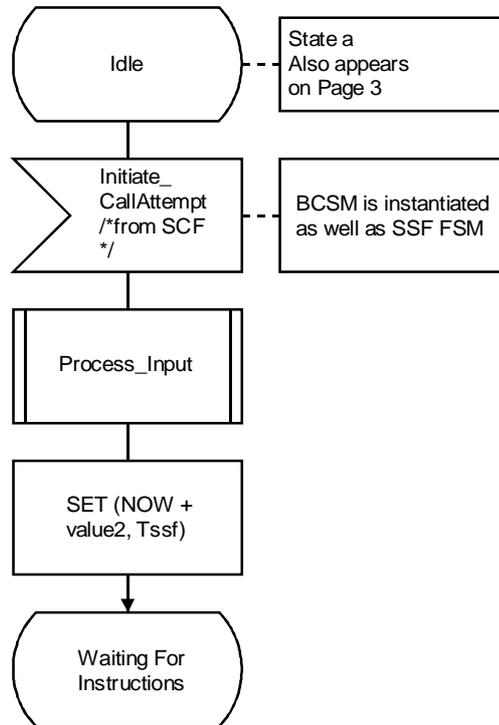
/* TDPEvent is a locally-defined name for the signal which is sent from the SCF.
 Its meaning is:
 Send either InitialDP or a DP specific operation
 (as defined in Q.1218, Section 3.1.1.5)
 */



T1171790-95/d46

FIGURE A.1/Q.1218 (feuille 3 de 17)

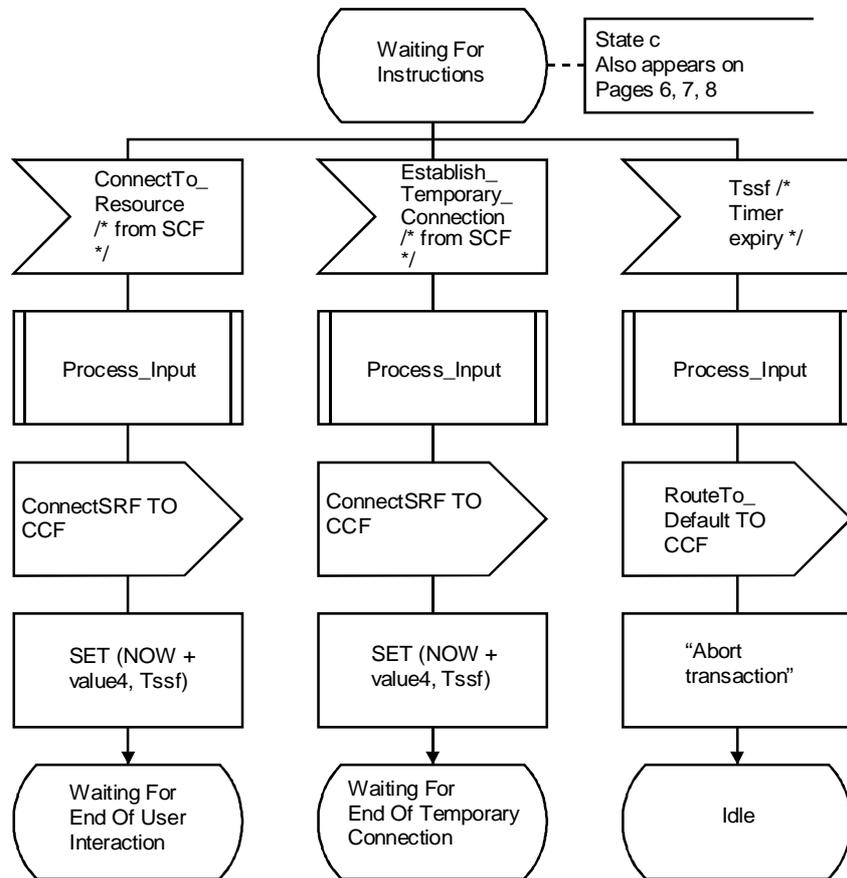
Diagramme SDL pour SSF-FSM



T1171800-95/d47

FIGURE A.1/Q.1218 (feuillet 4 de 17)
Diagramme SDL pour SSF-FSM

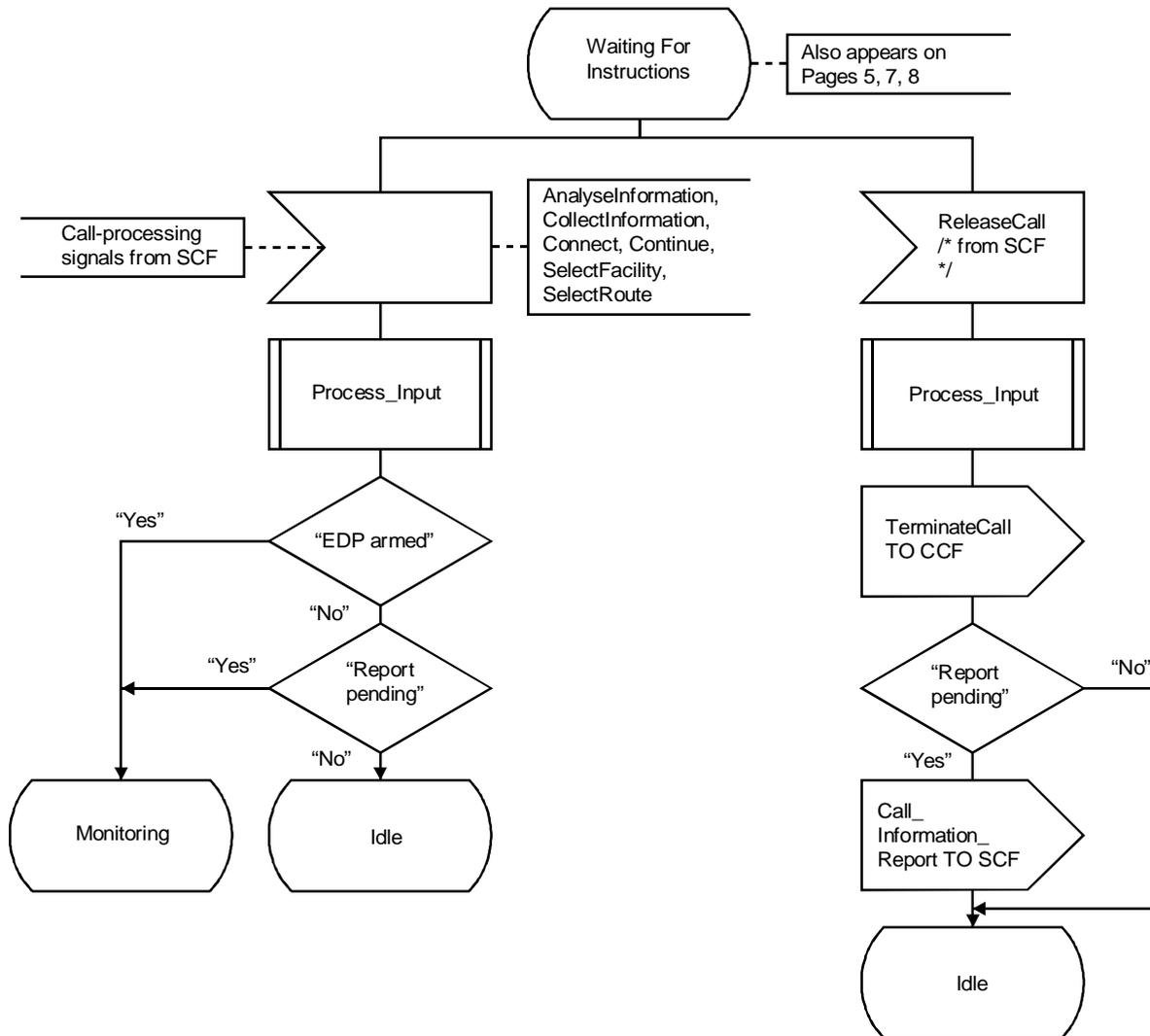
/* Use of timer Tssf in states Waiting For End of User Interaction and Waiting For End Of Temporary Connection is optional.
 Timer value after receipt of Cancel may be value2, if call resulted from InitiateCallAttempt operation
 */



T1171810-95/d48

FIGURE A.1/Q.1218 (feuillet 5 de 17)
Diagramme SDL pour SSF-FSM

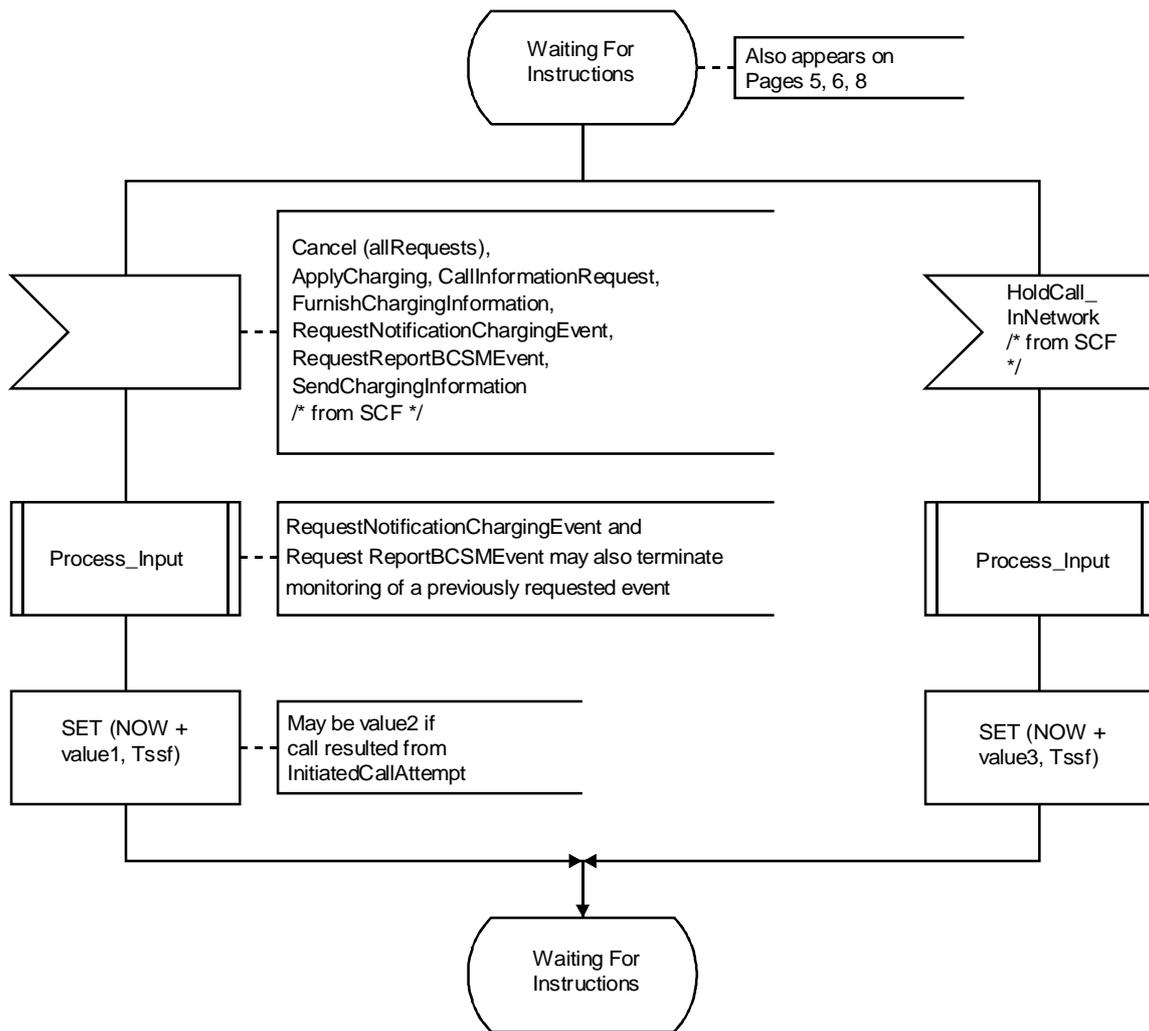
/* Where test is made for CallInformationReport pending, ApplyChargingReport is also to be handled the same way */



T1171820-95/d49

FIGURE A.1/Q.1218 (feuille 6 de 17)

Diagramme SDL pour SSF-FSM



T1171830-95/d50

FIGURE A.1/Q.1218 (feuillet 7 de 17)

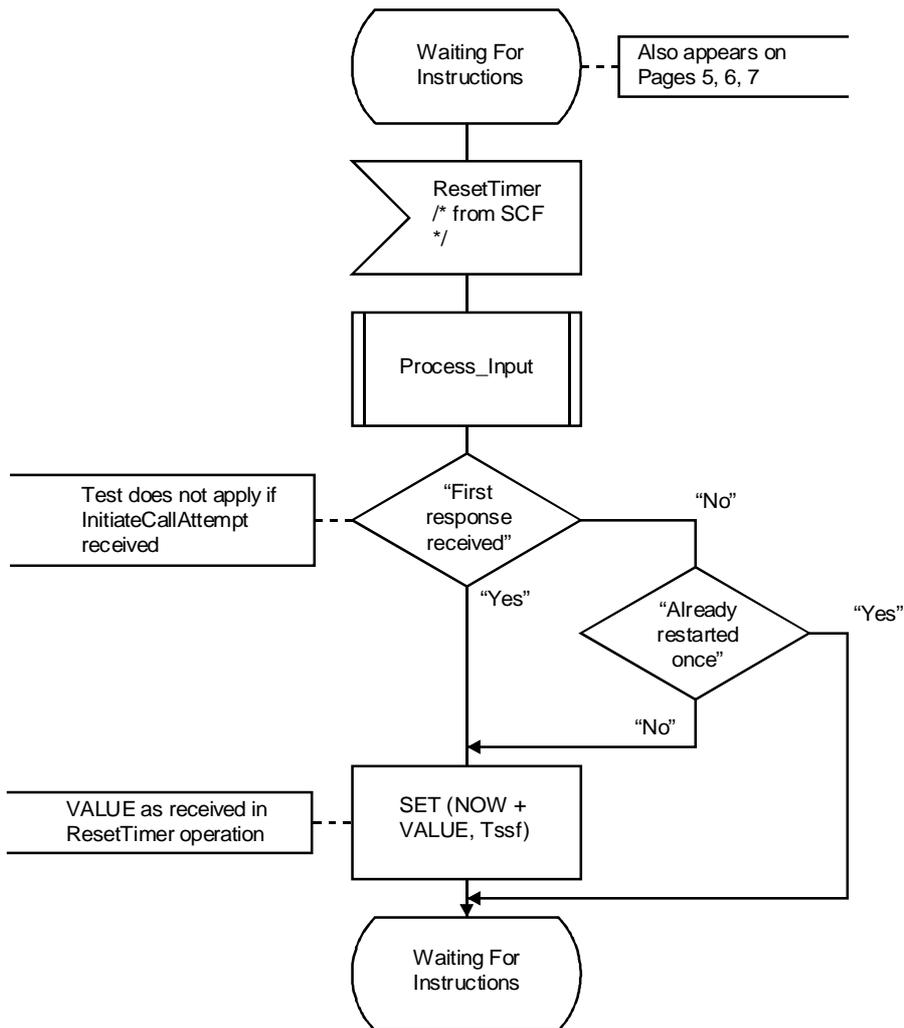
Diagramme SDL pour SSF-FSM

/* While waiting for the first response from the SCF, the timer Tssf can be restarted only once by a ResetTimer operation. Subsequent to the first response, the timer can be restarted any number of times.

ResetTimer itself is not treated as the first response.

When InitiateCallAttempt has been received, the timer can be restarted any number of times.

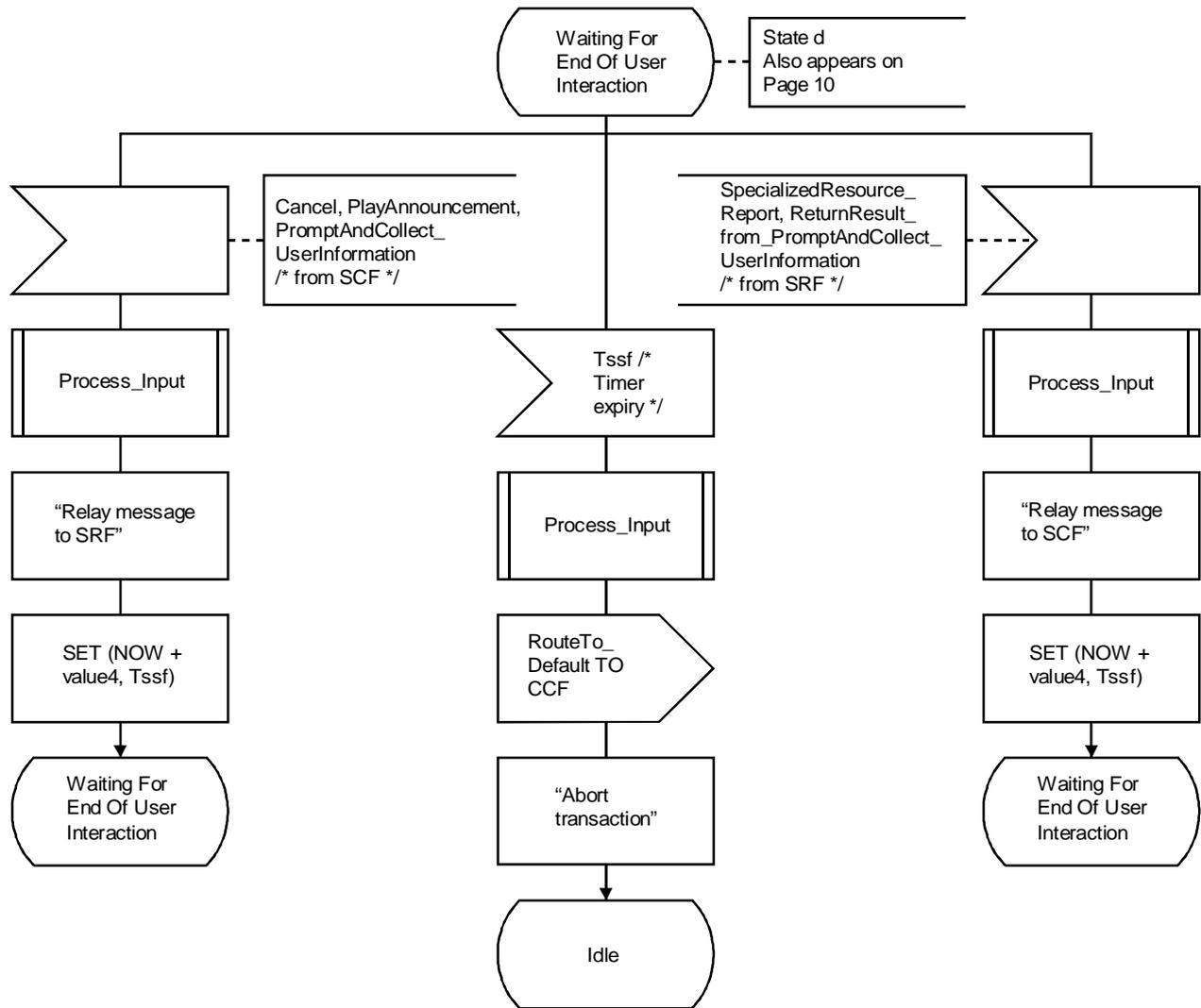
*/



T1171840-95/d51

FIGURE A.1/Q.1218 (feuillet 8 de 17)
Diagramme SDL pour SSF-FSM

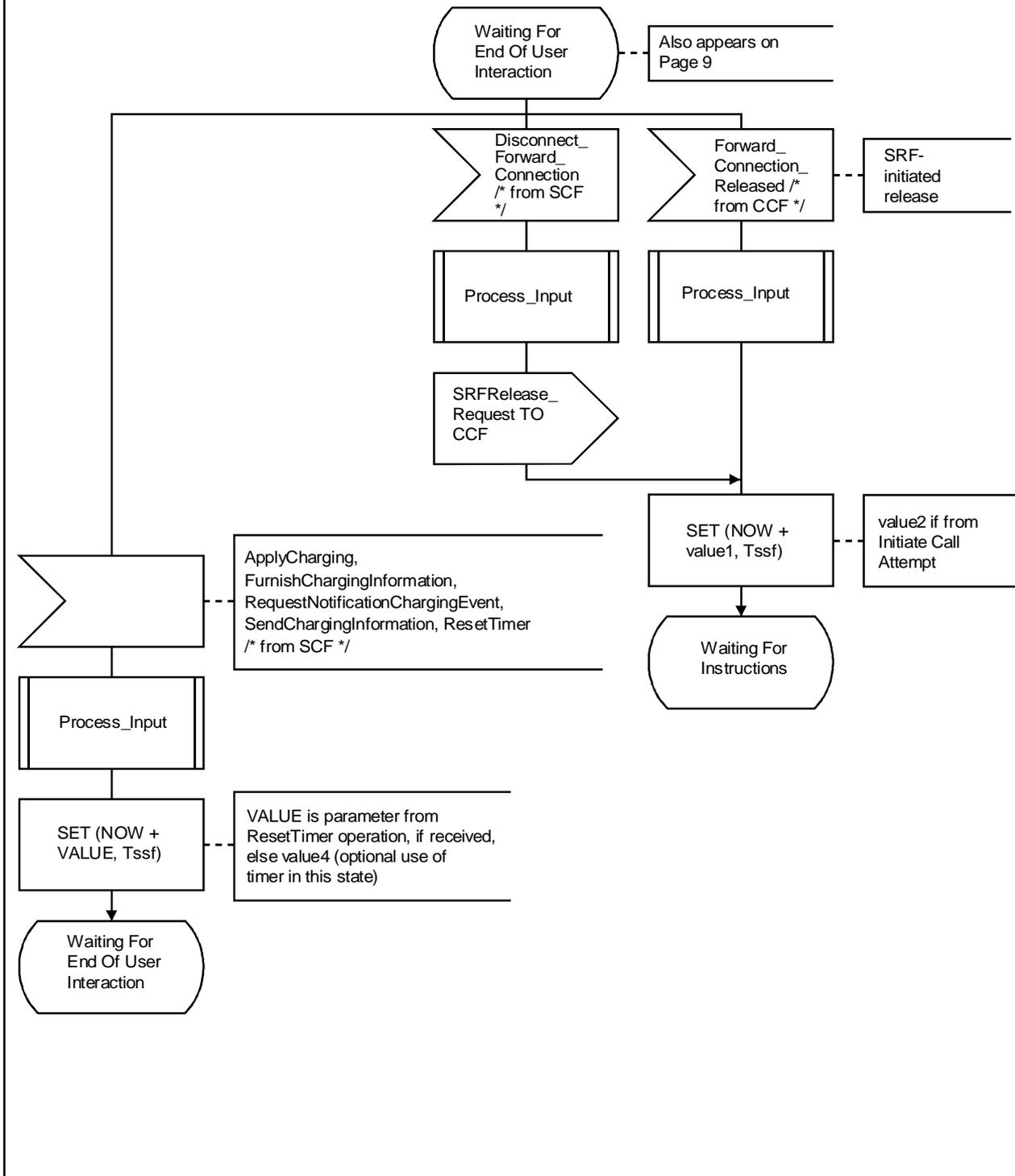
/* Cancel here is cancellation of an announcement.
Use of timers for this state is optional.
*/



T1171850-95/d52

FIGURE A.1/Q.1218 (feuille 9 de 17)

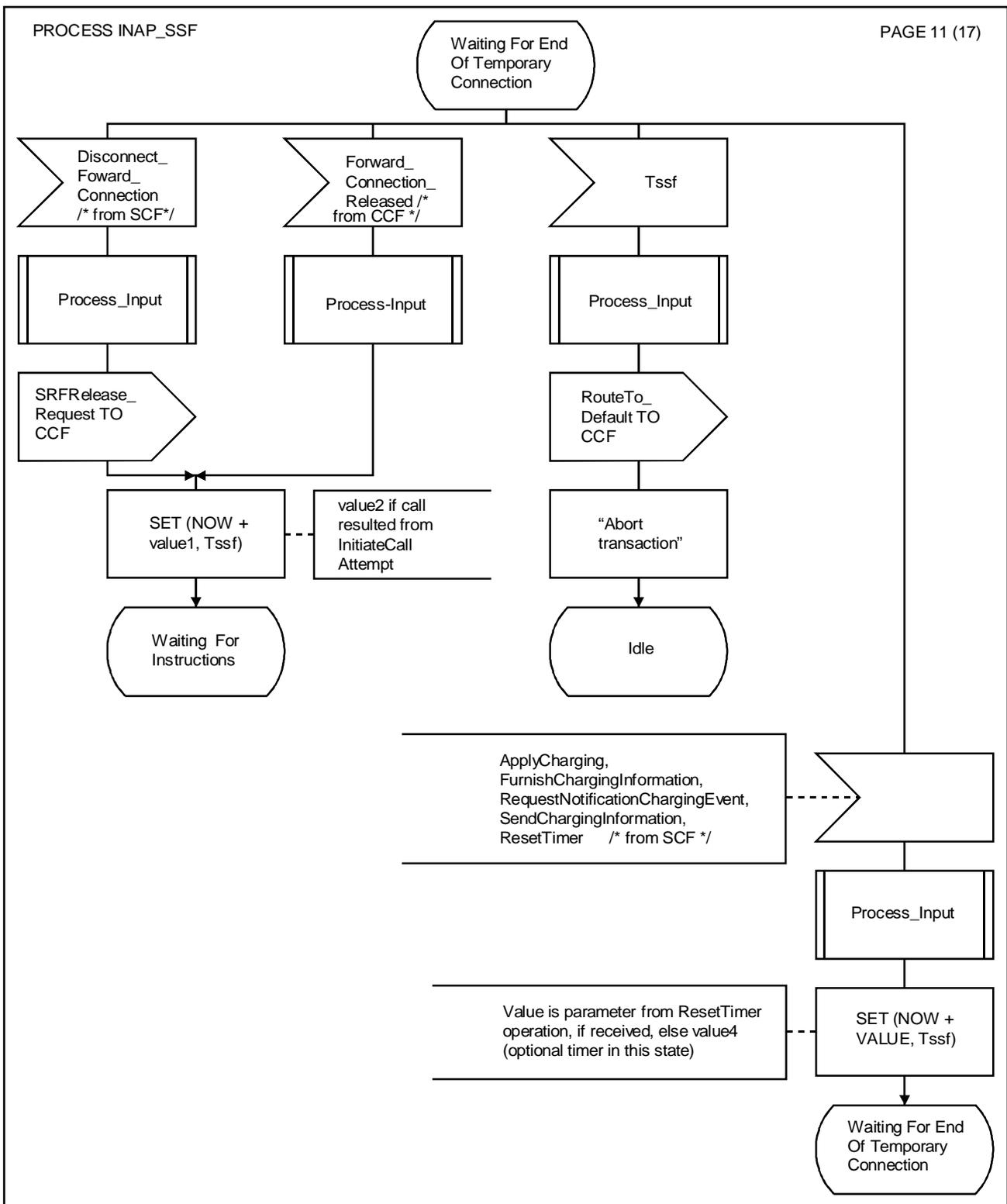
Diagramme SDL pour SSF-FSM



T1171860-95/d53

FIGURE A.1/Q.1218 (feuillet 10 de 17)

Diagramme SDL pour SSF-FSM



T1171870-95/d54

FIGURE A.1/Q.1218 (feuillet 11 de 17)
Diagramme SDL pour SSF-FSM

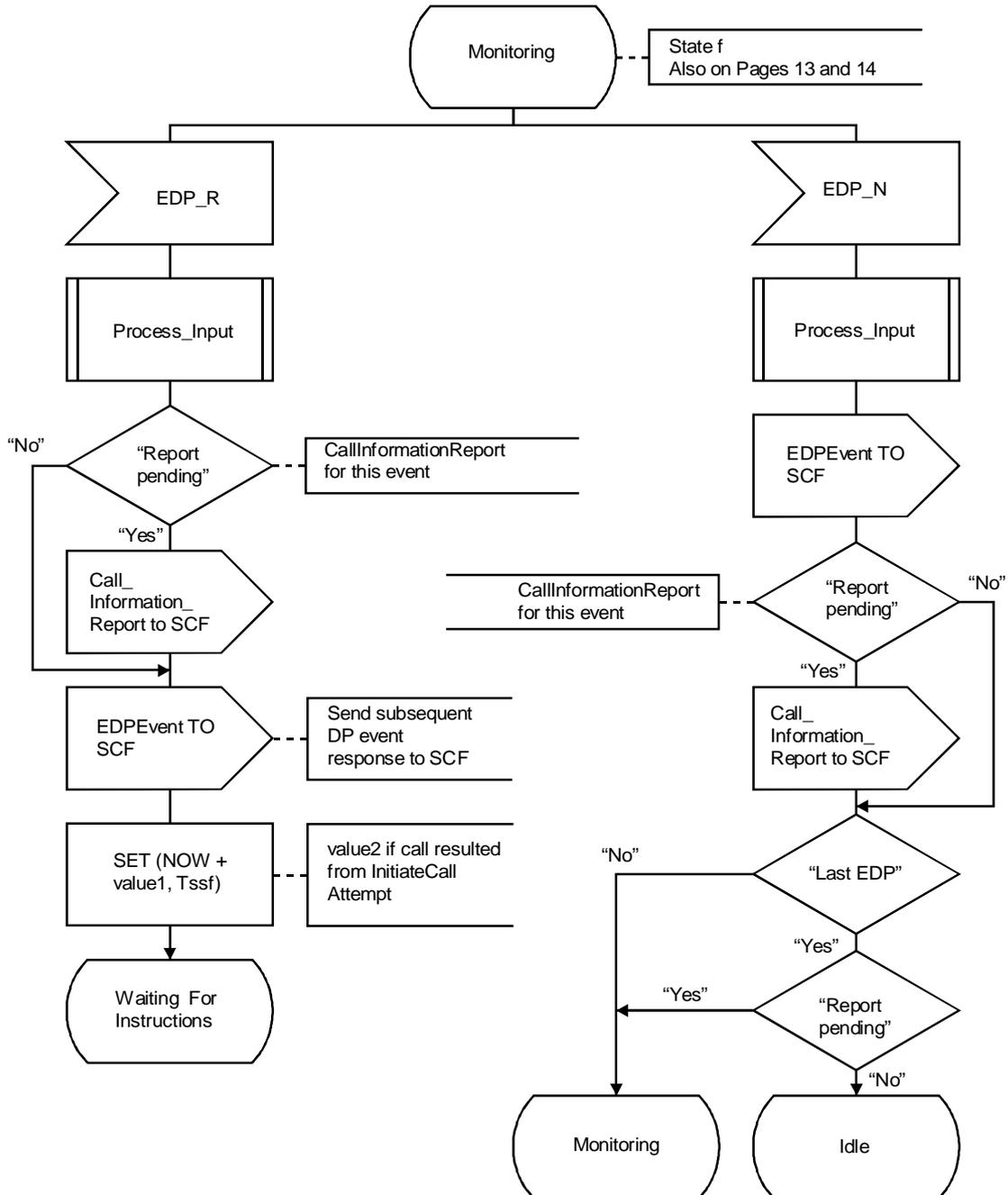
/* Only EDPs are handled within the existing FSM. If a TDP is armed it will create a new instance of an FSM to handle the interaction.

DPs 4, 5, 6, 8 in O_BCSM and 13, 14, 16 in T_BCSM shown here.

Handling of EDPs at DPs 9, 10, 17 and 18 (abandon and disconnect) is shown on page 16.

The description here follows CallInformationReport procedure, not the text of Q.1218, Section 3.1.1.5.6.

*/

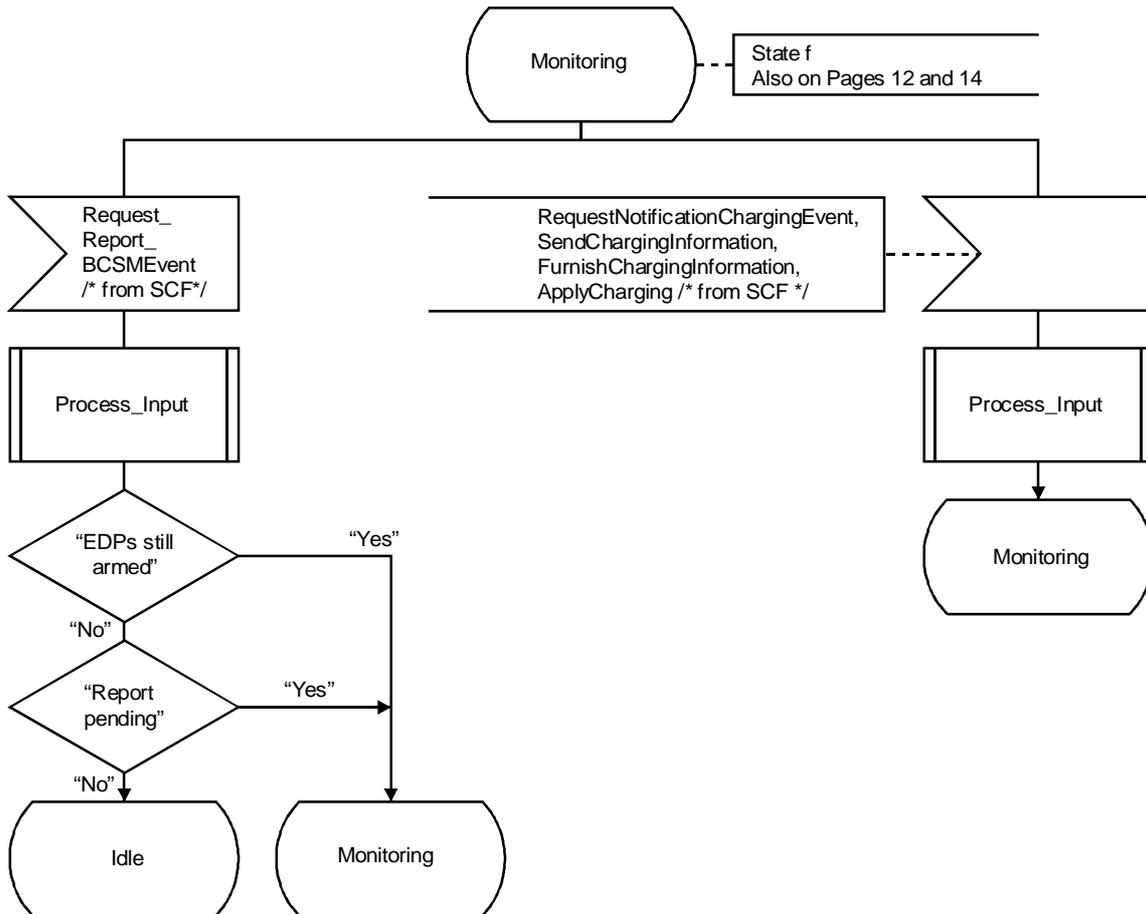


T1171880-95/d55

FIGURE A.1/Q.1218 (feuillet 12 de 17)

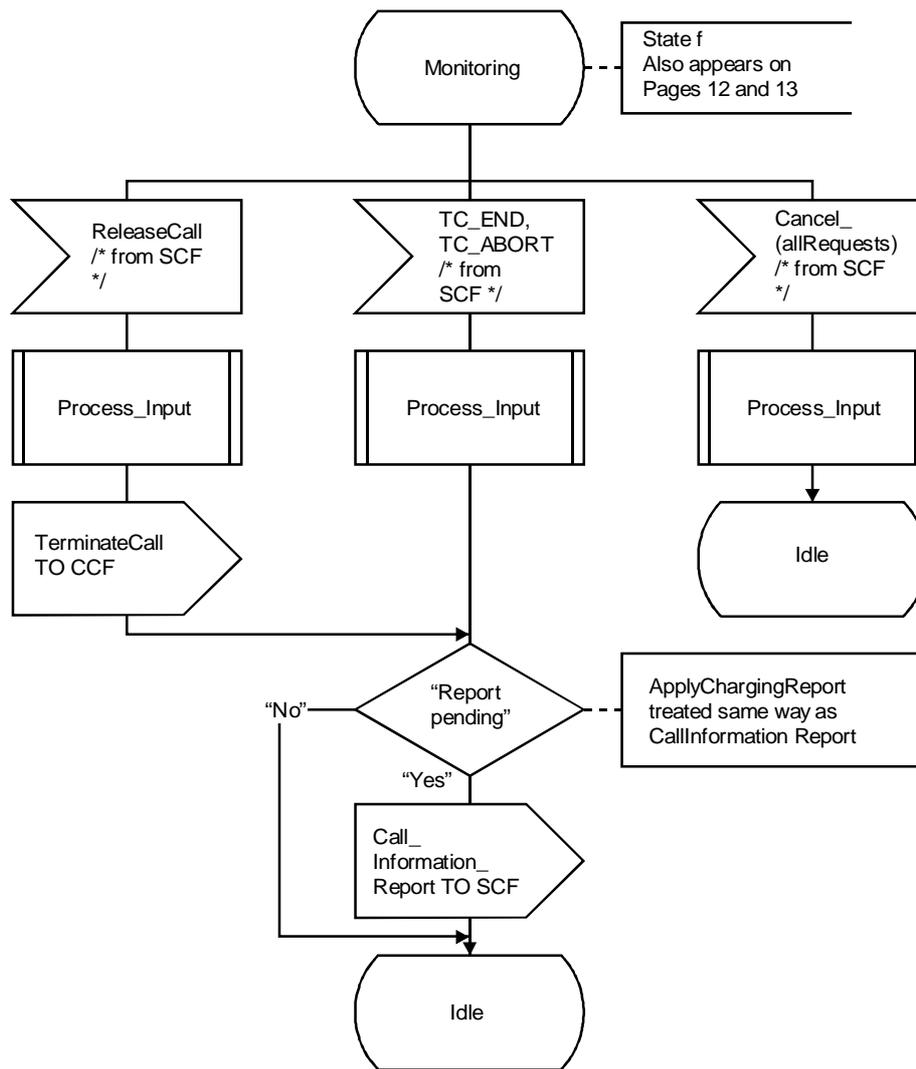
Diagramme SDL pour SSF-FSM

/* The reports which can be pending here are CallInformationReport, ApplyChargingReport and EventNotificationCharging
RequestReportBCSMEvent and RequestNotificationChargingEvent can terminate monitoring for a previously requested event.
*/



T1171890-95/d56

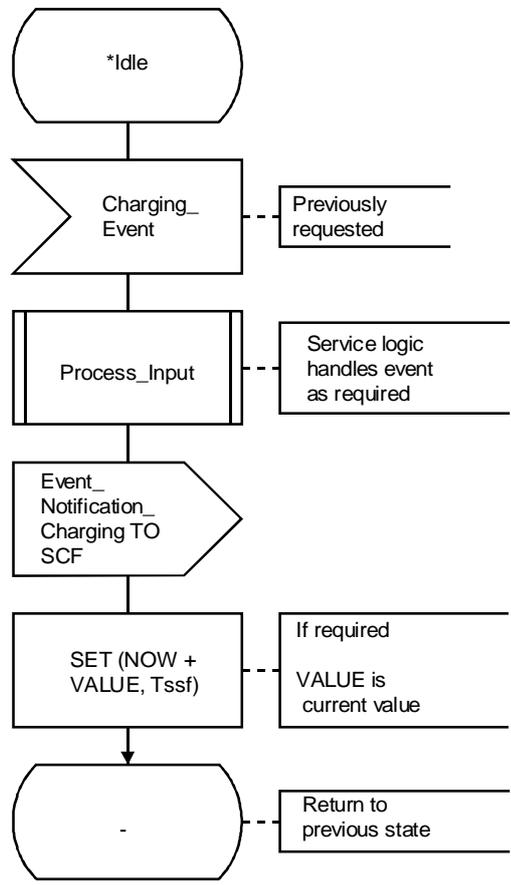
FIGURE A.1/Q.1218 (feuillet 13 de 17)
Diagramme SDL pour SSF-FSM



T1171900-95/d57

FIGURE A.1/Q.1218 (feuillet 14 de 17)
Diagramme SDL pour SSF-FSM

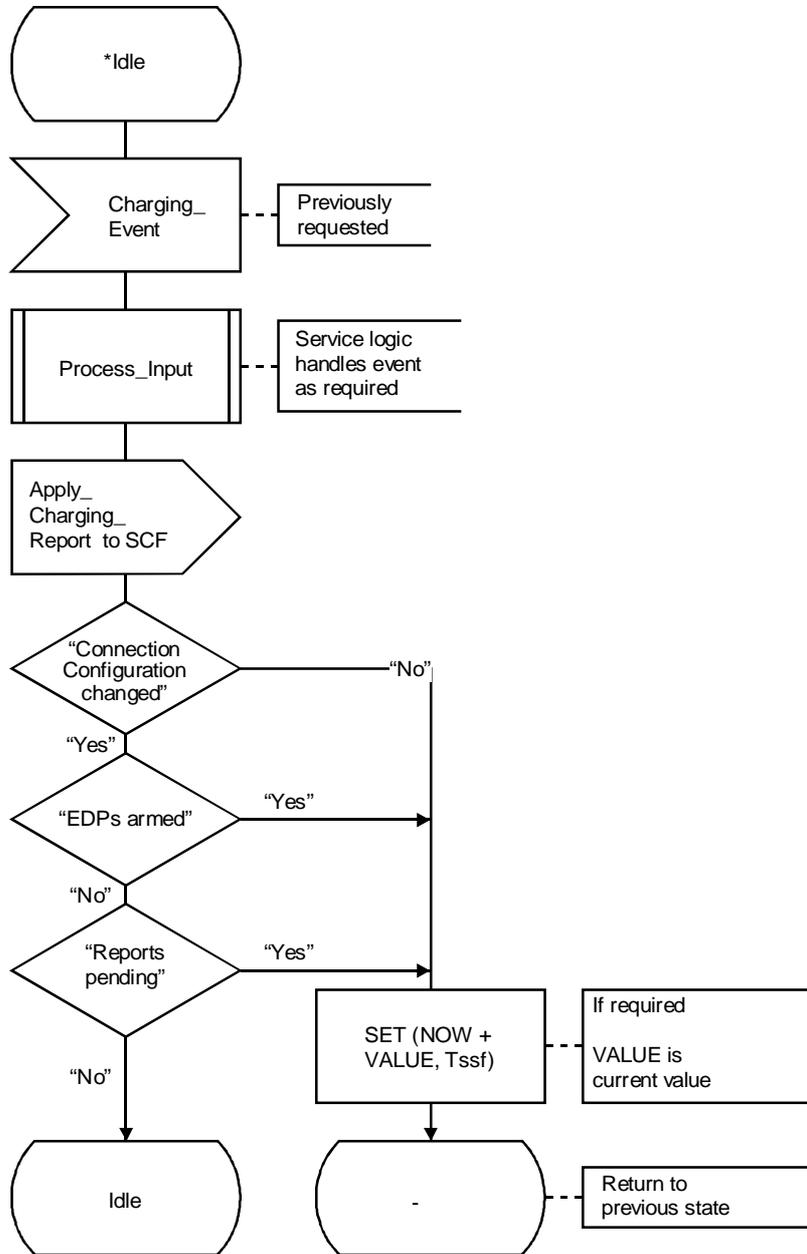
/* Charging events requested by RequestNotificationChargingEvent can be handled in any state except Idle */



T1171910-95/d58

FIGURE A.1/Q.1218 (feuillet 15 de 17)
Diagramme SDL pour SSF-FSM

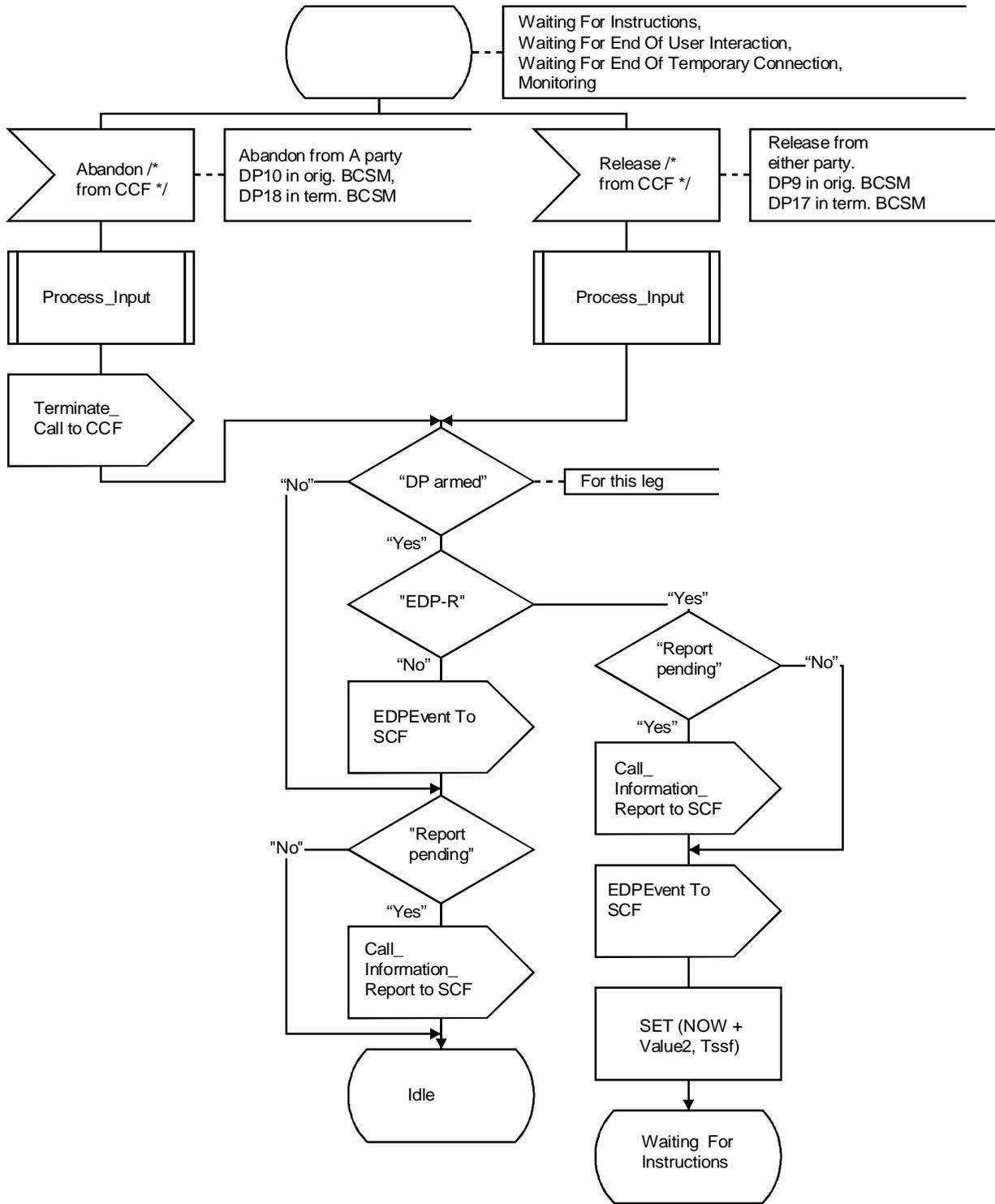
/* Charging events requested by ApplyCharging can be handled in any state except Idle */



T1171920-95/d59

FIGURE A.1/Q.1218 (feuille 16 de 17)
Diagramme SDL pour SSF-FSM

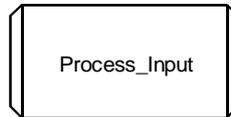
/*
 Cleardown sequences for
 - A party abandon
 - A and B party disconnection */



T1171930-95/d60

FIGURE A.1/Q.1218 (feuillet 17 de 17)
 Diagramme SDL pour SSF-FSM

```
/*  
SDL for Assist/Handoff SSF for INAP  
Based on Q.1218, Section 3.1.1.6  
  
Version 3.1 January 1995  
*/
```



```
/* Data declarations */  
  
Timer Tssf;
```

```
/* The identification of the possible values of Tssf is given in the SDL  
diagrams for the SSF.  
  
Reference Q.1218, Section 3.1.1.5  
*/
```

```
/*  
A locally defined procedure Process_Input is used to indicate analysis of an input from an external source to  
determine whether service logic processing (outside the SSF FSM) is required.  
  
No details of possible processing in the procedure are given, as it is intended only to indicate that processing  
may be required, not its exact nature.  
*/
```

FIGURE A.2/Q.1218 (feuillet 1 de 8)
Diagramme SDL pour SSF-FSM assistante/transfert de tâche

```

/* Signal definitions – First part.
These signals to and from the CCF are internal indications which are not defined in the
IN CS-1 Recommendations.
The names are therefore local names only.
*/

```

```

/* From CCF */

```

```

SIGNAL AssistRequired, BackwardConnectionReleased,
ForwardConnectionReleased;

```

```

/* To CCF */

```

```

SIGNAL ConnectSRF, RouteToDefault, SRFReleaseRequest,
TerminateCall;

```

```

/* Indication from SRF representing bearer signalling release request */
SIGNAL ReleaseRequestFrom SRF;

```

```

/* Signal definitions – Second part. Operations defined in IN CS-1
Recommendations. */

```

```

/* To SCF */

```

```

SIGNAL AssistRequestInstructions;

```

```

/* From SCF */

```

```

SIGNAL ConnectTo Resource, ApplyCharging, FurnishChargingInformation,
SendChargingInformation, ResetTimer, DisconnectForwardConnection,
ReleaseCall;

```

```

/* Signal definitions – Third part. Operations defined in
IN CS-1 Recommendations. */

```

```

/* Relay from SCF to SRF */

```

```

SIGNAL Cancel, PlayAnnouncement, PromptAndCollectUserInformation;

```

```

/* Relay from SRF to SCF */

```

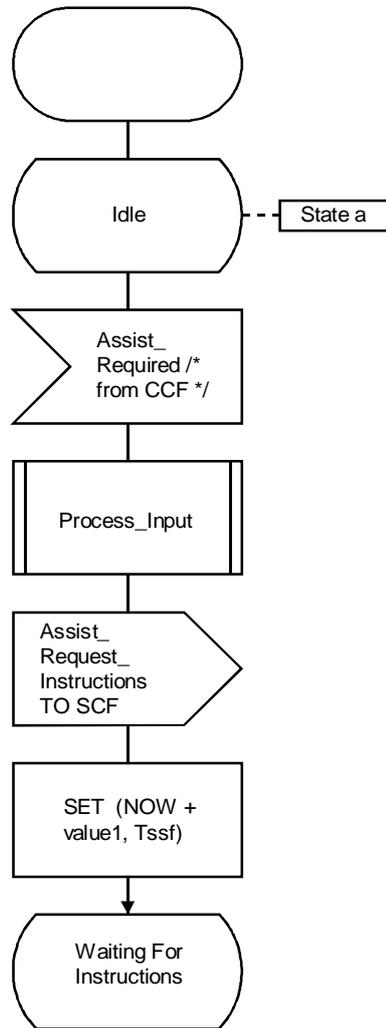
```

Signal ReturnResult_from_PromptAndCollectUserInformation,
SpecializedResourceReport;

```

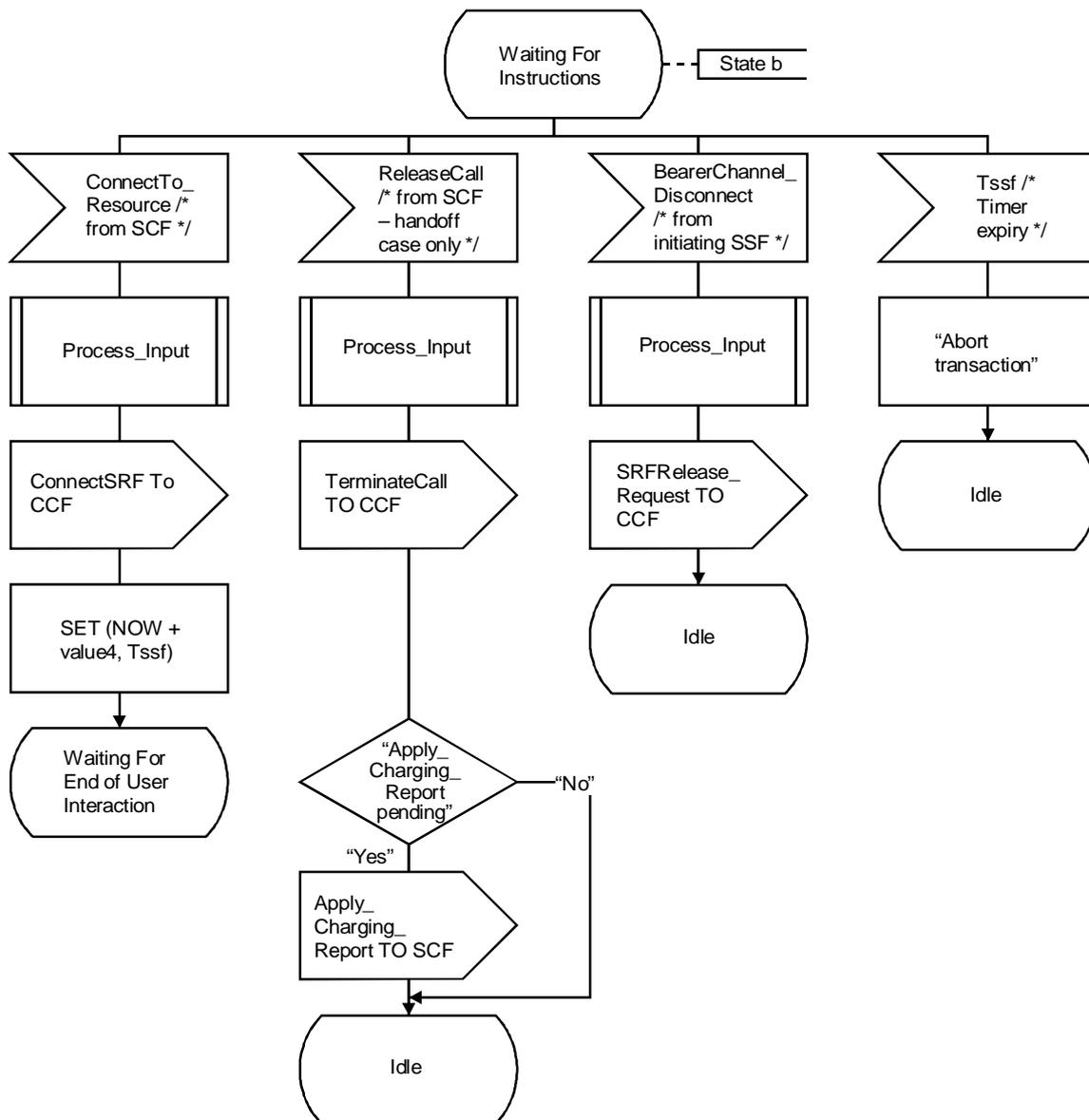
FIGURE A.2/Q.1218 (feuillet 2 de 8)

Diagramme SDL pour SSF-FSM assistante/transfert de tâche



T1171960-95/d63

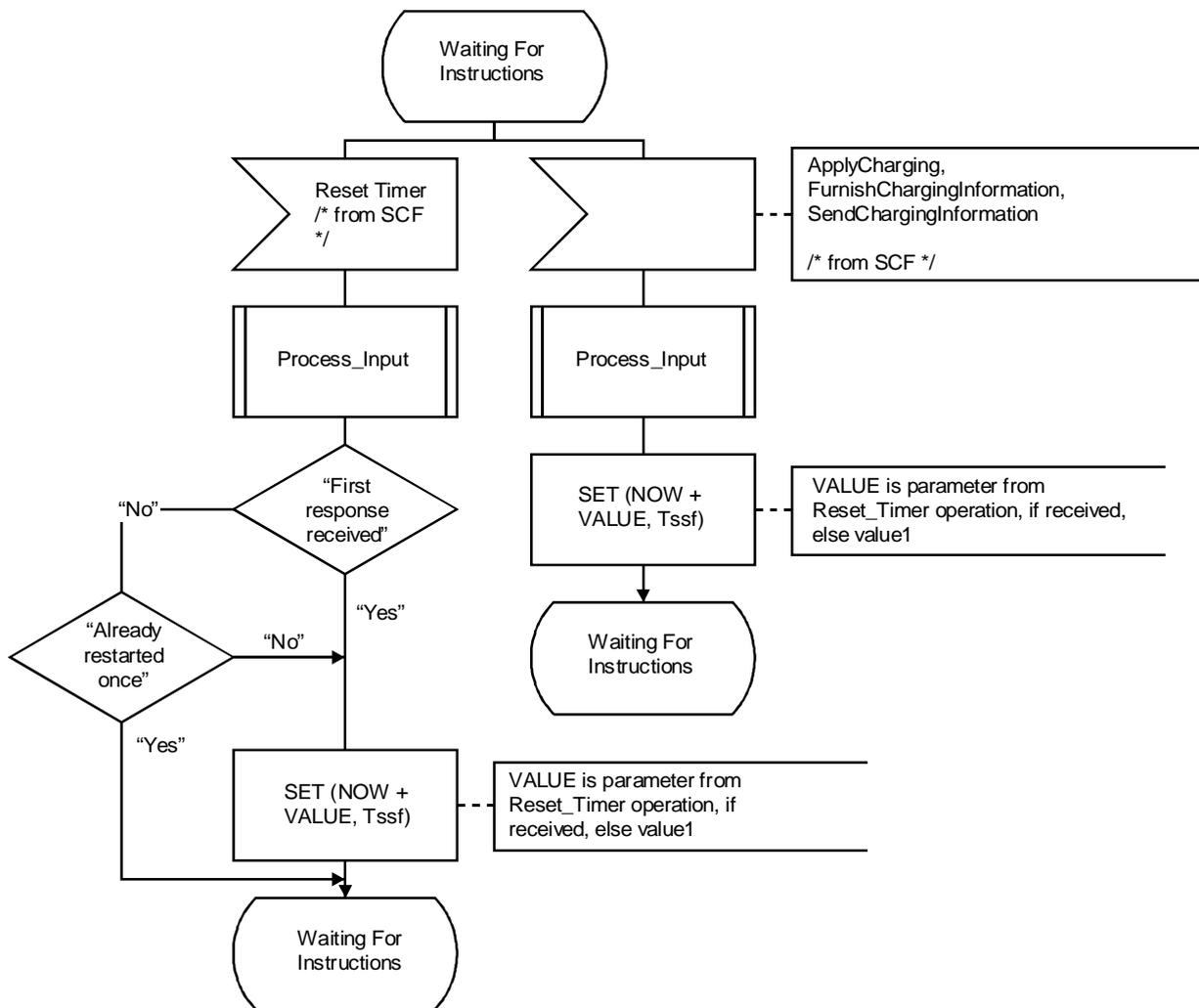
FIGURE A.2/Q.1218 (feuillet 3 de 8)
Diagramme SDL pour SSF-FSM assistante/transfert de tâche



T1171970-95/d64

FIGURE A.2/Q.1218 (feuille 4 de 8)

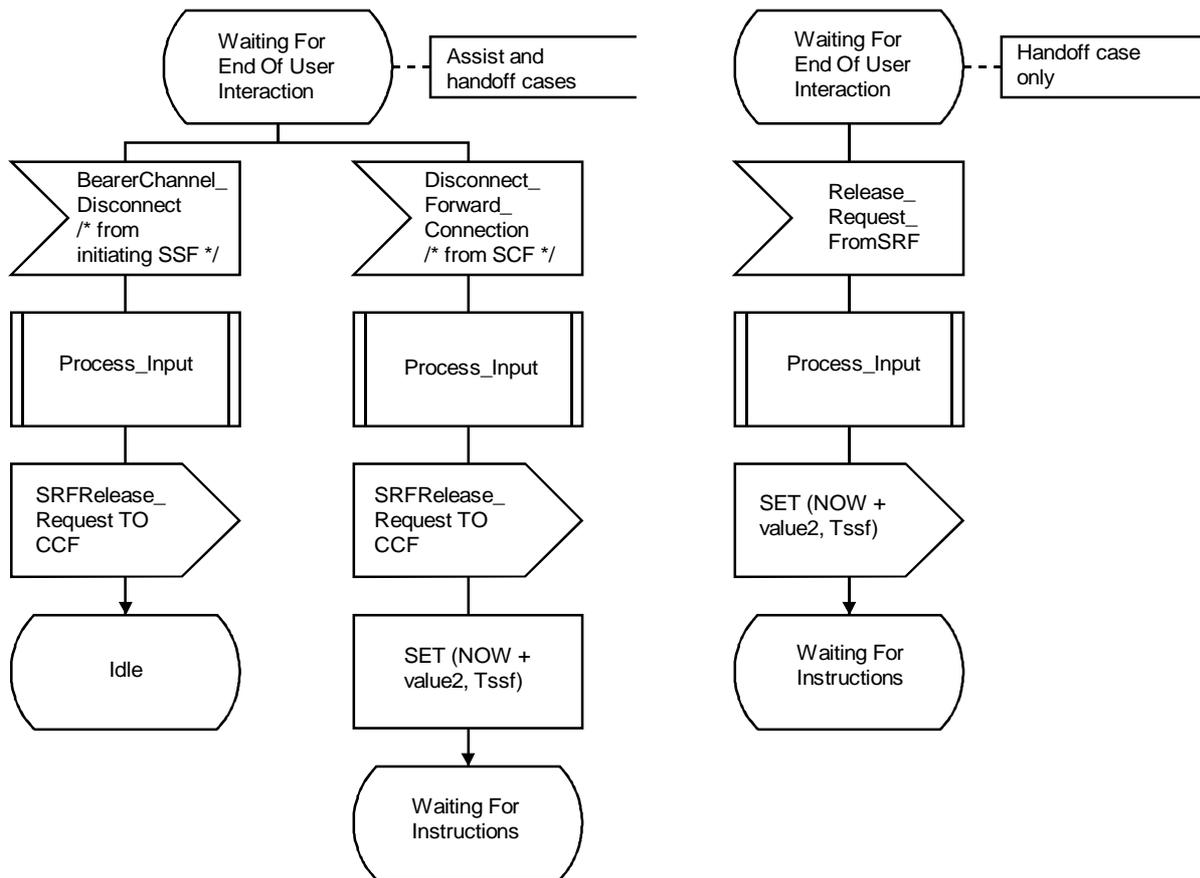
Diagramme SDL pour SSF-FSM assistante/transfert de tâche



T1171980-95/d65

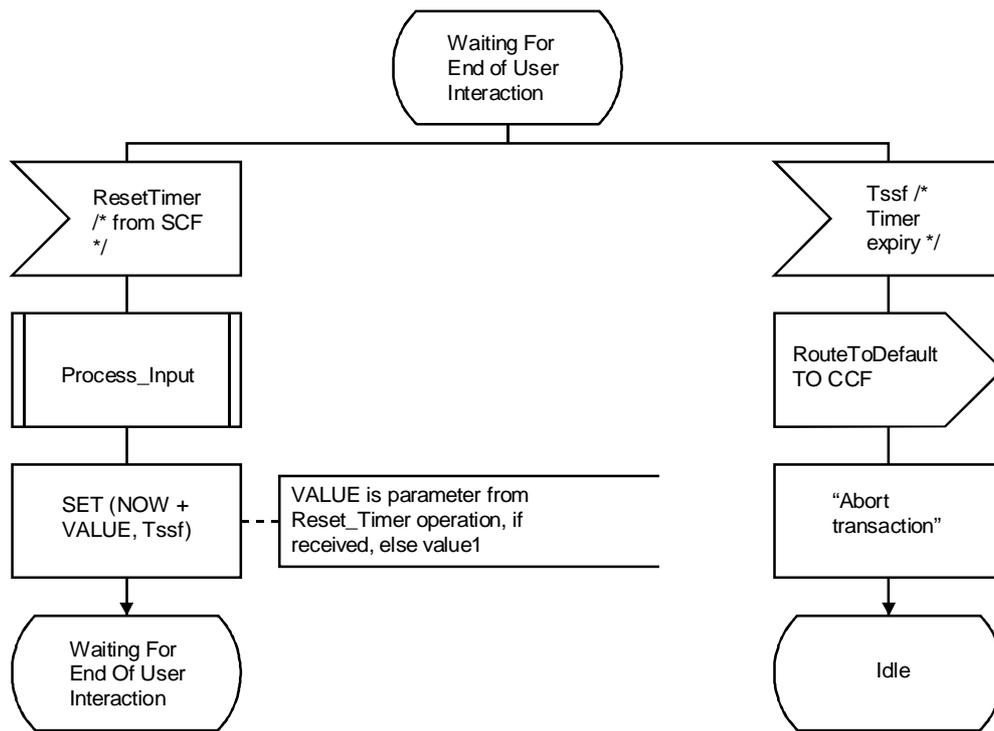
FIGURE A.2/Q.1218 (feuille 5 de 8)
Diagramme SDL pour SSF-FSM assistante/transfert de tâche

/*
 In the handoff case, the Handoff SSF returns to Waiting For Instructions state at the end of user interaction (i.e. when the SRF has been disconnected) and then takes over the call. From that point on, the SDL becomes that shown for Process INAP_SSF. This has not been shown here.
 */



T1172000-95/d67

FIGURE A.2/Q.1218 (feuillet 7 de 8)
 Diagramme SDL pour SSF-FSM assistante/transfert de tâche



T1172010-95/d68

FIGURE A.2/Q.1218 (feuille 8 de 8)
Diagramme SDL pour SSF-FSM assistante/transfert de tâche

```
/*  
SCF SDLs for INAP.  
Based on Q.1218, Section 3.1.2.5, "The SCSM".  
  
Version 2.1 January 1995  
*/
```

Process_Input

```
/* Data declarations */  
Timer Tscf_ssf, Tassist_handoff, Tq;
```

```
/* Timer Tscf_ssf can have four different values, as described in Q.1218, Section 3.1.2.5.  
These are denoted valueA, valueB, valueC and valueD.  
Timer assist_handoff is given valueE. Queuing timer Tq is given valueQ.  
  
*/
```

```
/*  
Procedure Process_Input analyses an input from an external source to determine  
processing required. It can invoke SCF service logic.  
  
Output Operation TO SSF or TO SRF sends one or more of the signals listed in the  
attached comment to the SSF or SRF, respectively.  
  
Output Exception TO SSF indicates that various types of system failure are to be  
handled.  
  
*/
```

FIGURE A.3/Q.1218 (feuillet 1 de 20)
Diagramme SDL pour SCF-FSM

/* Signal definitions – First part.
 These signals from (internal) SCF logic are all internal indications which are defined as internal events in the IN CS-1 Recommendations.
 The names are based on the event names.

*/

/* From SCF service logic */
 SIGNAL SL_Invocation, Non-Call_Processing_Instructions, SR_Facilities_Needed,
 Call_Processing_Instruction_Ready_e2.3, Call_Processing_Instruction_Ready_e2.4,
 Ready_for_Queueing_Processing, Processing_Failure, Instruction_Ready,
 Instruction_Ready_Relay, Assist_Needed, Hand_off_Needed,
 More_Information_Needed, Continue_SCF_Processing_e4.3,
 Continue_SCF_Processing_e4.3', Continue_SCF_Processing_e4.3' ',
 Cancellation_Required;

/* Signal definitions – First part continued */

/* From SCF service logic */
 SIGNAL Busy_Line/Trunk, Idle_Line/Trunk,
 Notification_or_Request_Continuing_Instruction,
 Monitoring_Cancel_Instruction, Release_Call_Instruction_e2.14,
 Release_Call_Instruction_e2.15;

/* Signal definitions – Second part.
 These signals to and from external FEs are indications which are not defined in the IN CS-1 Recommendations. The names are therefore local names only.

*/

/* Local names defined for signals SSF to SCF */
 SIGNAL TDPEvent; /* Indicates one of InitialDP or a DP-specific operation */
 SIGNAL EDPEvent; /* Indicates one of EventReportBCSM or a DP-specific operation */

/* To SSF */
 SIGNAL Exception;

/* From SSF */
 SIGNAL Abort_from_SSF, Initial_SSF_Failure;

/* From SRF */
 SIGNAL Failure_from_SRF;

FIGURE A.3/Q.1218 (feuillet 2 de 20)

Diagramme SDL pour SCF-FSM

/* Signal definitions – Third part. Defined in IN CS-1 Recommendations. */

/* From SSF or SRF */
SIGNAL AssistRequestInstructions;

/* From SSF – DP-specific operations */

SIGNAL AnalysedInformation, CollectedInformation, OAnswer,
OCalledPartyBusy, ODisconnect, OMidCall, ONoAnswer,
OriginationAttemptAuthorized, RouteSelectFailure, TAnswer,
TCalledPartyBusy, TDisconnect, TermAttemptAuthorized, TMidCall,
TNoAnswer;

/* Other operations from SSF */
SIGNAL InitialDP, EventReportBCSM, ApplyChargingReport,
CallInformationReport, EventNotificationCharging;

/* Signal definitions – Fourth part. Defined in IN CS-1 Recommendations. */

/* To SSF */
SIGNAL ResetTimer, ApplyCharging, CallInformationRequest,
Cancel, FurnishChargingInformation, RequestNotificationChargingEvent,
RequestReportBCSMEvent, SendChargingInformation,
AnalyseInformation, CollectInformation, Connect, Continue, ReleaseCall,
SelectFacility, SelectRoute, ConnectToResource,
EstablishTemporaryConnection;

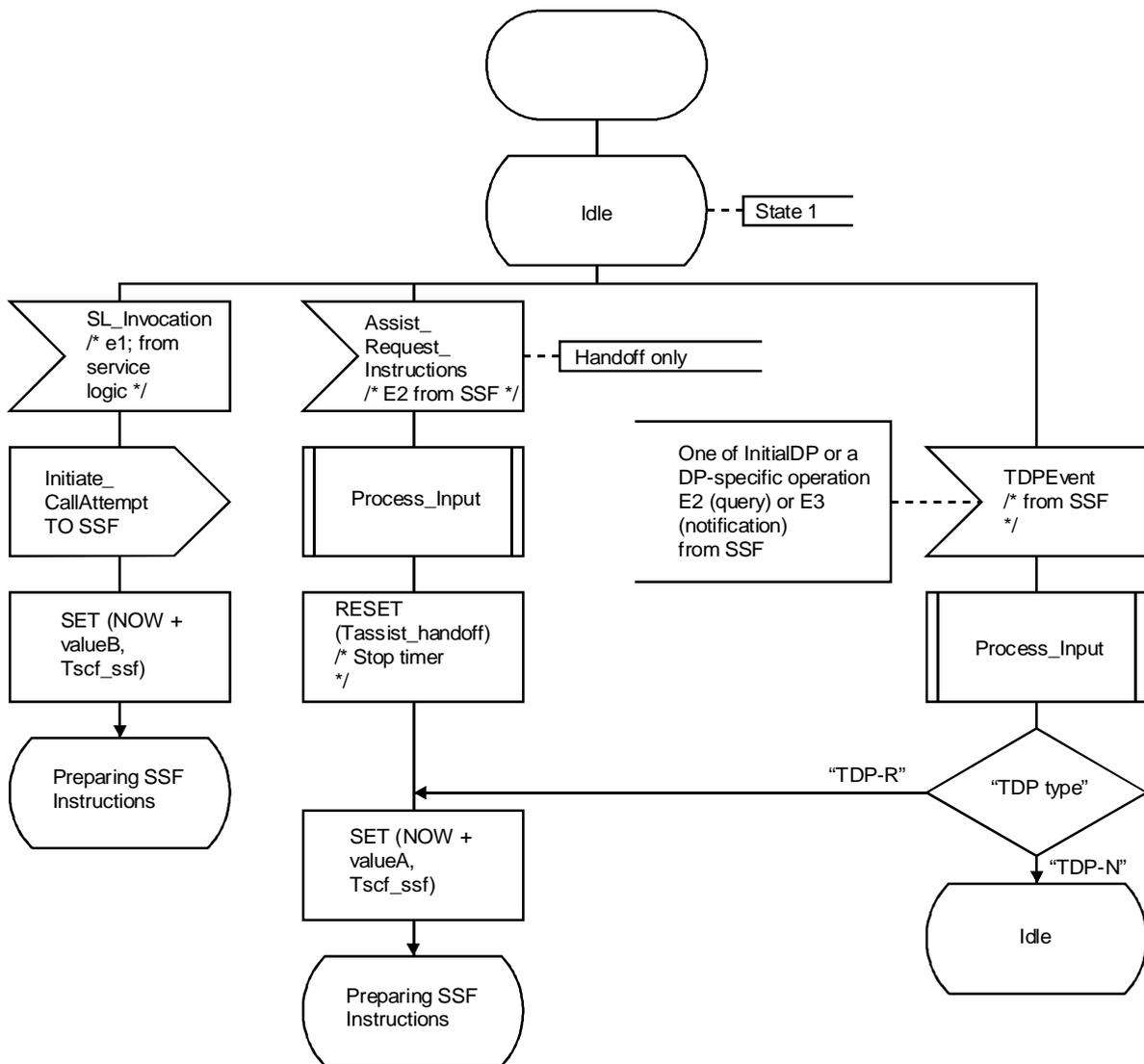
/* Signal definitions – Fifth part. Operations defined in IN CS-1 Recommendations. */

/* To SRF */
SIGNAL PlayAnnouncement, PromptAndCollectUserInformation;

/* From SRF */
SIGNAL ReturnResult_from_PromptAndCollectUserInformation,
SpecializedResourceReport;

FIGURE A.3/Q.1218 (feuille 3 de 20)

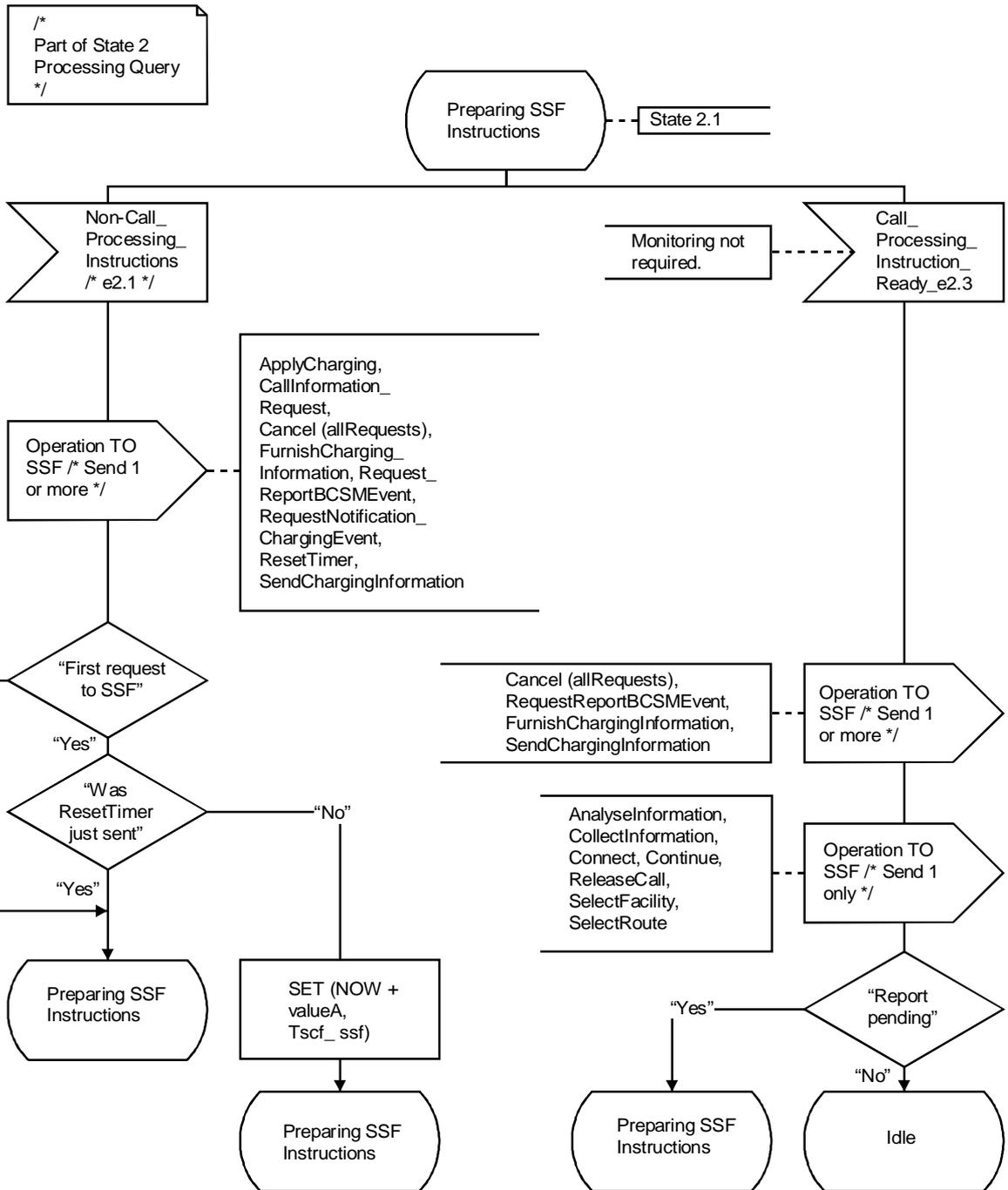
Diagramme SDL pour SCF-FSM



T1172050-95/d72

FIGURE A.3/Q.1218 (feuillet 4 de 20)

Diagramme SDL pour SCF-FSM

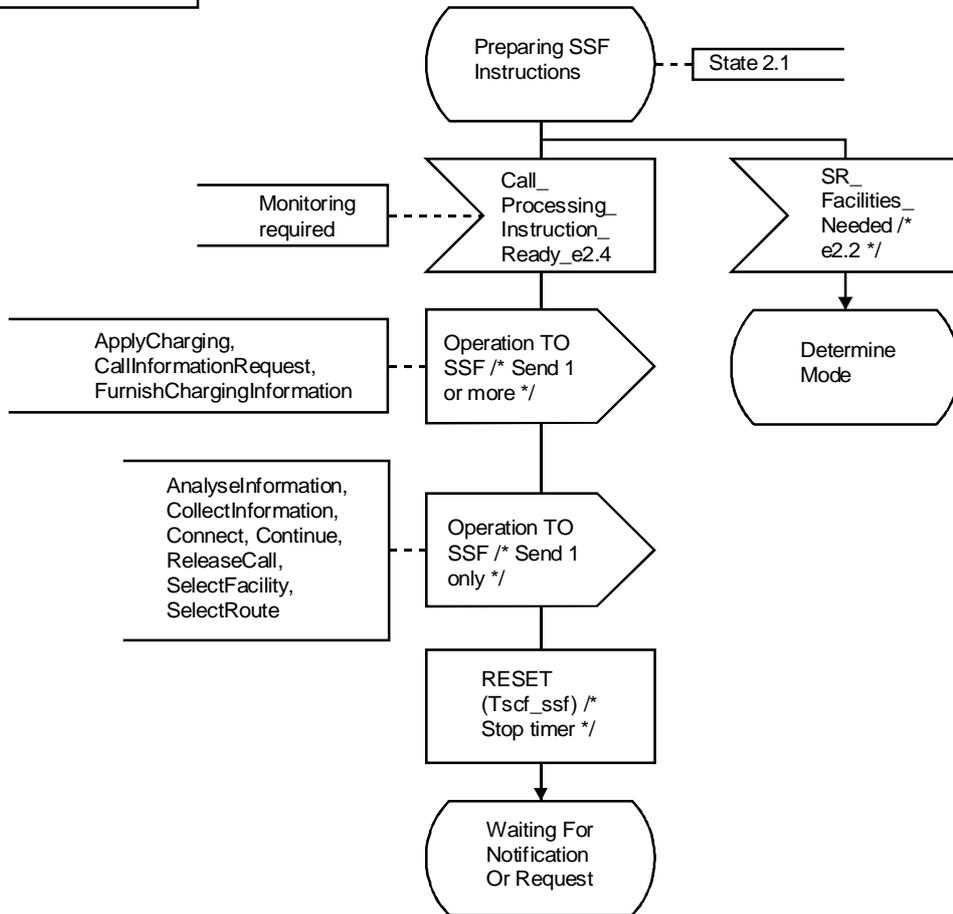


T1172060-95/d73

FIGURE A.3/Q.1218 (feuille 5 de 20)

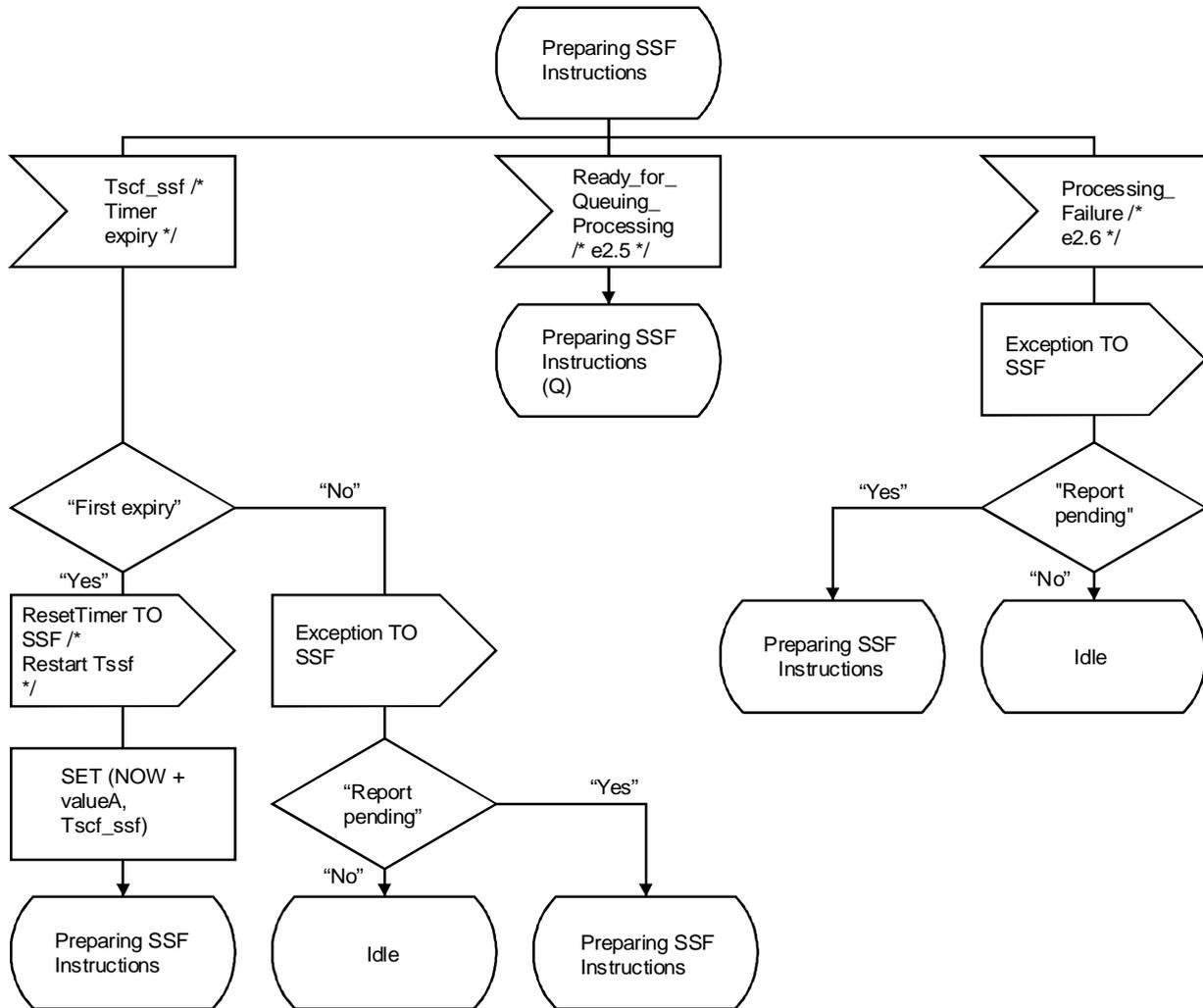
Diagramme SDL pour SCF-FSM

/*
Part of State 2
Processing Query
*/



T1172070-95/d74

FIGURE A.3/Q.1218 (feuillet 6 de 20)
Diagramme SDL pour SCF-FSM

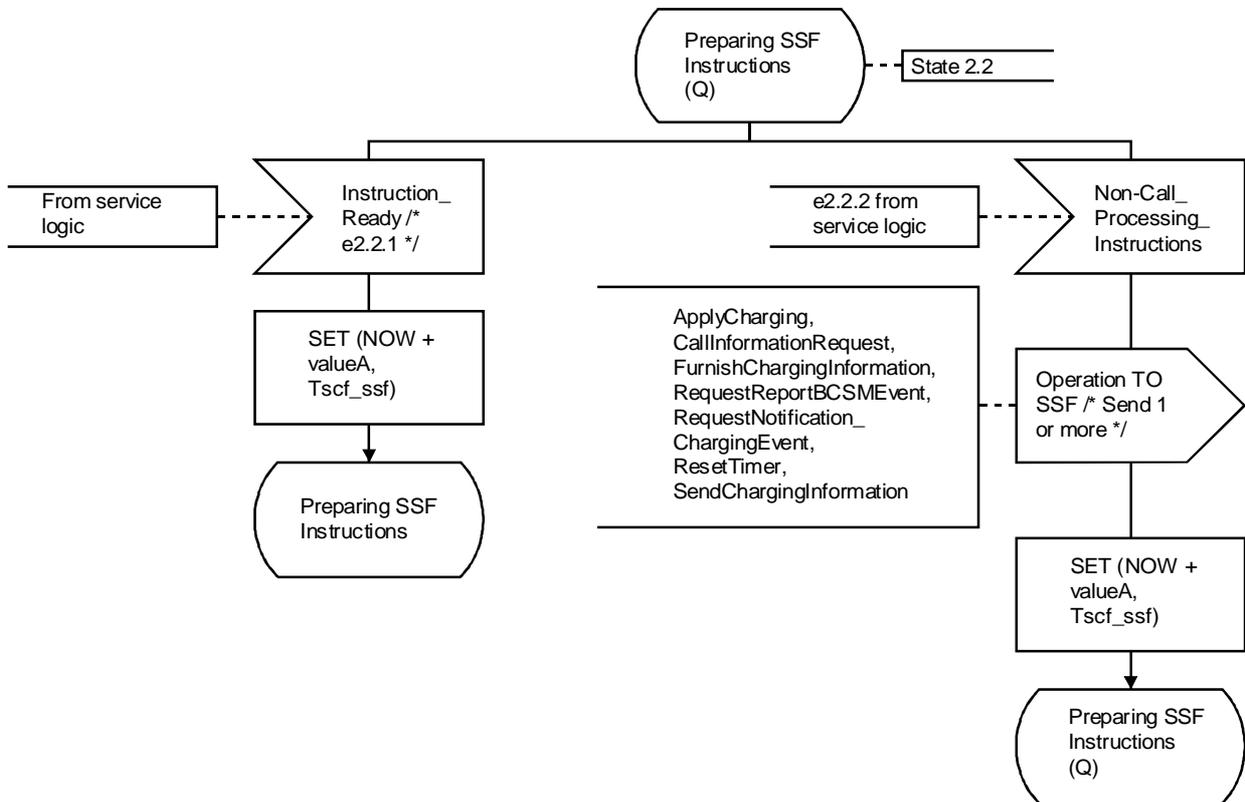


T1172080-95/d75

FIGURE A.3/Q.1218 (feuillet 7 de 20)

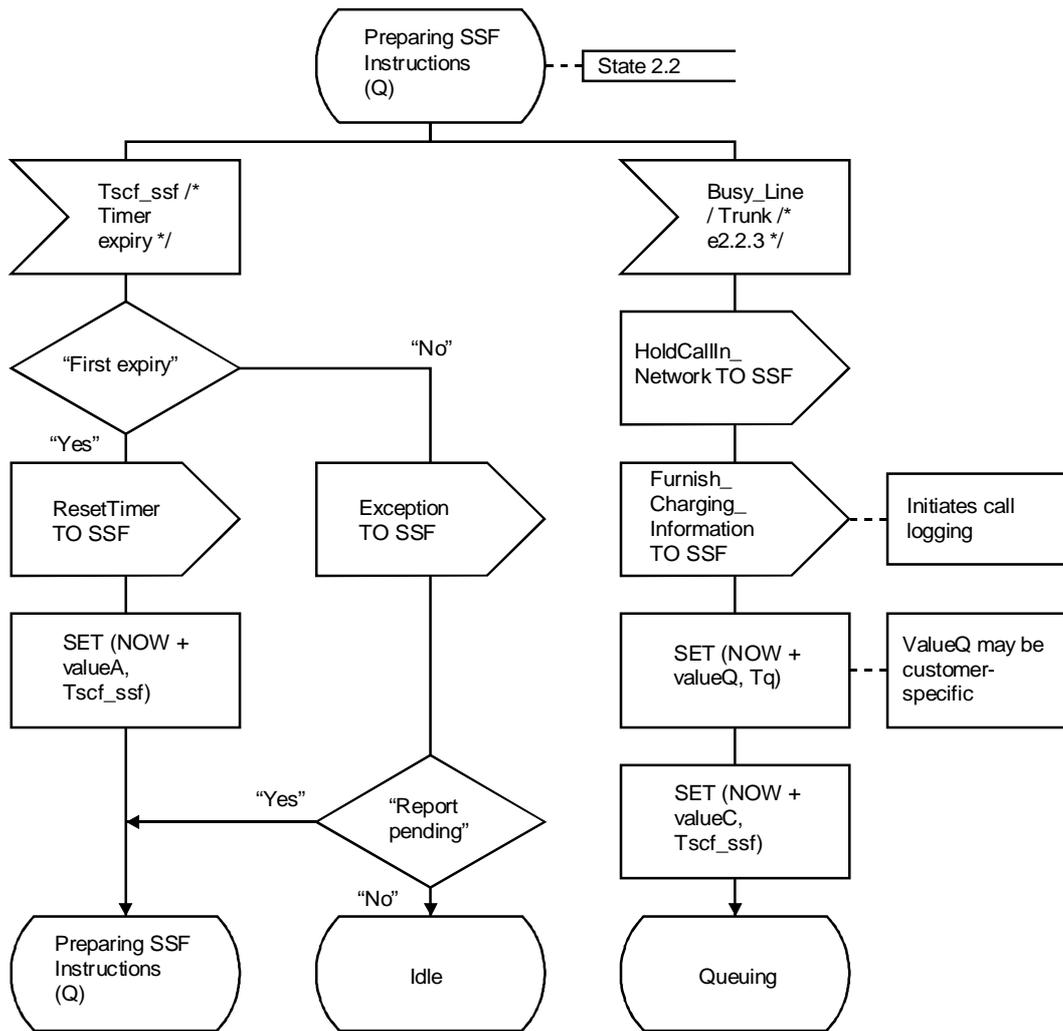
Diagramme SDL pour SCF-FSM

/*
 The Queuing FSM is not described in detail in Q.1218.
 Part of State 2 Processing Query
 */



T1172090-95/d76

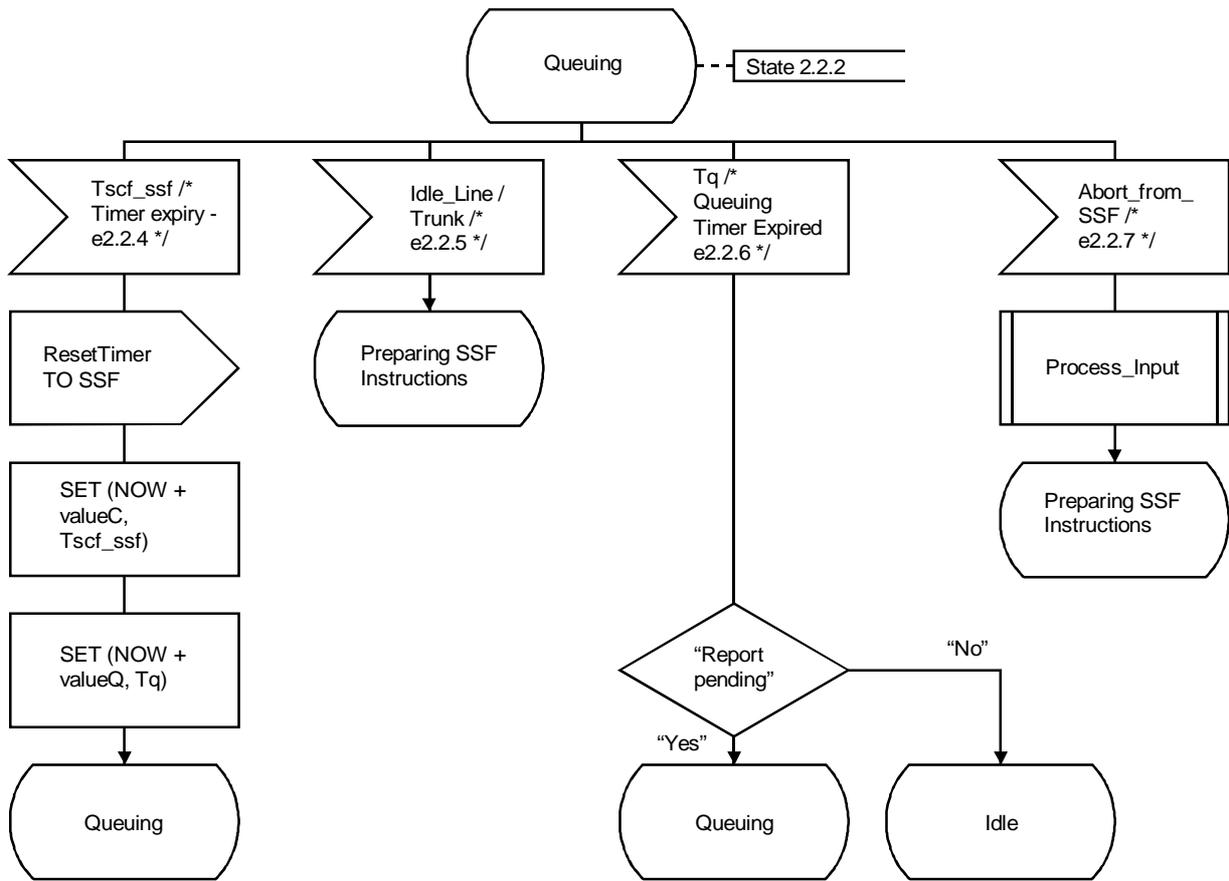
FIGURE A.3/Q.1218 (feuillet 8 de 20)
 Diagramme SDL pour SCF-FSM



T1172100-95/d77

FIGURE A.3/Q.1218 (feuillet 9 de 20)

Diagramme SDL pour SCF-FSM



T1172110-95/d78

FIGURE A.3/Q.1218 (feuillet 10 de 20)

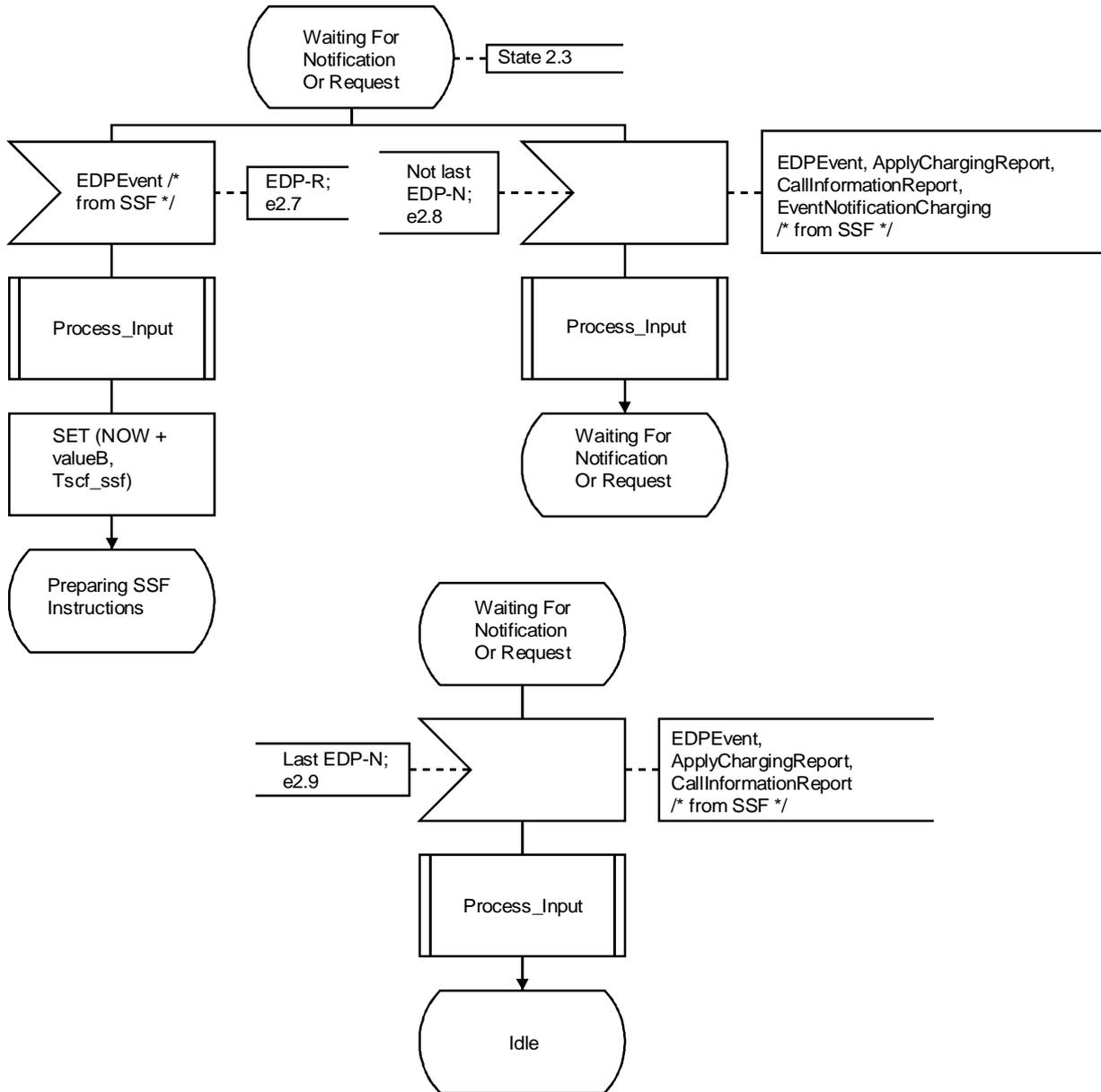
Diagramme SDL pour SCF-FSM

```

/*
Part of State 2 Processing_Query.
Timer Tscf_ssf not used in this State.

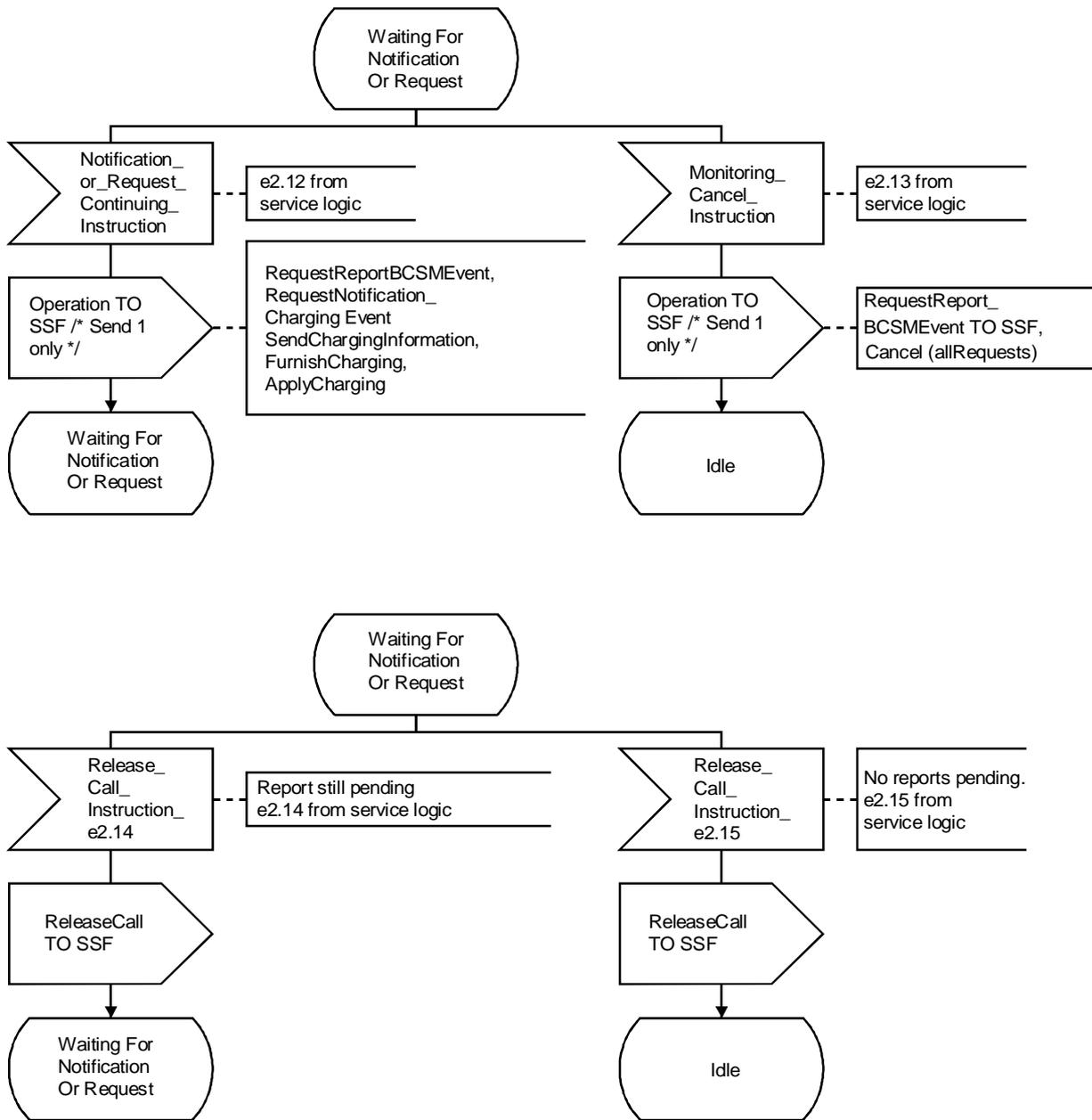
EDPEvent indicates reception of one of InitialDP or
DP-specific operation from SSF.

*/
    
```



T1172120-95/d79

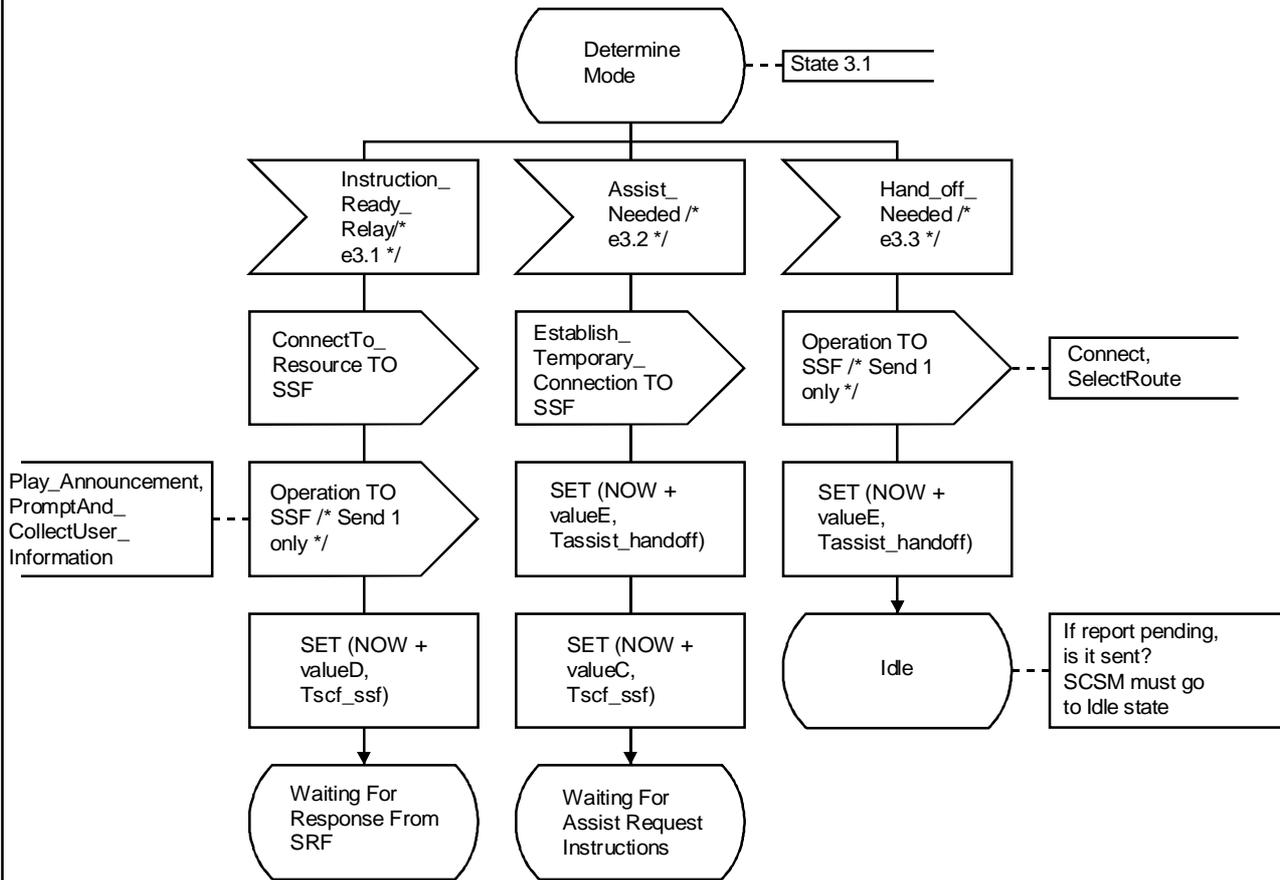
FIGURE A.3/Q.1218 (feuillet 11 de 20)
Diagramme SDL pour SCF-FSM



T1172130-95/d80

FIGURE A.3/Q.1218 (feuillet 12 de 20)
Diagramme SDL pour SCF-FSM

/*
Part of State 3
Routing To Resource.
*/

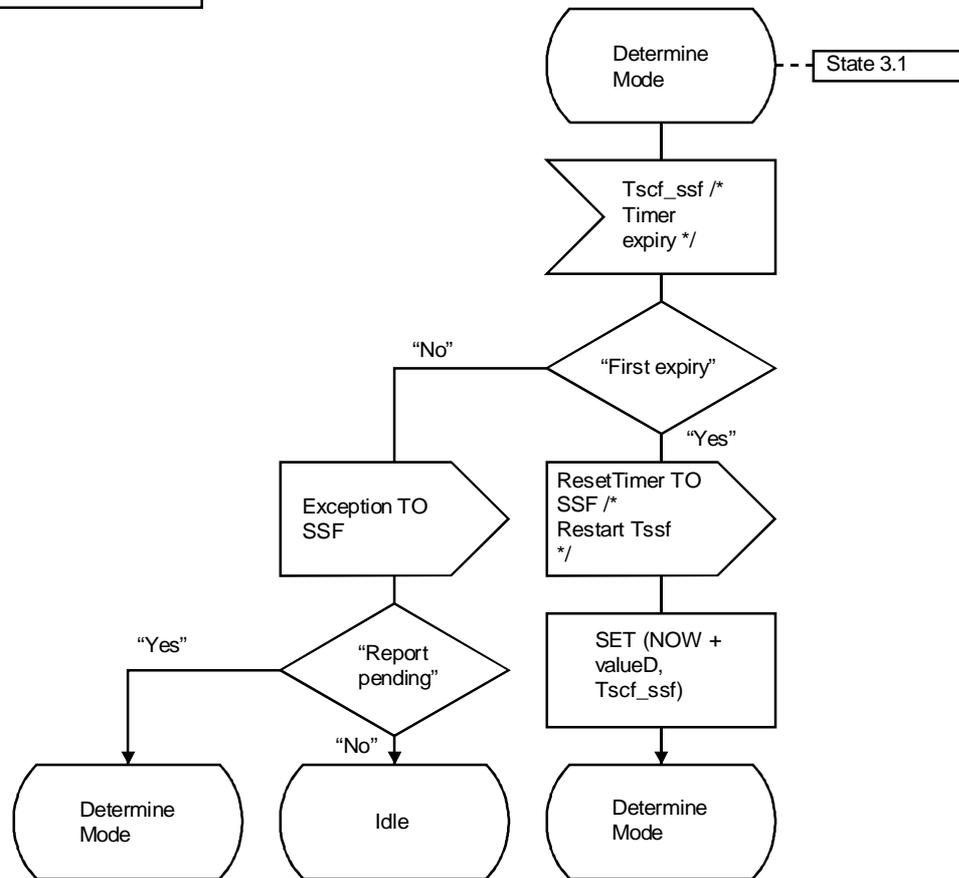


T1172140-95/d81

FIGURE A.3/Q.1218 (feuillet 13 de 20)

Diagramme SDL pour SCF-FSM

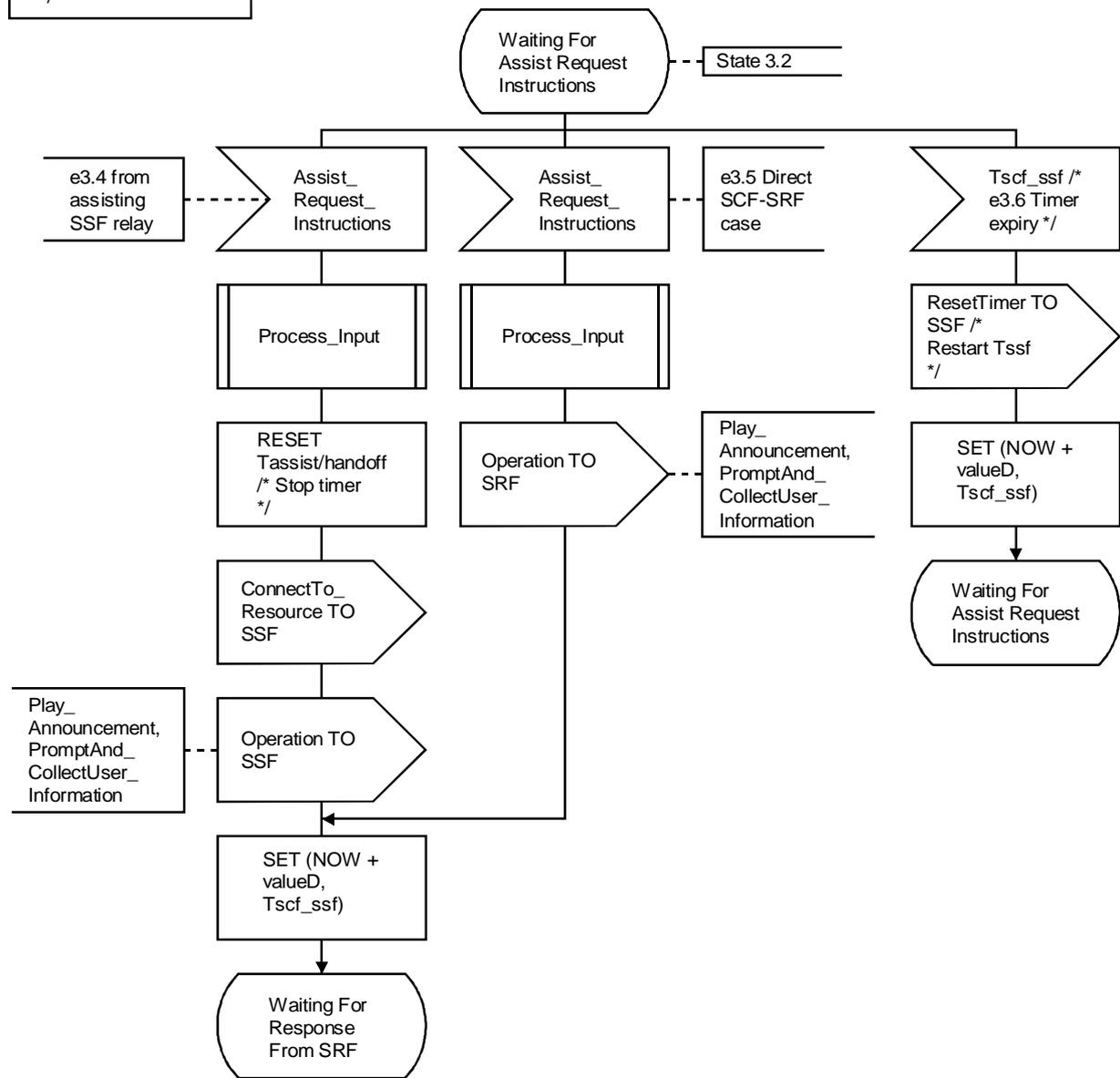
/*
 Part of State 3
 Routing To Resource.
 Timer Tscf_ssf expiry is not
 explicitly shown in Q.1218.
 */



T1172150-95/d82

FIGURE A.3/Q.1218 (feuillet 14 de 20)
Diagramme SDL pour SCF-FSM

/*
Part of State 3
Routing To Resource
*/

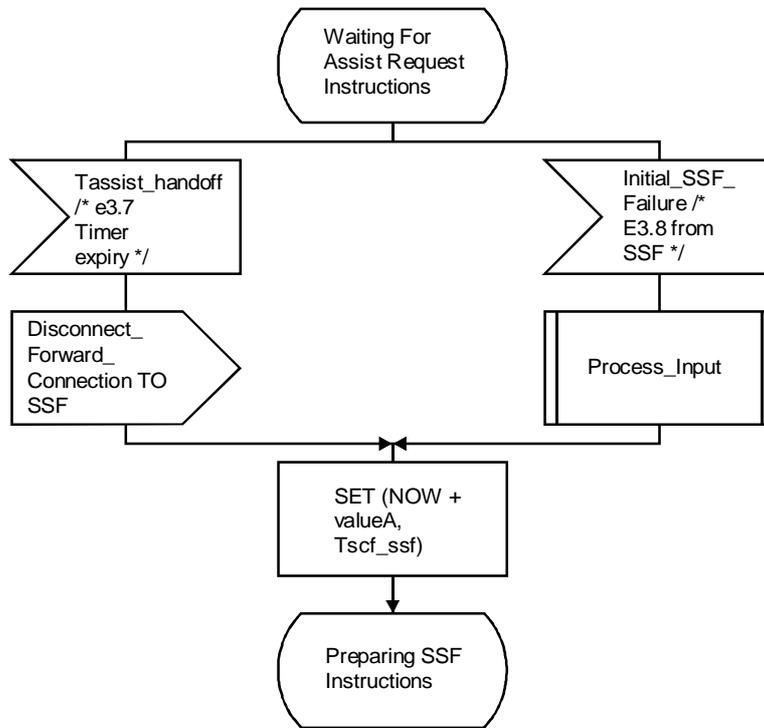


T1172160-95/d83

FIGURE A.3/Q.1218 (feuillet 15 de 20)

Diagramme SDL pour SCF-FSM

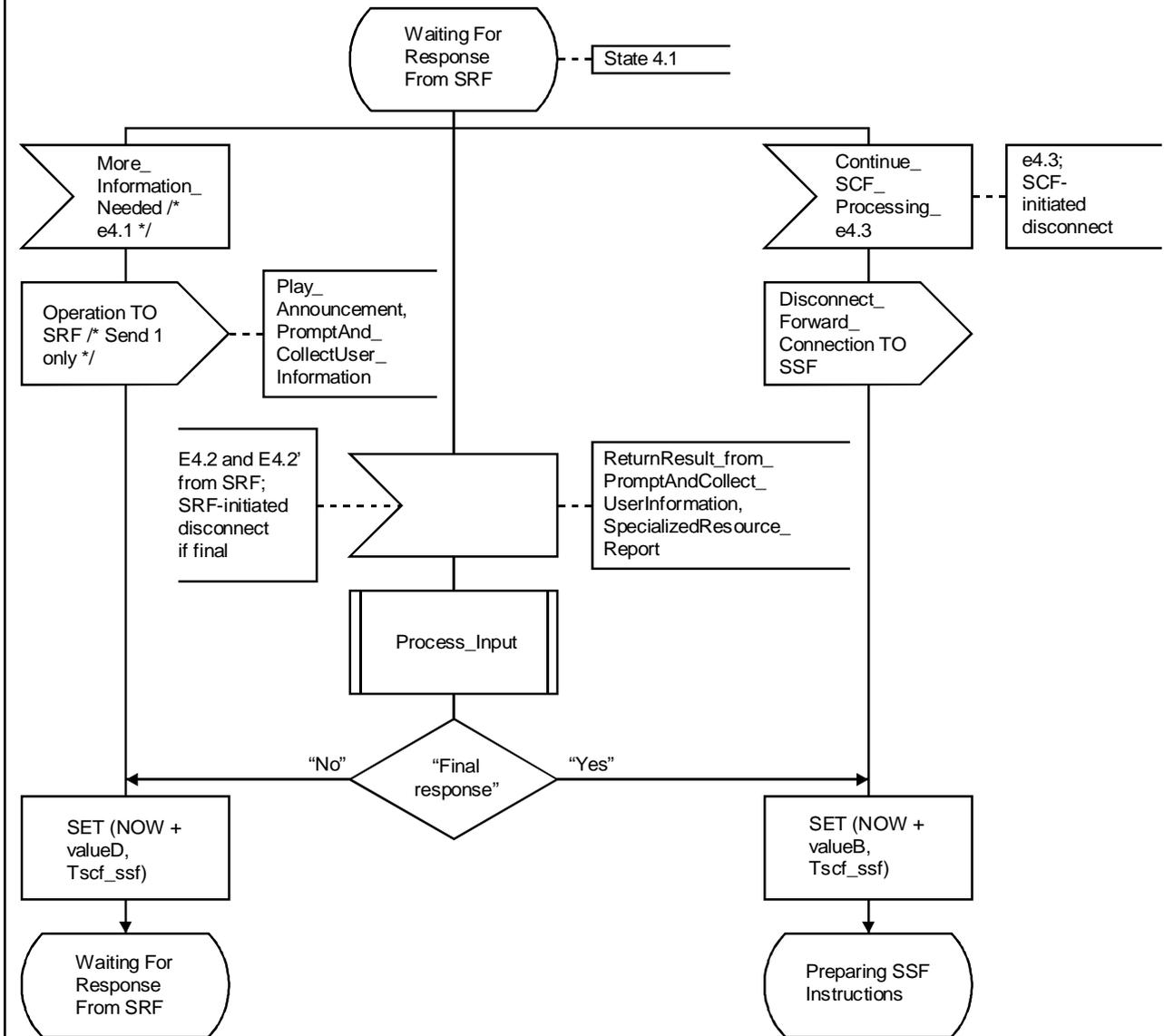
/*
Part of State 3
Routing To Resource
*/



T1172170-95/d84

FIGURE A.3/Q.1218 (feuillet 16 de 20)
Diagramme SDL pour SCF-FSM

/*
 Part of State 4 User Interaction
 Output operations to SRF do not distinguish
 between direct SCF-SRF and SSF relay cases.
 */

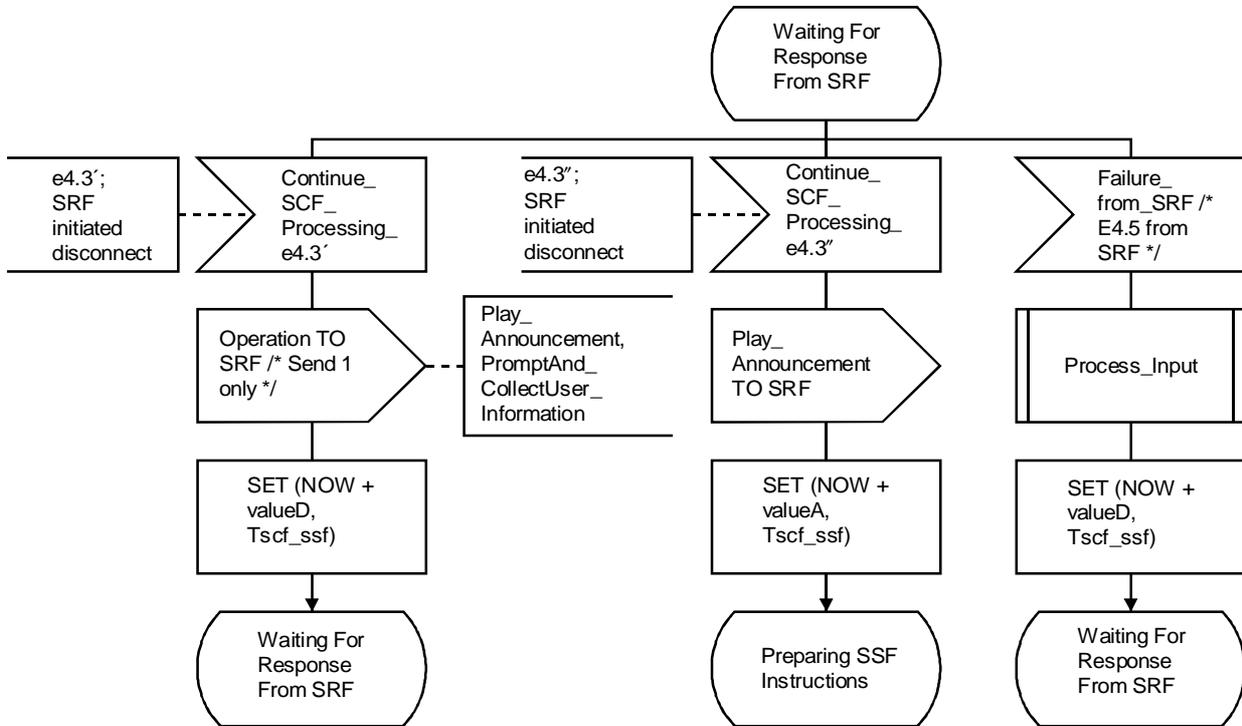


T1172180-95/d85

FIGURE A.3/Q.1218 (feuillet 17 de 20)

Diagramme SDL pour SCF-FSM

/*
 Part of State 4 User Interaction
 Output operations to SRF do not distinguish
 between direct SCF-SRF and SSF relay cases.
 */

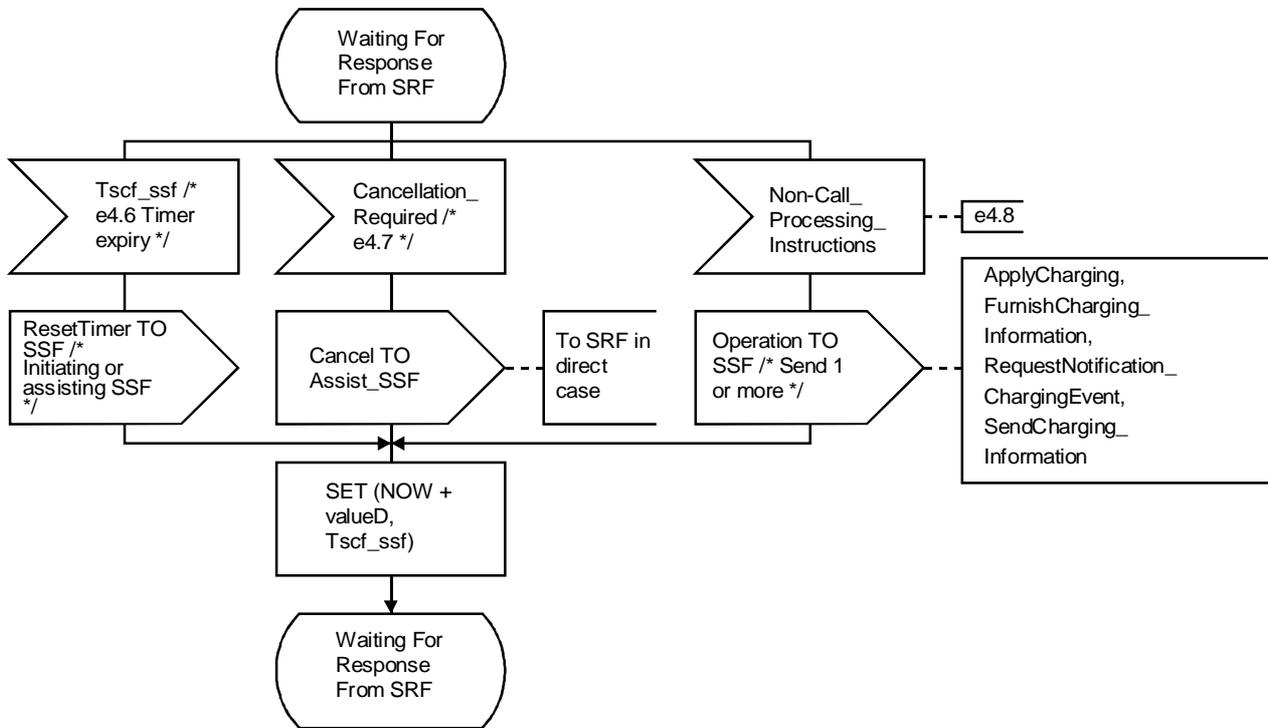


T1172190-95/d86

FIGURE A.3/Q.1218 (feuille 18 de 20)

Diagramme SDL pour SCF-FSM

/*
Part of State 4 User Interaction
*/

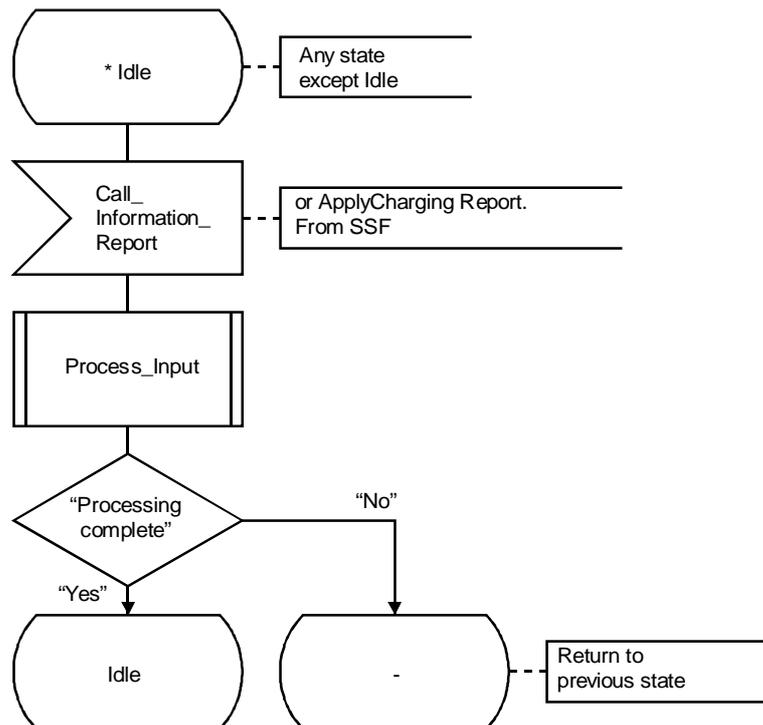


T1172200-95/d87

FIGURE A.3/Q.1218 (feuille 19 de 20)

Diagramme SDL pour SCF-FSM

```
/* CallInformationReport and ApplyChargingReport can be received
in any state except Idle.
If processing has been completed and SCSM is only awaiting receipt of report,
then SCSM goes to Idle, else it remains in the previous state.
*/
```

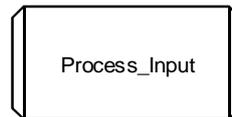


T1172210-95/d88

FIGURE A.3/Q.1218 (feuillet 20 de 20)

Diagramme SDL pour SCF-FSM

```
/*  
SRF SDLs for INAP.  
Based on Q.1218, Section 3.1.3.4, "The SRSM".  
  
Version 2.0 January 1995  
*/
```



```
/* Data declarations */  
  
Timer Tsrf;
```

```
/*  
  
A locally defined procedure Process_Input is used to indicate analysis of an input from an  
external source to determine whether service logic processing (outside the SRF FSM) is required.  
  
No details of possible processing in the procedure are given, as it is intended only to indicate that  
processing may be required, not its exact nature.  
  
The output Send_PAPC_Result TO SCF sends the appropriate SpecializedResourceReport or  
ReturnResult_from_PromptAndCollectUserInfo operation to the SCF (including  
error reports).  
  
*/
```

FIGURE A.4/Q.1218 (feuillet 1 de 7)

Diagramme SDL pour SRF-FSM

/* Signal definitions – First part.

Internal indications which are defined in the IN CS-1 Recommendations as event names, not operations. Names are therefore local names only.

*/

/* From SRF service logic */

SIGNAL Assist_Request_Instructions_Needed, SRF_Report_to_SCF;
PAPC_Cancelled_to_SCF, Cancel_Error_to_SCF, Disconnect_to_SSF;

/* Signal definitions – Second part.

Internal indications which are defined in the IN CS-1 Recommendations as event names, not operations. Names are therefore local names only.

/* From SSF */

SIGNAL Connect_Request_from_SSF, Connection_Released_from_SSF;

/* Signal definitions – Third part. Operations defined in IN CS-1 Recommendations.

*/

/* From SCF */

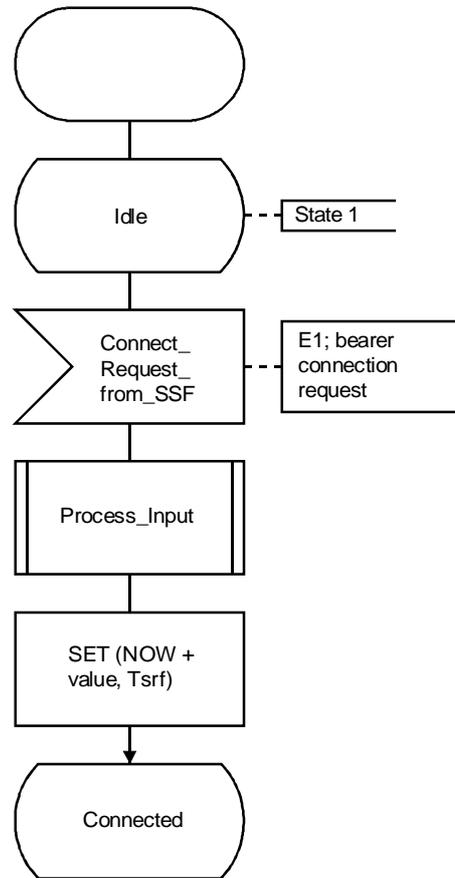
SIGNAL PlayAnnouncement, PromptAndCollectUserInformation, Cancel;

/* To SCF */

SIGNAL AssistRequestInstructions;

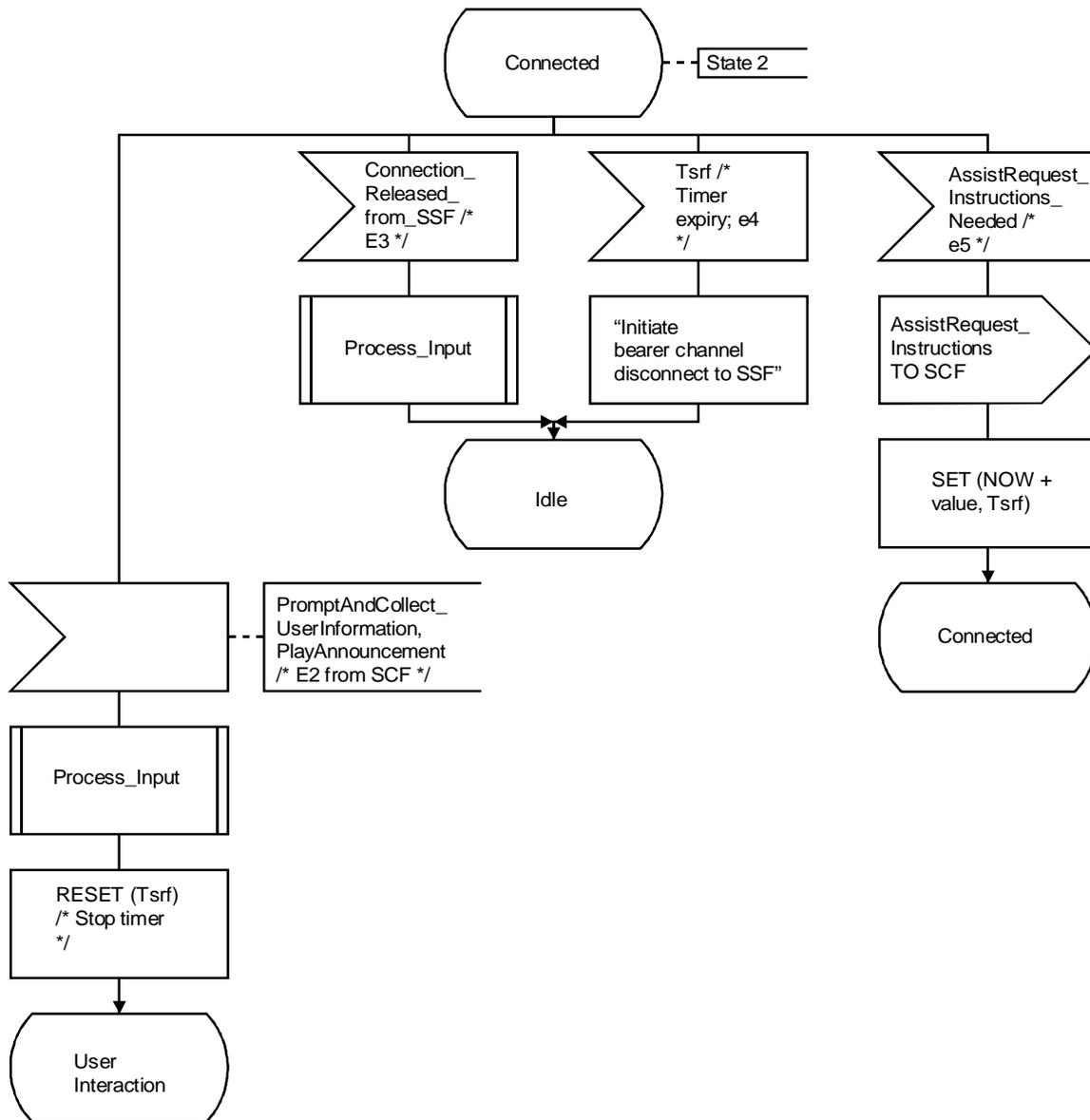
FIGURE A.4/Q.1218 (feuillet 2 de 7)

Diagramme SDL pour SRF-FSM



T1172240-95/d91

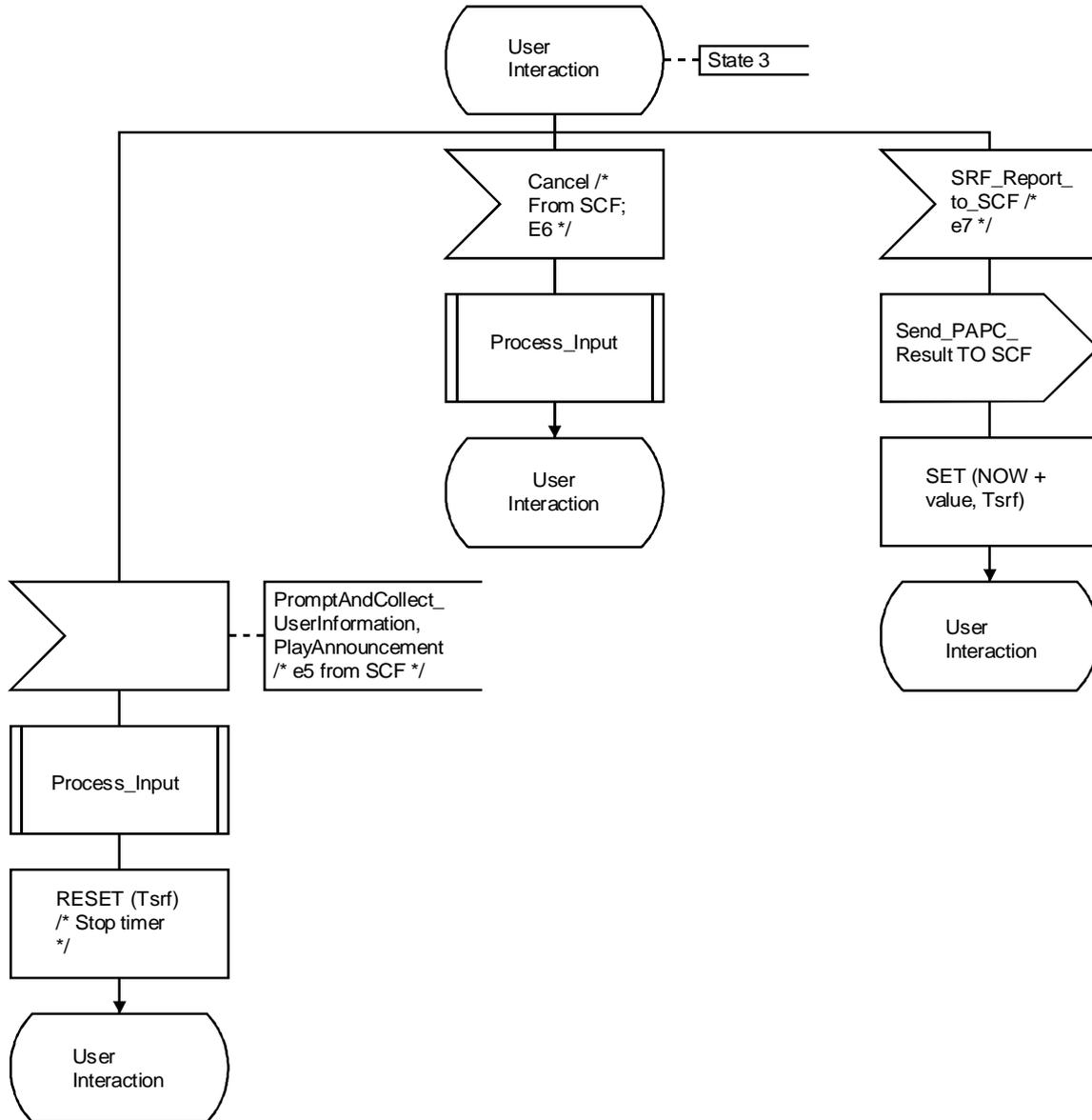
FIGURE A.4/Q.1218 (feuillet 3 de 7)
Diagramme SDL pour SRF-FSM



T1172250-95/d92

FIGURE A.4/Q.1218 (feuille 4 de 7)

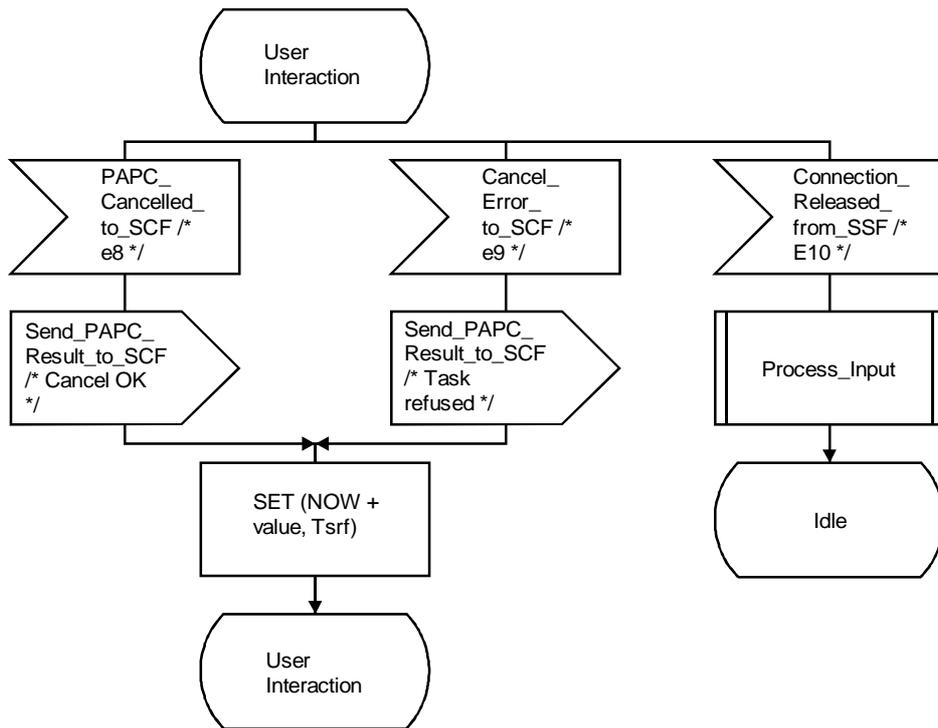
Diagramme SDL pour SRF-FSM



T1172260-95/d93

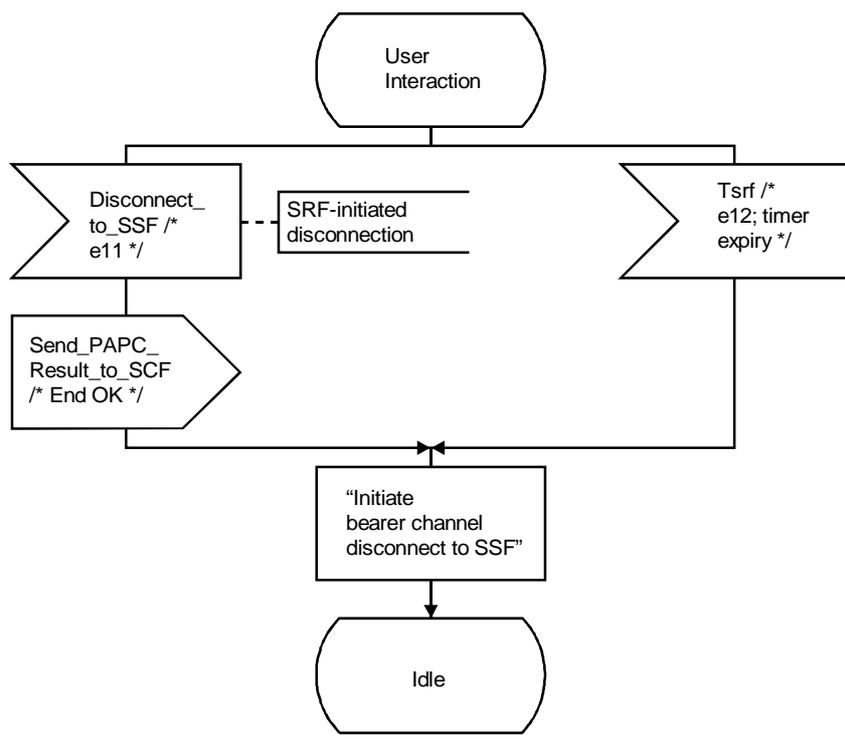
FIGURE A.4/Q.1218 (feuillet 5 de 7)

Diagramme SDL pour SRF-FSM



T1172270-95/d94

FIGURE A.4/Q.1218 (feuillet 6 de 7)
 Diagramme SDL pour SRF-FSM



T1172280-95/d95

FIGURE A.4/Q.1218 (feuillet 7 de 7)
Diagramme SDL pour SRF-FSM

Annexe B

Description des automates SCSM (Etats associés à l'entité SDF) et SDSM

(Cette annexe fait partie intégrante de la présente Recommandation)

Cette description contient un texte et une description SDL.

B.1 Description du processus SCSM

Quand le processus d'application du côté SCF décide de créer un lien entre une entité SCF et l'entité SDF pour l'utilisateur, une instance de processus SCSM sera créée par le processus d'application. Deux processus interviennent pour l'établissement d'un lien, notamment l'authentification et la négociation de version. L'authentification sera concernée par l'utilisateur et l'annuaire. La norme fournit les méthodes suivantes de techniques d'authentification:

- l'utilisation de mots de passe; ou
- l'utilisation de mots de passe protégés.

Quand une instance de processus SCSM est créée, elle est mise à l'état "Repos".

B.1.1 Etat 1 – "Repos"

B.1.1.1 Procédures normales

Quand l'instance de processus SCSM est dans l'état "Repos" et quand il est nécessaire que le processus d'application interroge une entité SDF, le processus d'application SCF enverra une primitive Bind.inv. L'argument bind sera enregistré et une temporisation T_1 sera démarrée. Cet événement provoquera une transition vers l'état "Wait_for_Subsequent_Requests" et d'autres événements seront attendus. La temporisation T_1 sera démarrée pour superviser l'état "Wait_for_Subsequent_Requests" et garantira que le processus SCSM est détruit si la communication avec le processus d'application est interrompue. Les services de gestion de dialogue du TC seront utilisés pour effectuer l'opération DirectoryBind et pour déclencher l'envoi de l'APDU associé à l'instance du processus de l'entité SDF homologue.

B.1.1.2 Procédures exceptionnelles

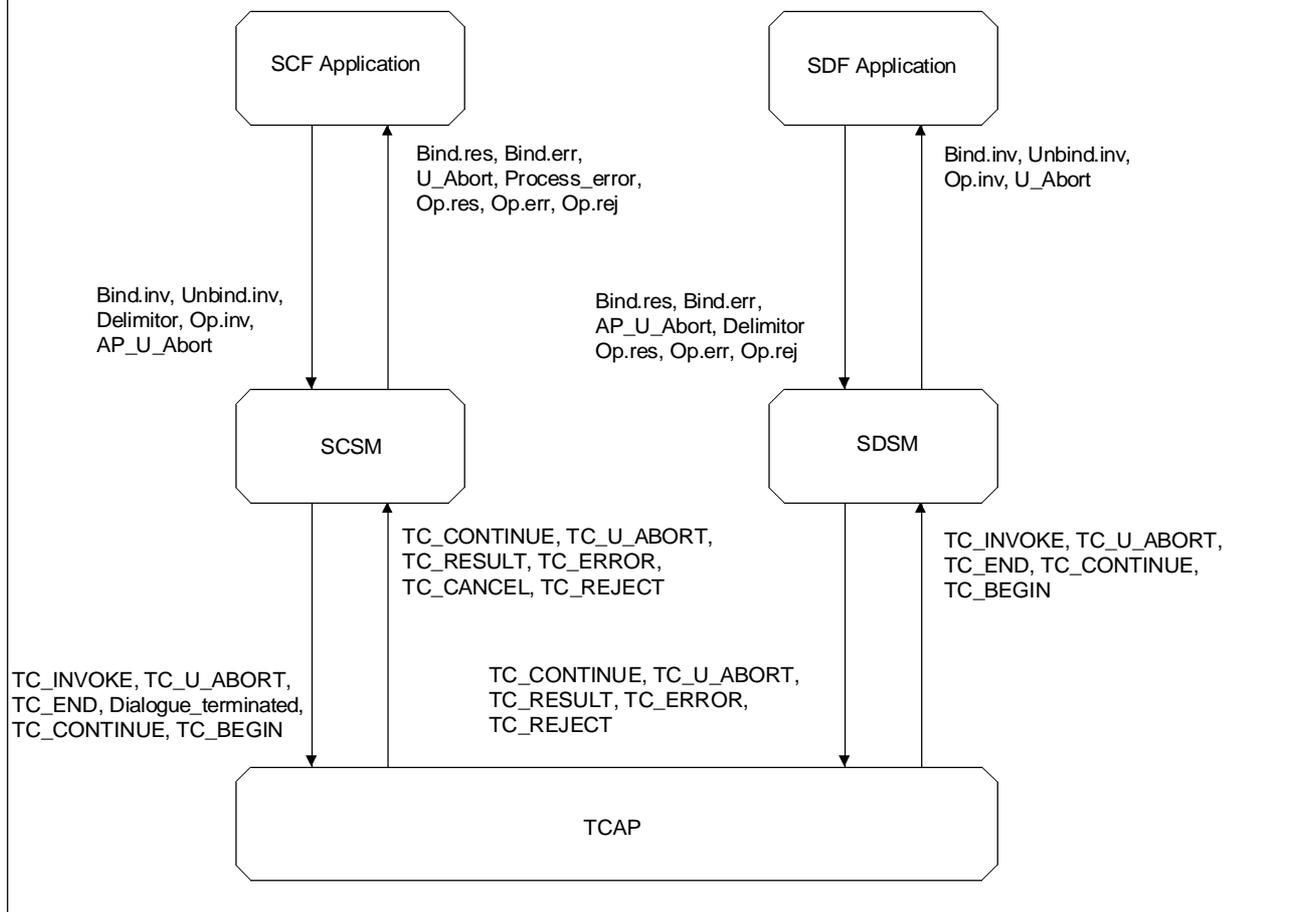
Aucune identifiée.

B.1.2 Etat 2 – "Wait for subsequent requests"

B.1.2.1 Procédures normales

Dans cet état, on prévoit que des opérations subséquentes doivent être envoyées à l'entité SDF avec l'opération Bind (dans le même message). Les séquences d'événements types suivantes sont examinées dans cet état:

- quand une primitive d'invocation d'opération est reçue du processus d'application, une primitive de demande de gestion de composantes TC-INVOKE devra être envoyée vers le TC et l'instance de processus retournera vers l'état "Wait_for_Subsequent_Requests". Cette opération devra être envoyée vers l'entité SDF dans un message contenant un argument Bind. L'instance de processus SCSM attendra une réponse de séparateur de primitives à partir du processus d'application. Les opérations devant être transmises par la primitive de demande TC-INVOKE peuvent être search, modifyEntry, addEntry ou removeEntry;
- quand un séparateur de primitives est reçu à partir du processus d'application, l'instance de processus SCSM arrêtera la temporisation T_1 , déclenchera la temporisation T_2 et enverra vers le TC une primitive de demande de gestion de dialogue TC-BEGIN. Une fois que la primitive de demande TC-BEGIN sera reçue par le TC, un message contenant l'argument Bind et d'autres arguments éventuels d'opération sera le cas échéant envoyé vers l'entité SDF. Cet événement provoquera la transition de l'instance de processus SCSM vers l'état 3 "Wait_for_Bind_result". L'instance de processus SCSM attendra la réponse en provenance de l'entité SDF. La temporisation T_2 sera démarrée pour superviser l'état "Wait_for_Bind_result" et garantira que le processus SCSM est détruit si la communication avec le processus d'application SDF homologue est interrompue.



T1169700-94/d96

FIGURE B.1/Q.1218

Modèle SCF-SDF**B.1.2.2 Procédures exceptionnelles**

- Si la temporisation T_1 expire et si l'instance de processus SCSM est dans l'état "Wait_for_Subsequent_Requests", la primitive "Dialogue terminated" sera envoyée à la coordination des composants du TC afin d'éliminer l'éventuelle opération d'appel d'annuaire mémorisée et d'attendre la transmission, une primitive "Process_error" devant être envoyée au processus d'application SCF et il devra être mis fin au processus SCSM.
- Si le processus d'application SCF veut interrompre la communication avec l'entité SDF homologue, il enverra une primitive "AP_U_Abort" vers l'instance de processus SCSM, cette instance de processus enverra une primitive "Dialogue terminated" vers la Coordination des Composants du TC afin de détruire les opérations d'appel d'annuaire éventuellement mémorisées et d'attendre la transmission, et il sera mis fin au processus SCSM.

B.1.3 Etat 3 – "Wait_for_Bind_result"

B.1.3.1 Procédures normales

Dans cet état, l'entité SCF attend la réponse en provenance de l'entité SDF. La réception de la réponse dans la primitive d'indication TC-CONTINUE du TC vers l'opération Bind précédemment présentée à l'entité SDF, arrêtera la temporisation T_2 , déclenchera l'envoi d'une primitive Bind.res vers le processus d'application et provoquera une transition de l'entité SCF vers l'état "Rattachement SDF".

B.1.3.2 Procédures exceptionnelles

- Quand la temporisation T_2 expire alors que l'instance de processus SCSM est dans l'état "Wait_for_Bind_result", la primitive de demande "TC-U-ABORT" doit être envoyée vers le gestionnaire de dialogue du TC afin d'interrompre le dialogue déclenché avec le processus de l'entité SDF homologue, une primitive "Process-error" étant envoyée vers le processus d'application SCF et le processus SCSM étant interrompu.
- Si le processus d'application SCF veut interrompre la communication avec l'entité SDF homologue, il enverra une primitive "AP_U_Abort" vers l'instance de processus SCSM, cette instance de processus enverra une primitive de demande "TC-U-ABORT" vers le gestionnaire de dialogue du TC afin d'interrompre le dialogue déclenché, avec le processus de l'entité SDF homologue et il sera mis fin au processus SCSM.
- Quand une primitive d'indication TC-U-ABORT contenant une composante d'erreur d'opération directoryBind est reçue du TC à partir du processus SDF homologue, cette opération devra être transmise dans la primitive "Bind.err" vers le processus d'application et il sera mis fin au processus SCSM.
- Quand une primitive d'indication TC-U-ABORT ne contenant pas de composante d'erreur d'opération directoryBind est reçue du TC à partir du processus de l'entité SDF homologue, une primitive "U_Abort" sera transmise vers le processus d'application et il sera mis fin au processus SCSM.

B.1.4 Etat 4 – "Rattachement SDF"

B.1.4.1 Procédures normales

Dans cet état, l'entité SCF a établi un accès authentifié vers l'entité SDF, et est dans l'attente de demandes de logique de traitement de service vers l'entité SDF, ou attend des réponses aux opérations précédemment présentées à l'entité SDF:

- quand une primitive d'invocation d'opération est reçue à partir du processus d'application, une primitive de demande de gestion de composantes TC-INVOKE sera envoyée en direction du TC et l'instance de processus retournera vers l'état "SDF_Bound". L'instance du processus SCSM attendra un séparateur de primitives de réponse à partir du processus d'application. Les opérations devant être transmises par la primitive de demande TC-INVOKE peuvent être search, modifyEntry, addEntry ou removeEntry;
- quand un séparateur de primitives est reçu à partir du processus d'application, une primitive de demande TC-CONTINUE sera envoyée vers le TC afin de transmettre l'opération ou les opérations reçue(s) à l'instance éloignée de processus d'application de l'entité SDF;
- quand une primitive d'indication TC-RESULT, déclenchée par la réception adéquate d'une réponse à l'opération précédemment présentée à l'entité SDF, est reçue du TC, l'instance de processus SCSM enverra le message d'erreur d'opération au processus d'application et reviendra vers le même état;
- quand une primitive d'indication TC-U-ERROR, indiquant que l'opération sollicitée du processus d'application SCF a échoué du côté SDF homologue, est reçue du TC, le processus SCSM enverra la réponse d'erreur d'opération vers le processus d'application et reviendra vers le même état;
- quand la primitive d'indication TC-L-REJECT, indiquant que la sous-couche composant du TC a reçu une opération invalide, est reçue, l'instance de processus SCSM enverra un rejet d'opération au processus d'application et reviendra vers le même état;

- quand la primitive d'indication TC-R-REJECT, indiquant que l'opération a été rejetée par la sous-couche composant distante, est reçue, l'instance de processus SCSM enverra un rejet d'opération au processus d'application et reviendra vers le même état;
- quand la primitive d'indication TC-U-REJECT, indiquant que l'opération a été rejetée par l'instance distante de processus d'application, est reçue, l'instance de processus SCSM enverra un rejet d'opération au processus d'application et reviendra vers le même état;
- quand la primitive d'indication TC-L-CANCEL, indiquant que la demande d'opération a été localement interrompue en raison d'un dépassement de temps, est reçue, l'instance de processus SCSM informera le processus d'application et reviendra vers le même état;
- quand une primitive d'invocation de détachement Unbind.inv, causée par la nécessité d'interrompre l'accès authentifié vers l'entité SDF, est reçue du processus d'application SCF, une primitive de demande TC-END devra être envoyée vers le TC, afin de mettre fin au dialogue avec le processus d'application SDF distant et l'instance de processus SCSM sera interrompue.

B.1.4.2 Procédures exceptionnelles

- Si le processus d'application SCF veut interrompre la communication avec l'entité SDF homologue, il enverra une primitive "AP_U_Abort" vers l'instance de processus SCSM, cette instance de processus enverra une primitive de demande "TC-U-ABORT" vers le gestionnaire de dialogue du TC pour interrompre le dialogue commencé avec le processus de l'entité SDF homologue, et le processus SCSM sera interrompu.
- Si une primitive d'indication "TC-U-ABORT" est reçue du TC, une primitive "U_Abort" est transmise au processus d'application et le processus SCSM sera interrompu.

B.2 Description du processus SDSM

Le service d'extrait d'annuaire SDF est défini en termes de nombre d'opérations que peut exécuter l'entité SDF à la demande de l'entité SCF au nom de l'utilisateur. Avant d'appeler ces opérations, l'entité SCF devra effectuer une opération bind (lien) à un point d'accès. L'entité SCF pourra ensuite solliciter des opérations, chaque sollicitation identifiant l'opération particulière à exécuter et transmettant les arguments définissant la demande réelle.

B.2.1 Etat 1 – "Repos"

B.2.1.1 Procédures normales

Quand le service de gestion de dialogue du TC envoie une primitive d'indication TC-BEGIN contenant l'opération DirectoryBind, une instance de processus SDSM sera créée, une primitive "Bind.inv" sera envoyée au processus d'application et la temporisation T₃ sera démarrée. Cet événement entraîne la transition vers l'état "Rattachement en cours". La temporisation T₃ est démarrée afin de superviser l'état "Rattachement en cours" et garantira que le processus SDSM est détruit quand la communication avec le processus d'application est interrompue.

B.2.1.2 Procédures exceptionnelles

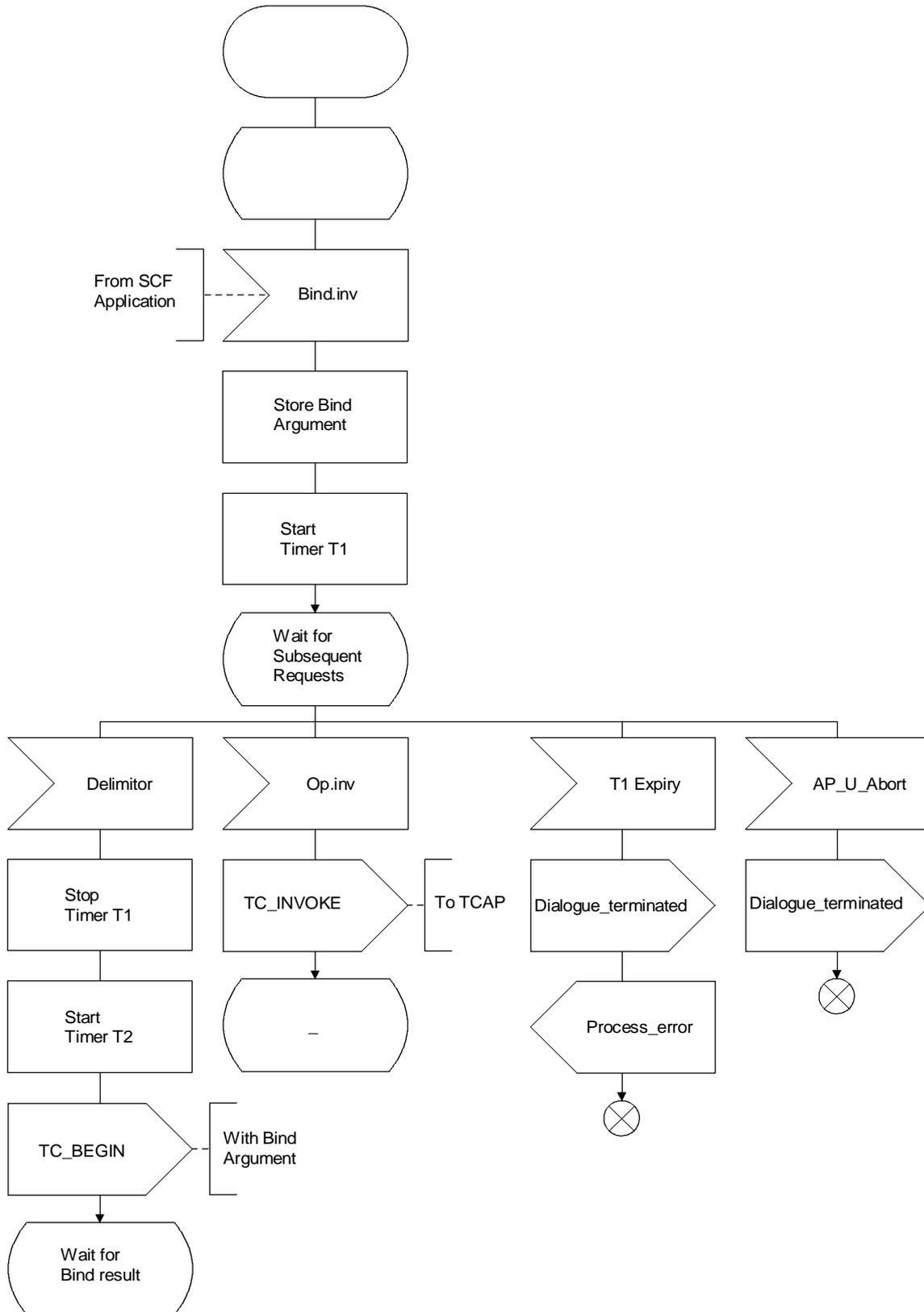
Aucune identifiée.

B.2.2 Etat 2 – "Rattachement en cours"

B.2.2.1 Procédures normales

Dans cet état, une demande de lien a été reçue de l'entité SCF. L'entité SDF accomplit les procédures de contrôle d'accès sous-jacentes à l'opération de lien (par exemple authentification d'accès). Il peut également y avoir l'éventualité qu'une opération de lien soit factice. Dans ce cas, l'authentification d'accès n'est pas nécessaire. Deux événements sont pris en compte dans cet état:

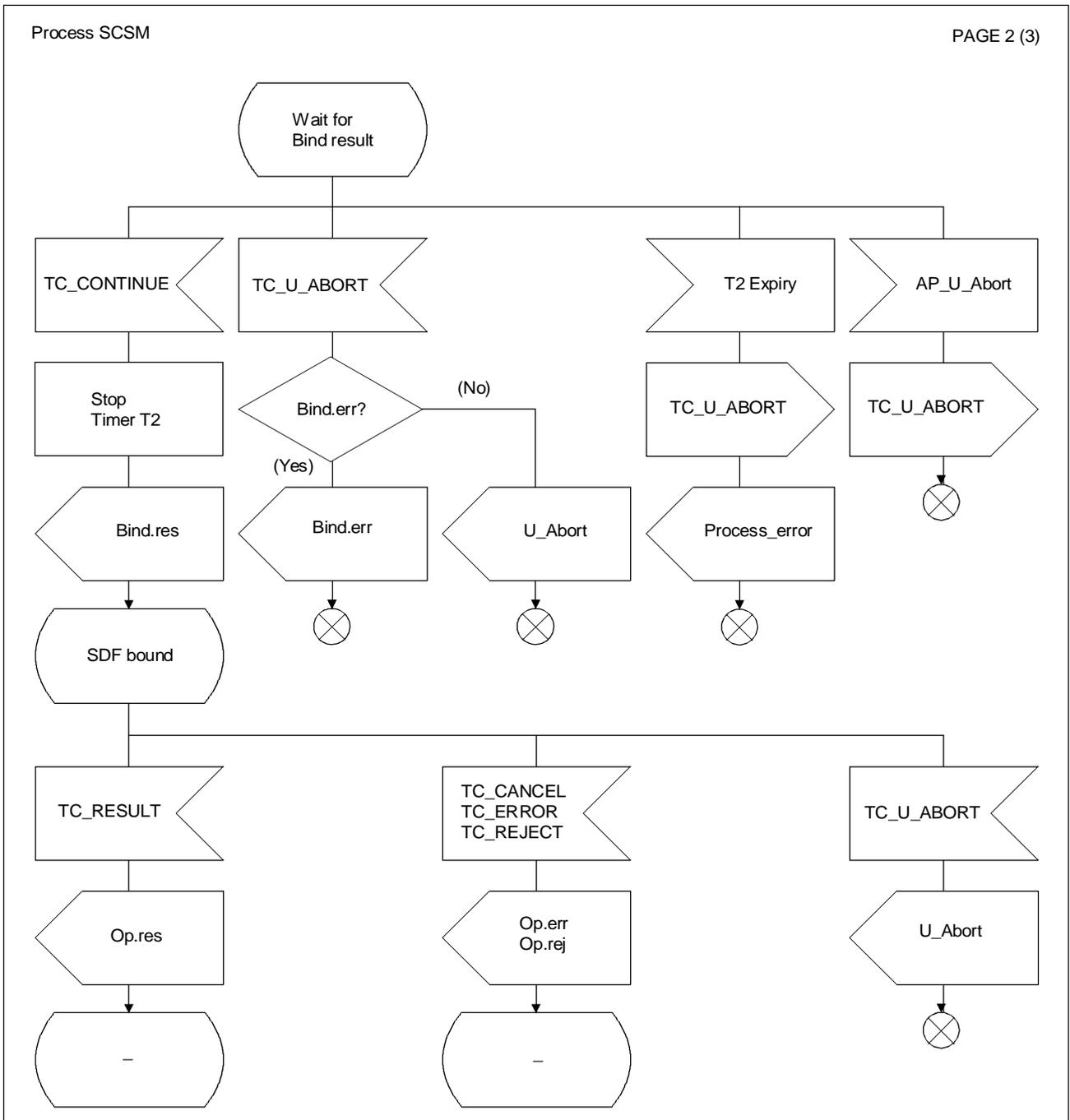
- quand la primitive de gestion de composantes, TC-INVOKE ind., causée par la réception d'opérations, est reçue avant que le résultat de l'opération de lien ne soit envoyé à partir du processus d'application, une primitive "Op.inv" sera envoyée au processus d'application et l'instance de processus SDSM reviendra vers l'état "Rattachement en cours";
- quand la demande de lien aboutit, le processus d'application SDF enverra vers l'instance de processus SDSM une primitive "Bind.res" contenant le message d'erreur du directoryBind. Sur réception de la primitive "Bind.res", l'instance de processus SDSM arrêtera la temporisation T₃, mémorisera localement l'information jusqu'à ce qu'un séparateur de primitives soit reçu du processus d'application, et transitera vers l'état "Rattachement SCF".



T1169710-94/d97

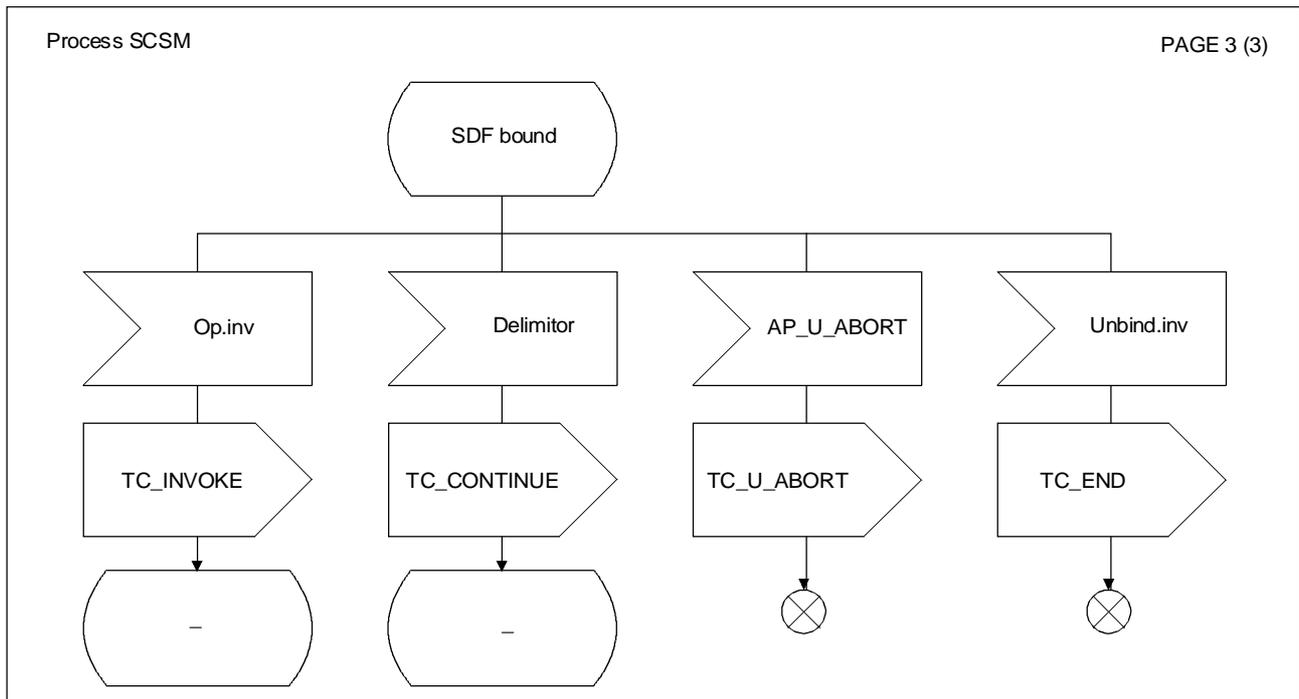
FIGURE B.2/Q.1218 (feuille 1 de 3)

SCSM



T1169720-94/d98

FIGURE B.2/Q.1218 (feuillet 2 de 3)
SCSM



T1169730-94/d99

FIGURE B.2/Q.1218 (feuillet 3 de 3)

SCSM

B.2.2.2 Procédures exceptionnelles

- Si la demande de lien échoue, le processus d'application enverra vers l'instance de processus SDSM une primitive "Bind.err" contenant les erreurs du directoryBind. L'instance de processus SDSM enverra au gestionnaire de dialogue de TC une primitive de demande "TC-U-ABORT", afin d'interrompre le dialogue engagé avec le processus de l'entité SCF homologue, et le processus SDSM sera interrompu.
- Si la temporisation T_3 expire, et si l'instance de processus SDSM est dans l'état "Bind_pending", la primitive de demande "TC-U-ABORT" sera envoyée au gestionnaire de dialogue de TC afin d'interrompre le dialogue engagé avec le processus de l'entité SCF homologue, une primitive "Process_error" étant envoyée au processus d'application SDF et le processus SDSM étant interrompu.
- Si le processus d'application SDF veut interrompre la communication avec l'entité SCF homologue, il enverra une primitive "AP_U_Abort" vers l'instance de processus SDSM, cette instance de processus enverra au gestionnaire de dialogue de TC une primitive de demande "TC-U-ABORT", afin d'interrompre le dialogue engagé avec le processus de l'entité SCF homologue, et le processus SDSM devra être interrompu.

B.2.3 Etat 3 – "Rattachement SCF"

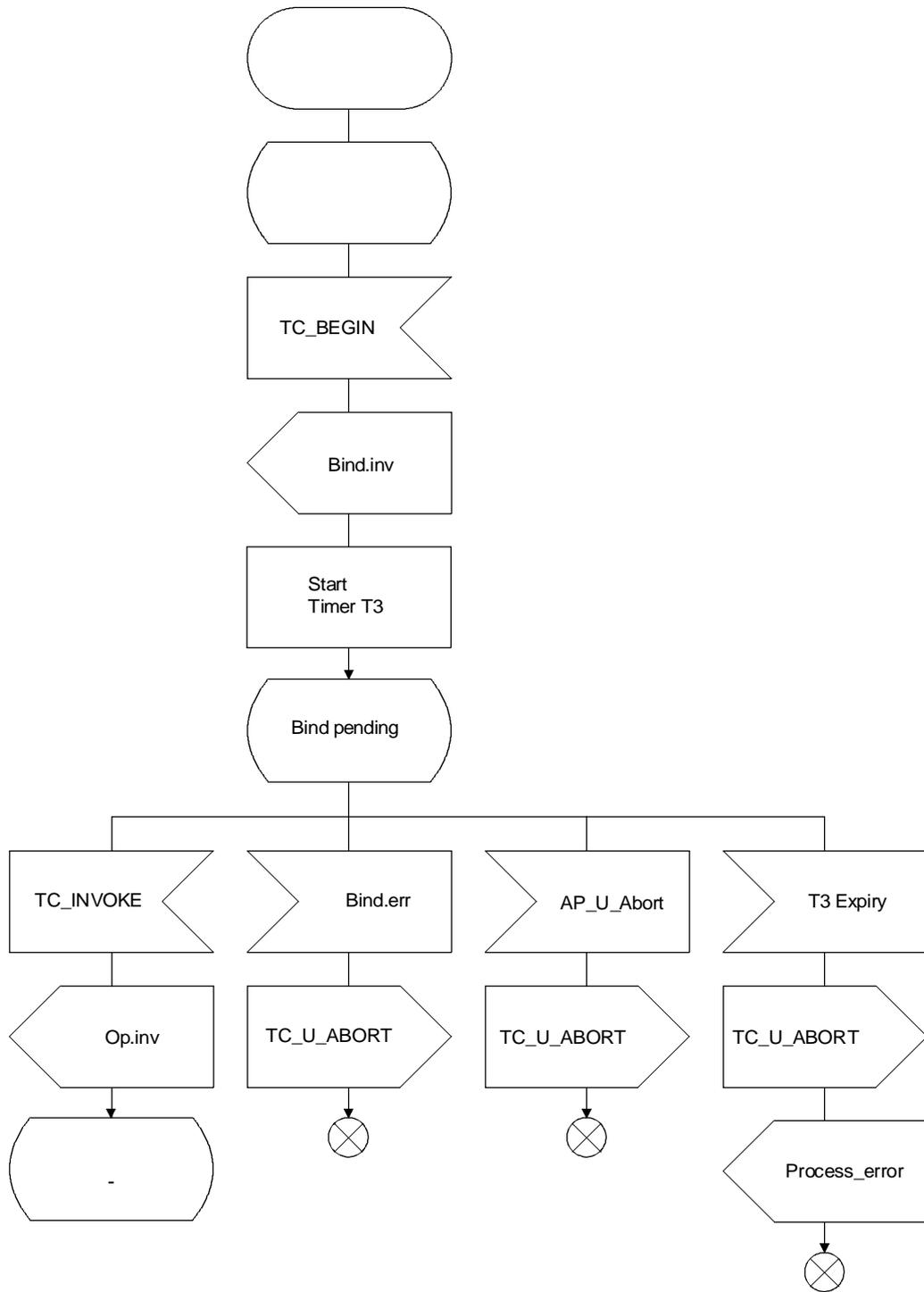
B.2.3.1 Procédures normales

Dans cet état, l'accès de l'entité SCF vers l'entité SDF est autorisé, et les opérations provenant de l'entité SCF sont acceptées. En plus d'attendre des demandes de l'entité SCF, l'entité SDF peut également envoyer dans cet état des réponses à des opérations précédemment demandées. Les séquences d'événements types suivantes sont examinées dans cet état:

- si une primitive Op.res, indiquant qu'une opération est exécutée avec succès, est reçue du processus d'application, une primitive de demande de gestion de composantes, TC-RESULT, sera envoyée vers le TC et l'instance de processus reviendra vers l'état "Rattachement SCF". L'instance de processus SDSM attendra une réponse du séparateur de primitives afin d'envoyer une primitive de demande TC-CONTINUE vers le gestionnaire de dialogue du TC afin que le TC puisse acheminer ce message d'erreur vers l'instance du processus SCSM homologue. Les messages d'erreur devant être renvoyés par la primitive de demande TC-RESULT peuvent être la réponse aux opérations search, modifyEntry, addEntry ou removeEntry précédemment reçues.
- si une primitive Op.err, indiquant qu'une opération précédemment reçue ne peut pas être exécutée avec succès, est reçue du processus d'application, une primitive TC-U-ERROR demande de gestion de composantes devra être envoyée vers le TC et l'instance de processus reviendra vers l'état "Rattachement SDF". L'instance de processus SDSM attendra un séparateur de primitives en réponse au processus d'application pour envoyer une primitive de demande TC-CONTINUE au gestionnaire de dialogue du TC, afin que le TC puisse transmettre ce message d'erreur vers l'instance de processus SCSM homologue. Les messages d'erreur devant être envoyés par la primitive de demande TC-U-ERROR peuvent être la réponse aux opérations search, modifyEntry, addEntry ou removeEntry.
- si un séparateur de primitives est reçu du processus d'application, l'instance de processus SDSM enverra au TC une primitive de demande de gestion de dialogue TC-CONTINUE. Une fois que la primitive de demande TC-CONTINUE sera reçue par le TC, un message contenant le message d'erreur Bind, le message d'erreur ou les arguments de rejet précédemment mémorisés, s'il n'a pas déjà été envoyé, sera envoyé vers l'entité SCF, ainsi que les éventuels résultats d'opération, message d'erreurs ou arguments de rejet. Cet événement provoquera le retour de l'instance de processus SDSM vers l'état "Rattachement SCF".
- si une primitive d'indication TC-END résultant de la réception de l'opération "Détachement de l'entité SCF", est reçue du gestionnaire de dialogue du TC, l'instance de processus SDSM enverra une primitive "Unbind.inv" au processus d'application et interrompra l'instance de processus SDSM. L'association SCF-SDF sera interrompue, et toutes les ressources associées seront libérées.
- si une primitive d'indication TC-INVOKE de gestion de composantes est reçue à partir du TC, une primitive "Op.inv" devra être envoyée au processus d'application et l'instance de processus SDSM devra revenir vers l'état "Rattachement SCF".

B.2.3.2 Procédures exceptionnelles

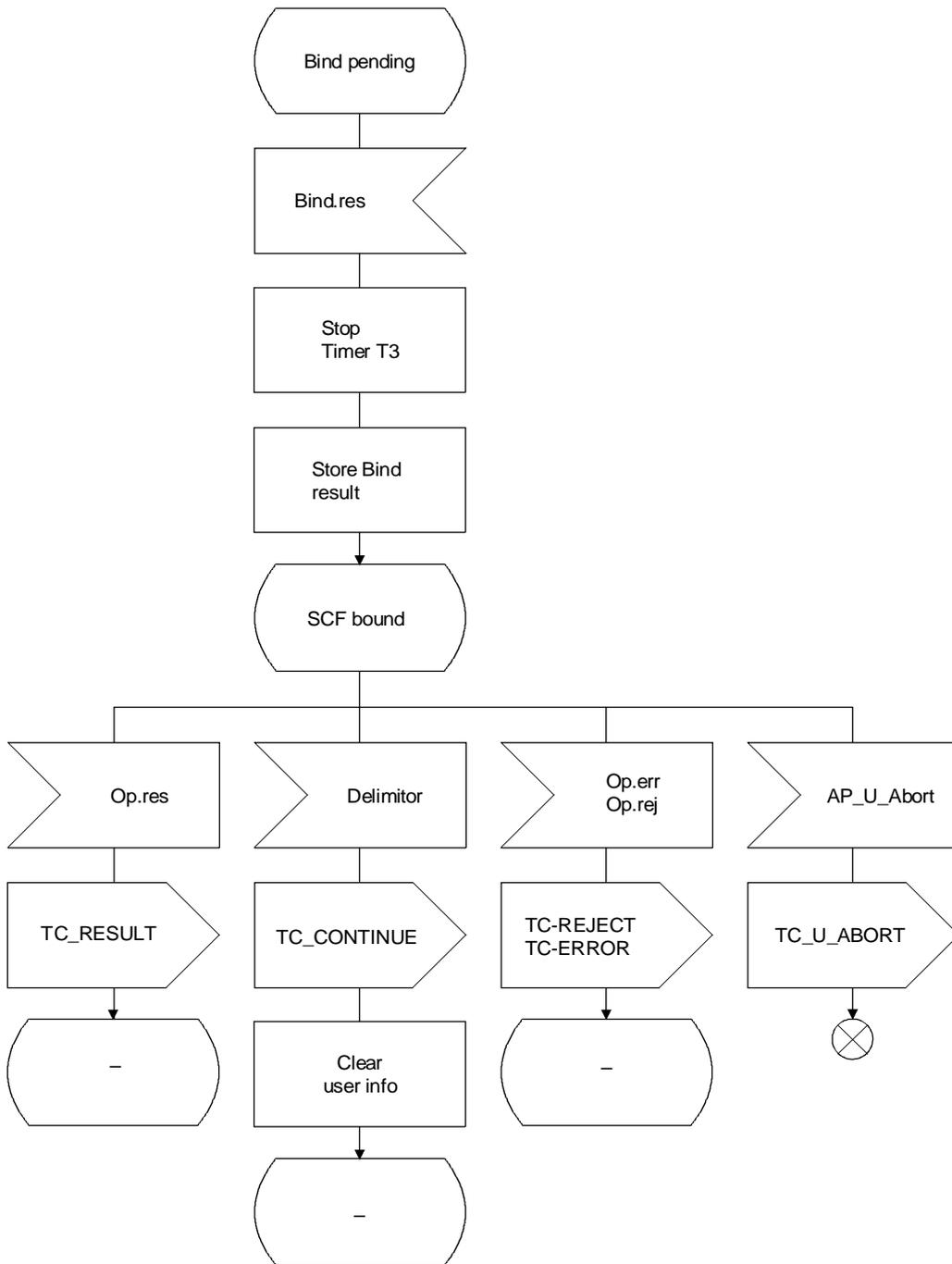
- Si une primitive Op.rej, indiquant qu'une opération précédemment reçue est rejetée, est reçue à partir du processus d'application, une primitive de demande TC-U-REJECT de gestion de composantes sera envoyée vers le TC et l'instance de processus reviendra vers l'état "Rattachement SDF". L'instance de processus SDSM attend en réponse du processus d'application un séparateur de primitives pour envoyer une primitive de demande TC-U-CONTINUE vers le gestionnaire de dialogue du TC, afin que celui-ci puisse transmettre ce rejet vers l'instance de processus SCSM homologue. Le rejet devant être transmis par la primitive de demande TC-U-REJECT peut être la réponse aux opérations search, modifyEntry, addEntry ou removeEntry.
- Si une primitive d'indication TC-U-ABORT est reçue du TC, une primitive "U_Abort" sera transmise au processus d'application et le processus SDSM sera interrompu.
- Si le processus d'application SDF veut interrompre la communication avec l'entité SCF homologue, il enverra une primitive "AP_U_Abort" vers l'instance de processus SDSM; ce processus enverra une primitive de demande "TC-U-ABORT" au gestionnaire de dialogue du TC afin d'interrompre le dialogue avec le processus de l'entité SCF homologue, et le processus SDSM sera interrompu.



T1169740-94/d100

FIGURE B.3/Q.1218 (feuillet 1 de 3)

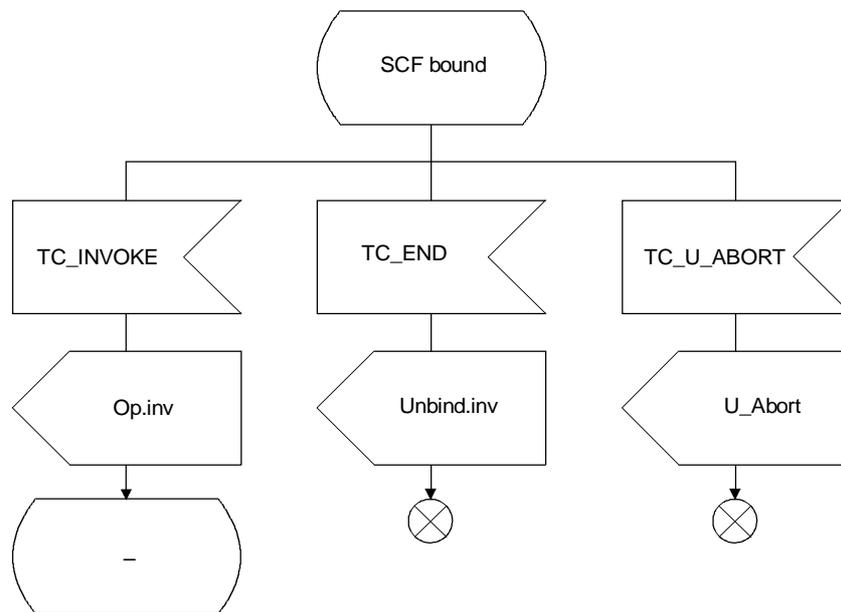
SDSM



T1169750-94/d101

FIGURE B.3/Q.1218 (feuille 2 de 3)

SDSM



T1169760-94/d102

FIGURE B.3/Q.1218 (feuillet 3 de 3)

SDSM**Appendice I****Modélisation des données de service**

(Cet appendice ne fait pas partie intégrante de la présente Recommandation)

Le présent appendice décrit les étapes requises et offre un ensemble de directives et d'exemples permettant de définir un modèle de données de service ainsi que son application sur le modèle d'information défini au 2.2.2.3. Ces directives et ces exemples constituent un mécanisme simple qui traite de ces points aux I.1.1 et I.2.1.

Des outils additionnels (par exemple des spécifications de contexte d'attribut) peuvent aussi apporter des solutions à la modélisation des données. Le présent appendice ne traite cependant pas de ces outils.

I.1 Étude du besoin de modélisation des données de service

L'adoption du modèle d'information de la Recommandation X.501 comme modèle pour la représentation des données de service dans l'entité SDF, ainsi que l'utilisation des opérations décrites dans la Recommandation X.511 pour l'interface entre l'entité SCF et l'entité SDF, ont soulevé la question de la façon dont les données de service du réseau intelligent devraient être mises en correspondance avec ce modèle d'information.

Des correspondances inappropriées des données de service avec le modèle d'information de la Recommandation X.500 utilisé dans l'entité SDF peut accroître considérablement la quantité des informations qui doivent être transférées sur l'interface SCF-SDF (dans certains cas, en excédant même la capacité du lien TCAP). Hormis le fait que le trafic de données sur l'interface sera plus important, le choix des correspondances peut aussi avoir une incidence sur la quantité de traitements devant être exécutés par les actions d'entité fonctionnelle (FEA) à l'intérieur de la Fonction commande de service (SCF), au point de rendre la réalisation des FEA de la fonction SCF dépendante des données de service.

Ce paragraphe met en évidence un ensemble unique de règles de correspondance qui peuvent être appliquées aux données de service pour simultanément minimiser le trafic sur l'interface de l'association SCF-SDF et maintenir une gestion des données indépendante du service.

I.1.1 Généralités sur la modélisation

Il y a trois critères à respecter pour l'organisation de données de service du réseau intelligent dans le modèle d'information de la Recommandation X.500. Chacun de ces critères devrait avoir la même importance.

- **Critère 1** – Minimiser le trafic entre l'entité SCF et l'entité SDF.

L'interface SCF-SDF sera probablement l'une des interfaces la plus utilisée dans l'environnement du réseau intelligent. Chaque instance SLPI de l'entité SCF peut avoir à initialiser de multiples accès aux données stockées dans l'entité SDF. Du point de vue performance, il est essentiel que les transferts de données soient réduits au minimum.

- **Critère 2** – Minimiser les traitements de données devant être effectués dans l'entité SCF.

Les traitements de données dans l'entité SCF sont effectués par les FEA définies dans la Recommandation Q.1214 pour gérer les flux d'informations. Ces FEA sont supposées être indépendantes du service pour lequel elles sont sollicitées. Si les données qui sont transférées n'ont pas la résolution correcte, le traitement dépendant du service peut être effectué par les FEA dans l'entité SCF. Des précautions doivent être prises pour la conception du modèle d'information de la Recommandation X.500 pour les données de service du réseau intelligent afin de maintenir l'indépendance du service de l'action d'entité fonctionnelle (FEA).

- **Critère 3** – Examiner l'effet de répartition des données dans la gestion des données de la Recommandation X.500.

Pour les systèmes de l'ensemble de capacités CS-1, la mise en œuvre SDF peut ne pas être répartie. Il est à noter que les systèmes prévus dans la Recommandation X.500 sont de façon inhérente des systèmes de gestion de données réparties. Si la conception de la structure de l'arbre d'informations d'Annuaire (DIT) ne tient pas compte de la répartition des données, l'Annuaire peut être impossible à gérer en termes de gestion des "connaissances" nécessaires pour accéder aux informations d'Annuaire à travers de nombreux agents de service d'annuaire (DSA).

Bien qu'on puisse arguer du fait qu'une réalisation centralisée d'annuaire puisse se faire sans tenir compte de la répartition, les exigences de performance peuvent exiger l'éclatement de l'annuaire sur plusieurs DSA locaux, rendant nécessaire que les connaissances de la répartition soient maintenues dans chaque agent DSA.

I.1.2 Modélisation du service – Définition de l'information nécessaire

La modélisation du service est définie dans la Recommandation Q.1203. Elle comprend les étapes suivantes:

- 1) identification des éléments de service constituant le service;
- 2) identification des données nécessaires à ces éléments de service;
- 3) identification des données partagées par les éléments utilisés par le service;
- 4) identification des opérations et traitements nécessaires sur les données de service;
- 5) identification des modalités de contrôle d'accès pour les données de service.

Ces étapes ne sont pas couvertes dans la présente Recommandation. Seules les étapes supplémentaires nécessaires pour faire correspondre les données de service avec le modèle d'information SDF défini au 2.2 sont traitées ici.

I.2 Conseils pour l'élaboration des modèles de données de service

Une fois que le modèle de données de service pour le réseau intelligent a été défini en termes de données pour les éléments de service, il est nécessaire de le faire correspondre avec le modèle d'information de la Recommandation X.500, selon les étapes suivantes:

- 1) sélection des ATTRIBUTS;
- 2) sélection des CLASSES D'OBJETS;
- 3) conception du service DIT;
- 4) affectation du Contrôle d'Accès aux ATTRIBUTS et CLASSES D'OBJETS.

I.2.1 Sélection des ATTRIBUTS

La sélection des ATTRIBUTS est affectée par les opérations et traitements exigés par l'entité SCF. Les directives ci-dessous sont fournies pour faciliter la sélection:

- **Directive 1** – Les ATTRIBUTS sont les plus petites unités adressables de données de service du réseau intelligent pouvant être transférées sur l'interface SCF-SDF pour traitement par l'entité SCF. Les données de type structuré doivent être évitées dans la définition des ATTRIBUTS.
- **Directive 2** – Tous les ATTRIBUTS de l'Annuaire devant être extraits par le service de réseau intelligent sont de préférence de VALEUR UNIQUE.
- **Directive 3** – Si les données de service contiennent des tableaux avec plusieurs colonnes d'informations (voir la Figure I.1), chaque colonne^{3), 4)} du tableau doit avoir une définition d'ATTRIBUT.

Par exemple, sur la Figure I.1a) (table d'acheminement horaire) trois ATTRIBUTS sont définis (Heure de début, Heure de fin et Destination). Pour la Figure I.1b) (table de numérotation abrégée) deux ATTRIBUTS sont définis (AbbrevNo et Destination) (Numéro abrégé et Destination).

Une fois tous les ATTRIBUTS identifiés, il peut être possible de réduire le nombre total d'ATTRIBUTS définis en raison des définitions identiques (par exemple Destination dans la Figure I.1).

Heure Début	Heure Fin	Destination
01:00	09:00	204 23456
09:01	15:00	204 12345
15:01	23:00	204 23456

a) Table d'acheminement horaire

Numéro Abrégé	Destination
01	204 23456
02	204 12345
03	204 23456

b) Table de numérotation abrégée

FIGURE I.1/Q.1218

Tableaux à plusieurs colonnes de données de service

³⁾ Cela suppose que les tables soient organisées comme dans la Figure I.1, chaque rangée dans la table représentant un groupe distinct d'informations. Si cette organisation est inversée, les colonnes contenant les groupes d'articles, chaque rangée dans la table devrait avoir une définition ATTRIBUT.

⁴⁾ Cette directive est normalement utilisée en liaison avec la Directive 6 pour déterminer les rubriques d'entrée dans l'arbre d'informations de l'Annuaire (DIT).

I.2.1.1 ATTRIBUT Définition de types

Un type d'ATTRIBUT est défini par une définition de données ASN.1.

```
thisAttribute ATTRIBUTE ::= {  
    ..  
    WITH SYNTAX thisAttributeASN1DataType,  
    ..  
    }
```

Un ATTRIBUT peut être défini comme un SOUS-TYPE d'un autre ATTRIBUT, ce qui signifie qu'il doit se conformer à la définition de l'ATTRIBUT de la même famille.

```
thisAttribute ATTRIBUTE ::= {  
    ..  
    SUBTYPE OF anotherAttribute,  
    ..  
    }
```

Alors que ceci fournit un moyen commode de réutiliser les définitions d'attributs, cela peut avoir un effet secondaire sur le Critère 1. Une opération de Recherche dans la Recommandation X.500 pour un type d'attribut spécifique produira non seulement les valeurs d'attribut souhaitées, mais également les valeurs d'attribut pour les attributs qui sont un sous-type de l'attribut demandé.

Il convient de noter qu'une opération de Recherche pour un attribut particulier ne renverra pas l'une quelconque des valeurs des attributs de la même famille.

Un ATTRIBUT peut-être défini comme un attribut collectif, ce qui implique que ses valeurs peuvent être réparties entre un certain nombre d'entrées d'Annuaire.

```
collectiveAttribute ATTRIBUTE ::= {  
    ..  
    COLLECTIVE TRUE,  
    ..  
    }
```

Des exemples d'attributs prédéfinis peuvent être trouvés à l' article 2/X.520.

I.2.1.2 Définition des valeurs autorisées

Deux méthodes peuvent être utilisées pour définir les valeurs autorisées d'un attribut. Ce sont les suivantes:

- 1) limiter les valeurs à un ensemble de valeurs spécifiques;
- 2) limiter les valeurs à une gamme de valeurs numériques.

La première méthode exige deux attributs supplémentaires:

- a) un attribut contenant les valeurs autorisées pour l'attribut en cause;
- b) un attribut de contrôle d'accès associant ces deux attributs. Cet attribut doit avoir une syntaxe de type ACIItem et doit contenir un champ d'item protégé (ProtectedItem) défini comme suit:

```
ProtectedItem ::= SEQUENCE {  
    ..  
    restrictedBy [10] SET OF RestrictedValue OPTIONAL,  
    ..  
    }
```

L'élément RestrictedValue contient les identificateurs de l'attribut courant et de l'attribut contenant les valeurs permises. Comme l'élément RestrictedValue fait partie de l'attribut de commande d'accès, la possibilité d'en modifier la valeur implique la possibilité de modifier tous les paramètres de commande d'accès de l'entrée courante figurant dans l'attribut de commande d'accès.

La deuxième méthode n'exige qu'un seul attribut supplémentaire:

- c) un attribut de contrôle d'accès contenant la gamme de valeurs pour l'attribut en cause. Cet attribut doit avoir une syntaxe de type ACIITEM et doit contenir un champ d'item protégé (ProtectedItem) défini comme suit:

```
ProtectedItem ::= SEQUENCE {  
    ..  
    rangeOfValues [7] Filter OPTIONAL,  
    ..  
    }
```

L'élément Filter contient la spécification de gamme de valeurs pour l'attribut. On notera que, pour modifier une seule partie de la gamme de valeurs, il faut remplacer l'ensemble de l'élément Filter. D'autre part, comme la valeur Filter fait partie de l'attribut de contrôle d'accès, la capacité de modifier la valeur Filter implique celle de modifier tous les paramètres de contrôle d'accès pour l'entrée actuellement contenue dans l'attribut de contrôle d'accès.

I.2.1.3 Définition des valeurs par défaut

La valeur par défaut d'un attribut est définie au moment de l'exécution par la spécification d'une valeur d'attribut avec un contexte dont le champ Fallback est mis à la valeur TRUE.

Les valeurs d'attribut dont les valeurs de repli de contexte sont mises à TRUE ne seront retournées que si aucune autre valeur d'attribut répondant aux critères n'est présente.

I.2.1.4 Définition des valeurs MAX des attributs

La limitation du nombre de valeurs qu'un attribut peut avoir est effectuée par spécification d'un attribut de contrôle d'accès contenant un champ ProtectedItem ayant la syntaxe suivante:

```
ProtectedItem ::= SEQUENCE {  
    ..  
    maxValueCount    [8] SET OF MaxValueCount OPTIONAL,  
    ..  
}
```

L'élément MaxValueCount contient l'identificateur d'attribut et le nombre maximal de valeurs pouvant être conservées dans l'attribut. Etant donné que la valeur MaxValueCount fait partie de l'attribut de contrôle d'accès, la capacité de modifier la valeur MaxValueCount implique celle de modifier tous les paramètres de contrôle d'accès pour l'entrée actuellement contenue dans l'attribut de contrôle d'accès.

I.2.1.5 Définition des durées de vie des attributs

On définit les durées de vie des attributs en fournissant un contexte temporel (TemporalContext) pour la valeur d'attribut (au moment de l'exécution).

I.2.1.6 Définition des règles de CORRESPONDANCE

Les règles de CORRESPONDANCE définissent comment les valeurs d'attributs sont comparées à d'autres valeurs extérieures pour être utilisées dans les Filtres de recherche et les correspondances de noms. Une définition d'ATTRIBUT a trois types différents de règles de correspondance.

EQUALITY (ÉGALITÉ) pour le test (valeur == valeur d'attribut).
ORDERING (SÉQUENCEMENT) pour le test (valeur < valeur d'attribut).
SUBSTRINGS (SOUS-CHAÎNES) pour le test de concordance des sous-chaînes.

```
thisAttribute ATTRIBUTE ::= {  
    ..  
    EQUALITY MATCHING RULE equalityMatchingRuleName,  
    ORDERING MATCHING RULE orderingMatchingRuleName,  
    SUBSTRINGS MATCHING RULE substringsMatchingRuleName,  
    ..  
}
```

Des exemples de règles de correspondance prédéfinies peuvent être trouvés dans l'article 3/X.520.

I.2.1.7 Attribution des identificateurs d'objets attributs

Un identificateur d'objet unique est attribué à chaque ATTRIBUT défini. La forme de cet identificateur d'objet devrait être la suivante:

-- *IN object identifier prefix*

in-oi OBJECT IDENTIFIER ::= { ccitt recommendation q 1218 ... }

-- *service attribute definitions*

attributeX OBJECT IDENTIFIER ::= { in-oi ServiceID attributeType(4) AttributeID }

-- *ServiceID defines the service/service feature to which the attribute belongs.*

-- *AttributeID defines the Attribute ID within the service: this is assigned by the Service*

-- *designer.*

I.2.2 Sélection des CLASSES D'OBJET

Une fois que les ATTRIBUTS utilisés dans le modèle de données de service ont été identifiés, ils peuvent être regroupés dans des éléments spécifiques de CLASSE D'OBJET.

I.2.2.1 Définition de la CLASSE D'OBJET

Une CLASSE D'OBJET définit la spécification d'une entrée d'Annuaire. Elle comprend un groupe d'ATTRIBUTS réunis dans une classe d'objets unique. Chaque instance d'une entrée d'Annuaire dans l'arbre DIT possède une identité unique (voir les spécifications en 6.2.2.2). Les directives suivantes peuvent être utilisées dans la définition des CLASSES D'OBJETS.

- **Directive 4** – Les valeurs des données de service pouvant partager une même identité unique commune doivent être groupées dans une entrée d'Annuaire comme ATTRIBUTS (par exemple données de service, données d'utilisateur pour un service).
- **Directive 5** – Les valeurs des données de service faisant partie d'un tableau à plusieurs colonnes (voir Directive 3) doivent être groupées dans une définition unique de CLASSE D'OBJET.

Une CLASSE D'OBJET peut être définie comme SOUS-CLASSE d'une autre CLASSE D'OBJET. Dans ce cas, la CLASSE D'OBJET contient tous les attributs et caractéristiques de la CLASSE D'OBJET de la famille.

Cela est identique à l'héritage en langage C++.

I.2.2.2 Sélection du nom RDN (NAME-FORM)

Chaque CLASSE D'OBJET peut avoir plusieurs attributs désignés comme attributs RDN potentiels. Pour cela, on utilise la spécification NAME-FORM. A la création de l'entrée d'Annuaire, l'un d'eux doit être désigné comme RDN actif (par le "Nom distingué" défini dans l'opération AddEntry). Une fois que l'objet a été créé, seul le RDN actif peut être utilisé dans les procédures de correspondance pour la localisation d'un objet avec les opérations d'accès (Search, ModifyEntry, RemoveEntry, ...).

Si deux objets de la même CLASSE D'OBJETS sont créés avec les mêmes valeurs d'ATTRIBUT mais avec des RDN actifs différents, ils seront considérés comme étant des objets distincts dans le DIT. Si le même nom RDN avait été utilisé pour le deuxième objet, une erreur de création d'objet aurait été signalée.

I.2.2.3 Attribution des identifiants d'objets aux CLASSES D'OBJET

Un identificateur d'objet est attribué à chaque CLASSE D'OBJET définie. La structure de cet identificateur d'objet doit être la suivante:

-- *IN object identifier prefix*

in-oi OBJECT IDENTIFIER ::= { ccitt recommendation q 1218 ... }

-- *service object-class definitions*

objectClassX OBJECT IDENTIFIER ::= { in-oi ServiceID objectClass(6) ObjectClassID }

-- *ServiceID defines the service/service feature to which the attribute belongs.*

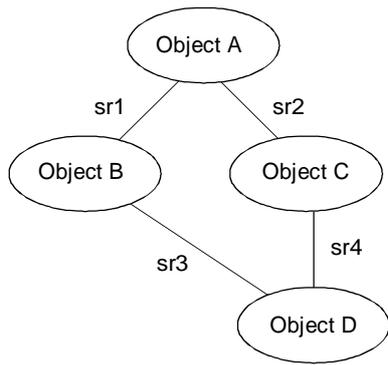
-- *ObjectClassID defines the Object Class ID within the service: this is assigned by the Service designer.*

I.2.3 Définition de l'arbre DIT

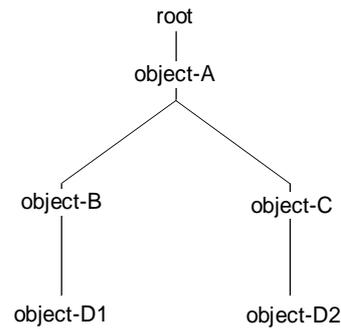
Ce paragraphe décrit la façon d'élaborer l'arbre DIT pour un modèle de données de service du réseau intelligent.

I.2.3.1 Définition de la hiérarchie des objets (RÈGLES DE STRUCTURE)

L'arbre DIT définit une hiérarchie pour les objets données. Chaque "Nom distingué" d'un objet comprend une concaténation du nom RDN de l'objet avec les RDN des objets supérieurs à celui-ci dans l'arbre DIT. La hiérarchie des objets dans l'arbre DIT est définie avec les RÈGLES DE STRUCTURE. La Figure I.2 explique la formation de l'arbre DIT par un ensemble des CLASSES D'OBJET et DES RÈGLES DE STRUCTURE.



a) Conception des règles structurelles



T1169770-94/d103

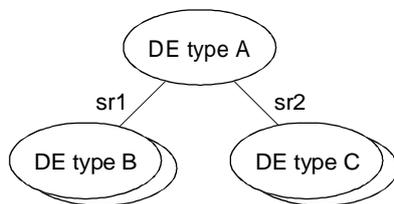
b) Structure de l'arbre DIT

FIGURE I.2/Q.1218

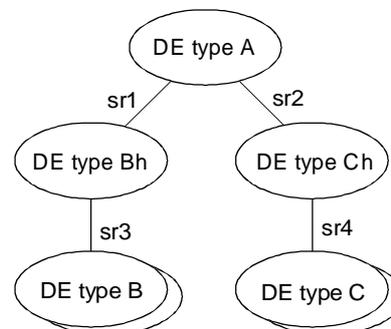
Correspondance entre LES RÈGLES DE STRUCTURE et l'arbre DIT

L'ensemble des règles qui suivent peuvent être utilisées pour construire l'arbre DIT à partir d'un ensemble de spécifications CLASSE D'OBJETS.

- **Directive 6** – Si les valeurs de données de service partagent une même structure commune mais peuvent avoir des ensembles de valeurs multiples (par exemple, rangées d'une table de traduction horaire, données spécifiques à l'utilisateur dans un service), celles-ci doivent être placées dans des entrées de l'Annuaire qui sont subordonnées à une autre entrée de l'Annuaire (c'est-à-dire qu'il existe une RÈGLE DE STRUCTURE définissant la relation entre les entrées d'Annuaire).
- **Directive 7** – Si des ensembles multiples d'entrées de données existent sous une entrée de données spécifique, il est recommandé que chacun de ces ensembles soit subordonné à une entrée de données unique, elle-même subordonnée à une entrée de données spécifique (voir la Figure I.3). Cela permettra une recherche plus efficace de l'ensemble des entrées de données en utilisant les opérations contenues dans la Recommandation X.500.



a) Après conception



T1169780-94/d104

b) Mise en œuvre recommandée

FIGURE I.3/Q.1218

Structure du modèle recommandé pour des ensembles multiples

I.2.3.2 Définition de chemins alternatifs de nommage (Alias)

Quand une classe d'objet a été "instanciée" son "Nom distingué" est fixé. Si un chemin alternatif de nommage est requis pour l'objet, un objet alias (mnémonique) doit être défini dans l'arbre DIT. Cet objet alias aura ses propres attributs RDN et comportera un pointeur vers l'objet référencé.

Un objet alias est une CLASSE D'OBJET qui est un sous-type de la CLASSE D'OBJETS alias prédéfinie.

I.2.3.3 Définition du nombre maximal d'entrées subordonnées

On peut limiter le nombre d'entrées subordonnées qui peuvent dépendre d'une entrée en spécifiant un attribut de contrôle d'accès défendant de cette entrée et contenant un champ d'item protégé, défini comme suit:

```
ProtectedItem ::= SEQUENCE {  
    ..  
    maxImmSub      [8]  INTEGER OPTIONAL,  
    ..  
}
```

La valeur maxImmSub contient le nombre maximal d'entrées subordonnées qui peuvent être conservées au-dessous de l'entrée actuelle. On notera que cette limite ne s'applique qu'à l'agent DSA actuel. Etant donné que la valeur maxImmSub fait partie de l'attribut de contrôle d'accès, la capacité de changer la valeur maxImmSub implique celle de modifier tous les paramètres de contrôle d'accès pour l'entrée contenue dans l'attribut Contrôle d'accès.

I.3 Exemple de service (similaire à UPT)

Ce paragraphe réalise un arbre DIT X.500 pour un sous-ensemble simplifié d'un service UPT utilisant les règles de correspondance.

I.3.1 Description d'exemple de service

L'exemple de service contient les éléments de service suivants:

- 1) appel entrant;
- 2) inscription de destinataire;
- 3) filtrage d'appel entrant;
- 4) appel sortant;
- 5) filtrage d'appel sortant par destinataire.

Ces éléments sont décrits brièvement ci-dessous, ainsi que les données qui seraient disponibles après l'étape de la conception du service (voir I.1.2). L'information pour chaque élément de service est présentée sous forme de tableau avec les colonnes suivantes:

- | | |
|---------------------------------------|--|
| – Données: | le nom d'objet de données. |
| – Définition: | définition du type d'objet de données (notation de type ASN.1 utilisée). |
| – Cardinalité: | celle-ci définit le nombre de versions de l'élément pouvant arriver (ex. une par utilisateur du service). |
| – Opérations des éléments de service: | celle-ci définit les opérations (interface SCF-SDF) utilisées pour manipuler l'objet de données. |
| – Contrôle d'accès: | celle-ci définit le niveau d'accès (lecture, écriture) nécessaire pour les différents niveaux d'opérateurs du service. |

NOTE – Ceci ne prétend pas être un service complet. Seuls les détails nécessaires sont fournis pour montrer comment les mises en correspondance avec le modèle d'informations SDF doivent être réalisées.

I.3.1.1 Élément de service – Appel entrant

L'opération normale de la fonction "Appel entrant" exécute l'algorithme d'acheminement suivant:

- 1) si l'utilisateur a une localisation enregistrée active, alors envoyer l'appel vers cette localisation;
- 2) si la fonction localisation horaire est activée, alors acheminer vers la localisation correspondante;
- 3) si le destinataire a une localisation par défaut, alors acheminement vers la localisation par défaut;
- 4) l'appel doit être rejeté avec l'annonce "utilisateur non disponible sur le système";

Toutes les localisations définies doivent faire partie d'un ensemble de localisations prédéfinies pour l'utilisateur.

Informations relatives aux éléments de service

Données	Définition	Cardinalité	Opérations des éléments de service	Contrôle d'accès
Localisations enregistrées	SET OF Location (ENSEMBLE de localisations)	une par utilisateur du service	néant	service: L gestion utilisateur: L/E utilisateur: L
Localisation horaire	SEQUENCE { startTime SEQUENCE { day DayOfWeek, time TimeOfDay }, stopTime SEQUENCE { day DayOfWeek, time TimeOfDay } location Location }	plusieurs par utilisateur du service	recherche	service: L gestion utilisateur: L/E utilisateur: L
Localisation courante	Location	une par utilisateur	recherche	service: L utilisateur: L/E
Expiration de la localisation courante	TimeOfDay	une par utilisateur	néant	service: L utilisateur: L/E
Expiration par défaut	TimeOfDay	une par utilisateur	néant	service: L gestion utilisateur: L/E utilisateur: L
Localisation par défaut	Location	une par utilisateur	recherche	service: L gestion utilisateur: L/E utilisateur: L

I.3.1.2 Élément du service – Enregistrement d'appel entrant

Ce service permet à l'utilisateur d'enregistrer sa localisation courante. La localisation courante peut être réglée soit sur le terminal courant (identifié par CLI ou une localisation définie). L'utilisateur peut facultativement régler une heure pour l'expiration de la localisation courante. Si aucune heure d'expiration est réglée, l'horaire d'expiration par défaut est utilisée. L'emplacement courant doit faire partie d'un ensemble de localisations prédéfinies.

Informations relatives à l'élément de service

Données	Définition	Cardinalité	Opérations des éléments de service	Contrôle d'accès
Localisations enregistrées	SET OF Location	une par utilisateur du service	néant	service: L gestion utilisateur: L/E utilisateur: L
Localisation courante	Location	une par utilisateur	modification entrée	service: L utilisateur: L/E
Expiration localisation courante	TimeOfDay	une par utilisateur	modification entrée	service: L utilisateur: L/E
Expiration par défaut	TimeOfDay	une par utilisateur	néant	service: L gestion utilisateur: L/E utilisateur: L

I.3.1.3 Elément de service – Filtrage d'appel entrant

Cet élément de service permet à l'utilisateur de rejeter un appel entrant s'il provient d'une localisation spécifique. Il n'y a pas de limite au nombre de localisations qui peuvent être utilisées pour le filtrage.

Informations relatives à l'élément de service

Données	Définition	Cardinalité	Opérations des éléments de service	Contrôle d'accès
Incall Screen List (Liste filtrage d'appel entrant)	SET OF Location SIZE (1..MaxListSize)	une par utilisateur du service	recherche	service: L gestion utilisateur: L/E utilisateur: L
MaxListSize	INTEGER	une par utilisateur du service	néant	service: L gestion service: L/E

I.3.1.4 Elément de service – Appel sortant

Cet élément de service permet à l'utilisateur d'effectuer un appel sortant à partir d'une localisation terminale. Cette localisation doit être l'une des localisations autorisées pour l'utilisateur. Le coût de l'appel est facturé à l'utilisateur du service et non pas au propriétaire du terminal.

Informations relatives à l'élément de service

Données	Définition	Cardinalité	Opérations des éléments de service	Contrôle d'accès
Registered Locations (Localisations enregistrées)	SET OF Location	une par utilisateur du service	recherche	service: L gestion utilisateur: L/E utilisateur: L
User Id (Identification de l'utilisateur)	ServiceNumber	une par utilisateur du service	coïncidence/ rattachement	gestion service: L/E service: L gestion utilisateur: L utilisateur: L
User Password	Password	une par utilisateur du service	coïncidence/ rattachement	service: L gestion utilisateur: L/E utilisateur: L

I.3.1.5 Elément de service – Filtrage d'appel sortant par localisation

Cet élément de service permet d'empêcher que des appels sortants soient transmis vers des adresses spécifiques. La liste des adresses prescrites est différente pour chacune des localisations que l'utilisateur peut utiliser.

Informations relatives à l'élément de service

Données	Définition	Cardinalité	Opérations des éléments de service	Contrôle d'accès
Outcall Screen List (Liste de filtrage des appels sortants)	SET OF { SEQUENCE { stopList SET OF Location, location Location } }	une par utilisateur du service	recherche	gestion utilisateur: L/E utilisateur: L

I.3.1.6 Informations générales relatives au service

Les éléments de service ont les interactions suivantes:

- le filtrage d'appel entrant est effectué avant l'appel entrant;
- l'appel sortant est effectué avant le filtrage d'appel sortant par localisation.

Informations complémentaires de service

Les paramètres de services complémentaires suivants sont fournis:

Données	Définition	Cardinalité	Opérations des éléments de service	Contrôle d'accès
Service Key	ServiceId	une par utilisateur du service	coïncidence/ rattachement	gestion service: L/E service: L gestion utilisateur: L utilisateur: L
User Name	String	une par utilisateur du service	coïncidence	gestion utilisateur: L/E utilisateur: L

I.3.2 Exemple de modelage des données de service

Maintenant que les informations de données de service ont été définies, peut commencer la mise en correspondance des données sur le modèle d'information selon les étapes décrites en I.2.

I.3.2.1 Sélection des attributs

La première étape est de déterminer les définitions d'ATTRIBUT nécessaires pour les données de service. Les identificateurs d'objets suivants sont utilisés pour la définition des identificateurs d'attributs:

exampleAttribute OBJECT IDENTIFIER ::= { in-oi exampleService(1) attributeType(4) }

Données – Enregistreur de localisation

L'enregistreur de localisation enregistre les localisations où un utilisateur peut enregistrer pour recevoir les appels entrants.

Localisations enregistrées	ENSEMBLE DE localisations	une par utilisateur du service	néant	service: L gestion utilisateur: L/E utilisateur: L
----------------------------	---------------------------	--------------------------------	-------	--

Pour mieux illustrer cet exemple, la définition d'une localisation peut être considérée comme l'équivalent d'un numéro de téléphone. Cela permet d'utiliser des définitions de SOUS-TYPE pour les attributs de service qui utilisent des définitions de types de localisation à partir d'un attribut prédéfini dans la Recommandation X.520 (telephoneNumber). La définition d'attribut d'enregistreur de localisation devient alors:

```
regLocations ATTRIBUTE ::= {
    SUBTYPE OF telephoneNumber,
    COLLECTIVE TRUE,
    ID in-at-regLocations }
```

Cet attribut est défini comme étant de type collectif car ses valeurs seront réparties entre un certain nombre d'entrées dans l'arbre DIT (voir I.3.3.2).

Données – Table d'acheminement horaire (Time Of Day Location)

La table d'acheminement horaire est en réalité une table de traduction comprenant 5 colonnes (startTime.day, startTime.time, stopTime.day, stopTime.time, location).

Time Of Day Location (Acheminement horaire)	SEQUENCE { startTime SEQUENCE { day DayOfWeek, time TimeOfDay }, stopTime SEQUENCE { day DayOfWeek, time TimeOfDay } location Location }	plusieurs par utilisateur du service	recherche	service: L gestion utilisateur: L/E utilisateur: L
--	---	---	-----------	--

En utilisant la Directive 3, cinq ATTRIBUTS différents peuvent être définis, un par colonne:

```
startDay ATTRIBUTE ::= {
  WITH SYNTAX      DaysOfWeek           -- from proposed X.520 Temporal Context
  SINGLE VALUE     TRUE
  ID               in-at-startDay }
```

```
startTime ATTRIBUTE ::= {
  WITH SYNTAX      DayTime               -- from proposed X.520 Temporal Context
  EQUALITY MATCHING RULE numericStringMatch -- from X.520
  ORDERING MATCHING RULE numericStringOrderingMatch -- from X.520
  SUBSTRINGS MATCHING RULE numericStringSubstringsMatch -- from X.520
  SINGLE VALUE     TRUE
  ID               in-at-startTime }
```

```
stopDay ATTRIBUTE ::= {
  WITH SYNTAX      DaysOfWeek           -- from proposed X.520 Temporal Context
  SINGLE VALUE     TRUE
  ID               in-at-stopDay }
```

```
stopTime ATTRIBUTE ::= {
  WITH SYNTAX      DayTime               -- from proposed X.520 Temporal Context
  EQUALITY MATCHING RULE numericStringMatch -- from X.520
  ORDERING MATCHING RULE numericStringOrderingMatch -- from X.520
  SUBSTRINGS MATCHING RULE numericStringSubstringsMatch -- from X.520
  SINGLE VALUE     TRUE
  ID               in-at-stopTime }
```

```
location ATTRIBUTE ::= {
  SUBTYPE OF      telephoneNumber
  SINGLE VALUE     TRUE
  ID               in-at-location }
```

Cet attribut nécessite une configuration additionnelle afin de limiter ses valeurs à celles qui sont contenues dans l'attribut "regLocations". Cette opération sera effectuée ultérieurement, au moyen d'un attribut de contrôle d'accès dans l'entrée contenant les deux attributs "regLocations" et "location".

Données – Localisation courante

La localisation courante est définie selon une localisation enregistrée pour un utilisateur. Elle a une durée de validité spécifique.

Localisation courante	Localisation	une par utilisateur	recherche	service: L utilisateur: L/E
-----------------------	--------------	---------------------	-----------	--------------------------------

La définition de l'attribut de l'emplacement courant devient donc:

```
currentLocation ATTRIBUTE ::= {
  SUBTYPE OF      telephoneNumber
  SINGLE VALUE     TRUE
  ID               in-at-cur-location }
```

Cet attribut nécessite une configuration additionnelle afin de limiter ses valeurs à celles qui sont contenues dans l'attribut "regLocations". Cette opération sera effectuée ultérieurement, au moyen d'un attribut de contrôle d'accès dans l'entrée contenant les deux attributs "regLocations" et "currentLocation".

On spécifie la durée d'application de cet attribut en associant à la valeur un contexte temporel dont le champ "endTime" est mis à la valeur d'expiration de l'emplacement courant ou à celle de l'expiration par défaut de l'emplacement courant. La sélection du champ "endTime" à utiliser est régie par la fonction qui fixe la valeur de l'attribut⁵⁾.

Données – Expiration de la localisation courante

Les données relatives à l'expiration de la localisation courante contiennent l'heure d'expiration pour les données de localisation courante.

Expiration de la localisation courante	TimeOfDay	une par utilisateur	néant	service: L utilisateur: L/E
--	-----------	---------------------	-------	--------------------------------

Etant donné que les données d'expiration de la localisation courante ne sont utilisées que pour régler la durée d'application de l'attribut "currentLocation", ces données ne sont pas conservées dans l'Annuaire en tant qu'attribut distinct. Elles sont cependant présentes sous la forme d'une valeur de contexte temporel associé à la valeur de l'attribut "currentLocation". On notera qu'il s'agit d'une valeur fournie facultativement par l'utilisateur. Si elle n'est pas fournie, on utilise à sa place la valeur d'expiration par défaut. Cette sélection est gérée par l'action FEA d'entité SCF chargée de gérer l'attribut.

Données – Expiration par défaut

Les données relatives à l'expiration par défaut contiennent l'heure d'expiration par défaut pour les données de fin de localisation courante.

Expiration par défaut	TimeOfDay	une par utilisateur	néant	service: L gestion utilisateur: L/E utilisateur: L
-----------------------	-----------	---------------------	-------	--

Etant donné que les données d'expiration par défaut ne sont utilisées que pour régler la durée d'application de l'attribut "currentLocation", ces données ne sont pas conservées dans l'Annuaire en tant qu'attribut distinct. Elles sont cependant présentes sous la forme d'une valeur de contexte temporel associée à la valeur de l'attribut "currentLocation". On notera que cette valeur n'est utilisée que si l'utilisateur ne fournit pas de valeur pour l'expiration de la localisation courante. Cette sélection est gérée par l'action FEA d'entité SCF chargée de gérer l'attribut.

Données – Localisation par défaut

La localisation par défaut définit la localisation à utiliser si toute autre option d'acheminement échoue.

Localisation par défaut	Localisation	une par utilisateur	recherche	service: L gestion utilisateur: L/E utilisateur: L
-------------------------	--------------	---------------------	-----------	--

La définition de l'attribut de la localisation par défaut devient alors:

```
defaultLocation ATTRIBUTE ::= {
    SUBTYPE OF      telephoneNumber
    SINGLE VALUE    TRUE
    ID              in-at-def-location }
```

⁵⁾ Bien que pouvant sembler exiger des informations propres au service dans l'action FEA d'entité SCF chargée de gérer l'attribut, ce concept de durée d'application est suffisamment général (c'est-à-dire qu'il peut être appliqué à un grand nombre d'attributs indépendants) pour qu'un mécanisme normalisé puisse être mis au joint afin de traiter ce concept de façon indépendante du service.

Cet attribut nécessite une configuration additionnelle afin de limiter ses valeurs à celles qui sont contenues dans l'attribut "regLocations". Cette opération sera effectuée ultérieurement, au moyen d'un attribut de contrôle d'accès dans l'entrée contenant les deux attributs "regLocations" et "defaultLocation".

Données – Liste de filtrage d'appel entrant

La liste de filtrage Incall définit une liste de localisations, avec une longueur maximale spécifique, à partir desquelles les appels vers l'utilisateur doivent être rejetés.

Liste de filtrage Incal	SET OF Location SIZE 1..MaxListSize	une par utilisateur de service	recherche	service: gestion utilisateur: utilisateur:	L L/E L
-------------------------	--	-----------------------------------	-----------	--	---------------

La liste de filtrage de l'appel en cours pourra être considérée de deux façons, soit comme une table avec colonne unique, soit comme un attribut avec des valeurs multiples. Dans ce cas, on pourra utiliser sans difficulté la version d'attribut à valeurs multiples, car l'opération contrôle simplement la présence des valeurs spécifiques et n'essaie pas de récupérer la valeur.

La définition de l'attribut Liste de filtrage d'appel entrant devient alors:

```
incallScreenList ATTRIBUTE ::= {
    SUBTYPE OF      telephoneNumber,
    ID              in-at-incall-screen }
```

Cet attribut nécessite une configuration additionnelle afin de limiter le nombre de valeurs qu'il peut contenir. Cette opération sera effectuée ultérieurement, au moyen d'un attribut de contrôle d'accès dans l'entrée contenant l'attribut "IncallScreenList".

Données – MaxListSize (taille maximale de la liste)

Les données MaxListSize contiennent la longueur maximale de la liste de filtrage d'appel entrant.

MaxListSize	INTEGER	une par utilisateur de service	néant	service: gestion service:	L L/E
-------------	---------	-----------------------------------	-------	------------------------------	----------

Etant donné que les données de taille maximale de la liste ne sont utilisées que pour régler le nombre maximal de valeurs de l'attribut "IncallScreenList", ces données ne sont pas conservées dans l'Annuaire en tant qu'attribut distinct. Elles sont cependant présentes sous la forme d'un item protégé de type maxValueCount dans un attribut de contrôle d'accès contenu dans l'entrée contenant l'attribut "IncallScreenList".

Données – Identification de l'utilisateur

Les données UserId définissent le nombre utilisé pour identifier l'utilisateur d'un service. Quand il est associé à un code préfixe de service (014 par exemple) ce nombre forme une localisation.

User Id	ServiceNumber	une par utilisateur de service	coïncidence/ rattachement	gestion service: service: gestion utilisateur: utilisateur:	L/E L L L
---------	---------------	-----------------------------------	------------------------------	--	--------------------

La définition de l'attribut d'identification de l'utilisateur devient donc:

```
userId ATTRIBUTE ::= {
    WITH SYNTAX      NumericString
    EQUALITY MATCHING RULE  numericStringMatch      -- from X.520
    ORDERING MATCHING RULE  numericStringOrderingMatch -- from X.520
    SUBSTRINGS MATCHING RULE  numericStringSubstringsMatch -- from X.520
    SINGLE VALUE
    ID              in-at-user-id }
```

Données – Mot de passe utilisateur

Le mot de passe utilisateur contient le numéro personnel d'identification. Il est utilisé pour valider l'enregistrement et les informations d'appel sortant.

User Password	Password	une par utilisateur de service	coïncidence/ rattachement	service: L gestion utilisateur: L/E utilisateur: L
---------------	----------	--------------------------------	------------------------------	--

Etant donné que cet attribut est utilisé dans l'opération de rattachement, le mot de passe utilisateur doit utiliser la définition de l'attribut "userPassword" figurant au 6.3/X.509.

Données – Liste de filtrage d'appel sortant

La liste de filtrage d'appel sortant fournit une liste de localisations interdites pour chaque enregistrement de localisation dont l'utilisateur dispose. C'est un tableau à deux colonnes (regLocation, stopList).

Outcall Screen List (Liste de filtrage Outcall)	SET OF { SEQUENCE { stopList SET OF Location, location Location } }	une par utilisateur de service	recherche	gestion utilisateur: L/E utilisateur: L
---	--	--------------------------------------	-----------	--

Deux attributs sont définis pour la liste de filtrage d'appel sortant:

```
outcallScreenLocation ATTRIBUTE ::= {  
    SUBTYPE OF     telephoneNumber  
    SINGLE VALUE   TRUE  
    ID             in-at-outcall-location }
```

Cet attribut nécessite une configuration additionnelle afin de limiter ses valeurs à celles qui sont contenues dans l'attribut "regLocations". Cette opération sera effectuée ultérieurement, au moyen d'un attribut de contrôle d'accès dans l'entrée contenant les deux attributs "regLocations" et "outcallScreenLocation".

```
outcallScreenStopList ATTRIBUTE ::= {  
    SUBTYPE OF     telephoneNumber  
    ID             in-at-outcall-stoplist }
```

Données – Service Key (Clé de service)

Les données Service Key identifient le service en cours d'utilisation.

Service Key	ServiceId	une par service	coïncidence/ rattachement	gestion service: L/E service: L gestion utilisateur: L utilisateur: L
-------------	-----------	-----------------	------------------------------	--

La définition de l'attribut Service Key devient donc:

```
serviceKey ATTRIBUTE ::= {  
    WITH SYNTAX   Object Identifier  
    SINGLE VALUE   TRUE  
    ID             in-at-service-key }
```

Données – User Name (nom d'utilisateur)

Les données User Name comprennent le nom d'utilisateur du service.

User Name	String	une par utilisateur de service	coïncidence	gestion utilisateur: L/E utilisateur: L
-----------	--------	--------------------------------	-------------	--

La définition de l'attribut UserName devient donc:

```
userName ATTRIBUTE ::= {
    SUBTYPE OF name,
    ID in-at-user-name }
-- from X.520
```

I.3.2.2 Sélection des CLASSES D'OBJETS

Une fois que les ATTRIBUTS ont été sélectionnés, ils peuvent être organisés en CLASSES D'OBJETS. En utilisant la Directive 5, les attributs suivants peuvent être groupés (les attributs entre [] indiquent des attributs référencés):

- startDay, startTime, stopDay, stopTime, location, [regLocations];
- outcallScreenLocation, outcallScreenStopList, [regLocations].

Les attributs restants doivent être groupés selon la Directive 4. Etant donné qu'on peut considérer que les attributs qui partagent la même cardinalité partagent la même identité, les groupements suivants sont définis:

- serviceKey;
- currentLocation, defaultLocation, incallScreenList, userId, userPassword, userName, [regLocations].

Les attributs peuvent être obligatoires ou facultatifs dans une classe d'objets. On peut déterminer, à partir de la logique de service, que les attributs currentLocation et defaultLocation sont facultatifs. De la même façon, un attribut qui est groupé comme faisant partie d'une table doit être considéré comme un attribut obligatoire. D'autres attributs devraient être rendus obligatoires également.

Les définitions des CLASSES D'OBJETS requises sont les suivantes:

```
serviceClass OBJECT-CLASS ::= {
    MUST CONTAIN serviceKey
    ID in-oc-service }

serviceUserClass OBJECT-CLASS ::= {
    MUST CONTAIN userId, userPassword, userName, incallScreenList, reqLocations,
    MAY CONTAIN currentLocation, defaultLocation
    ID in-o-service-user }

timeOfDayLocation OBJECT-CLASS ::= {
    MUST CONTAIN startDay, startTime, stopDay, stopTime, location, reqLocations
    ID in-oc-time-location}

outcallScreenList OBJECT-CLASS ::= {
    MUST CONTAIN outcallScreenLocation, outcallScreenStopList, reqLocations
    ID in-oc-outcall-screen}
```

La définition des attributs de contrôle d'accès utilisés pour effectuer le contrôle d'accès par attributs a été différée jusqu'à ce que la conception de l'arbre DIT soit terminée (voir I.3.2.4).

Une fois les classes d'objets définies, les attributs de nommage des objets doivent être assignés. Cela nécessite l'identification de l'un des attributs obligatoires qui possède une valeur unique pour chaque occurrence de l'objet.

L'étude des objets définis permet d'utiliser les attributs de nommage suivants:

```
serviceClass      serviceKey
serviceUserClass  userId
timeOfDayLocation ? (aucun attribut n'est utilisable)
outcallScreenList outcallScreenLocation
```

Le seul objet qui pose un problème de nommage est l'objet `timeOfDayLocation`. Il y a deux moyens de résoudre ce problème. Le premier consiste à définir un attribut unique (`tableRowIndex`, par exemple) pour permettre le nommage. Cela n'a pas d'incidence sur l'exécution des opérations et peut s'avérer utile dans les opérations de gestion qui ne sont pas prises en compte ici.

Le second consiste à associer les attributs `StartDay` et `StartTime` pour former un attribut composite, dont les valeurs seraient alors uniques. Cette méthode impliquerait de redéfinir les RÈGLES DE CORRESPONDANCE et des traitements additionnels.

On retiendra la première solution pour sa simplicité. Les formes de noms nécessaires avec le nouvel attribut et avec la classe d'objets révisée sont les suivants:

```

tableRowIndex ATTRIBUTE ::= {
    WITH SYNTAX Integer
    SINGLE VALUE TRUE
    ID in-at-maxListValues }

timeOfDayLocation OBJECT-CLASS ::= {
    MUST CONTAIN tableRowIndex, startDay, startTime, stopDay, stopTime, location,
    regLocations
    ID in-oc-time-location }

serviceName NAME-FORM ::= {
    NAMES serviceClass
    WITH ATTRIBUTES serviceKey
    ID in-nf-service-name }

serviceUserName NAME-FORM ::= {
    NAMES serviceUserClass
    WITH ATTRIBUTES userId
    ID in-nf-service-user-name }

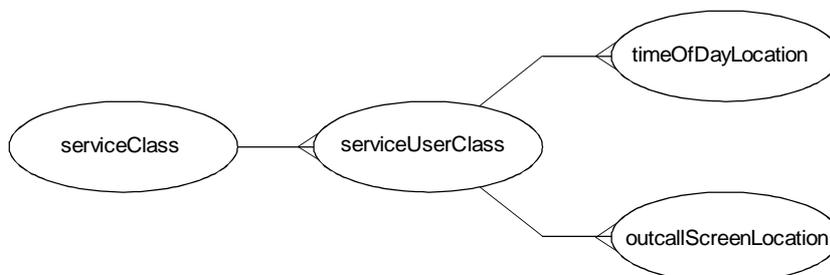
timeOfDayLocationName NAME-FORM ::= {
    NAMES timeOfDayLocation
    WITH ATTRIBUTES tableRowIndex
    ID in-nf-tod-location-name }

outcallScreenName NAME-FORM ::= {
    NAMES outcallScreenList
    WITH ATTRIBUTES outcallScreenLocation
    ID in-nf-outcall-screen-name }

```

I.3.2.3 Conception de l'arbre DIT du service

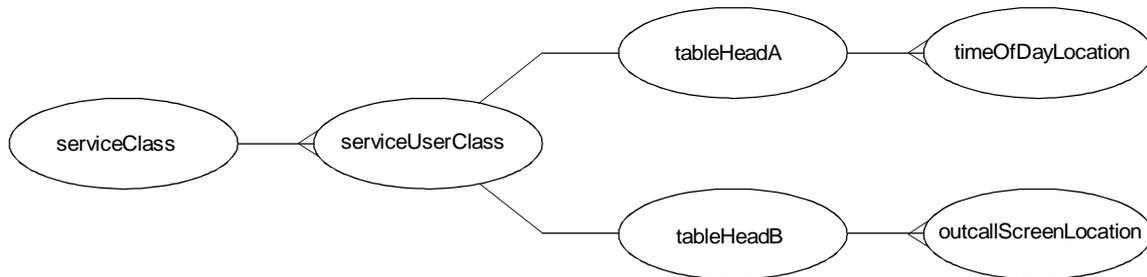
La conception de l'arbre DIT peut commencer à ce point. A partir des informations de cardinalité, les relations de CLASSES D'OBJETS suivantes peuvent être obtenues (voir la Figure I.4).



T1169790-94/d105

FIGURE I.4/Q.1218
Relations entre classes d'objets

Etant donné que la "classe serviceUserClass" a deux ensembles d'entrées qui lui sont subordonnés, il y a lieu d'appliquer la Directive 7, qui fera ajouter un niveau d'objet supplémentaire, comme indiqué à la Figure I.5.



T1169800-94/d106

FIGURE I.5/Q.1218
Relations entre classes d'objets selon la Directive 7

Cela nécessite une classe d'objet et des définitions d'attribut supplémentaires, telles que:

```

tableHeadIdentifier ATTRIBUTE ::= {
    WITH SYNTAX Integer
    SINGLE VALUE TRUE
    ID oi-gen-table-head }

tableHead OBJECT-CLASS ::= {
    MUST CONTAIN tableHeadIdentifier
    ID in-oc-tableHead }

tableHeadA OBJECT-CLASS ::= {
    MUST CONTAIN tableHeadIdentifier
    ID in-oc-tableHeadA }

tableHeadA NAME-FORM ::= {
    NAMES tableHeadA
    WITH ATTRIBUTEs tableHeadIdentifier
    ID in-nf-tableHeadNameA }

tableHeadB OBJECT-CLASS ::= {
    MUST CONTAIN tableHeadIdentifier
    ID in-oc-tableHeadB }

tableHeadNameB NAME-FORM ::= {
    NAMES tableHeadB
    WITH ATTRIBUTEs tableHeadIdentifier
    ID in-nf-tableHeadNameB }
  
```

Par conséquent, les règles de structure requises pour définir la structure de l'arbre DIT sont les suivantes:

```

sr-root STRUCTURE-RULE ::= {
    NAME FORM serviceClassName
    ID sr0 }
  
```

```

sr-user STRUCTURE-RULE ::= {
    NAME FORM          serviceClassName
    SUPERIOR RULES    sr0
    ID                 sr1 }

sr-tableA STRUCTURE-RULE ::= {
    NAME FORM          tableHeadNameA
    SUPERIOR RULES    sr1
    ID                 sr2 }

sr-tod-location STRUCTURE-RULE ::= {
    NAME FORM          timeOfDayLocationName
    SUPERIOR RULES    sr2
    ID                 sr3 }

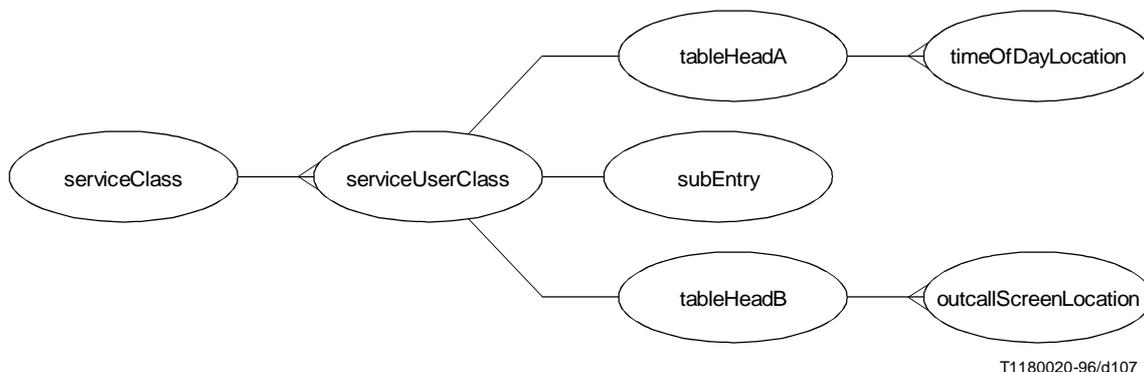
sr-tableB STRUCTURE-RULE ::= {
    NAME FORM          tableHeadNameB
    SUPERIOR RULES    sr1
    ID                 sr4 }

sr-outcall-screen STRUCTURE-RULE ::= {
    NAME FORM          outcallScreenListName
    SUPERIOR RULES    sr4
    ID                 sr5 }

```

La création d'une entrée de la CLASSE D'OBJETS "subEntry" implique la spécification du sous-arbre auquel les attributs collectifs s'appliquent. Dans le cas de l'attribut collectif "regLocations", la sous-entrée est subordonnée à une entrée de la CLASSE D'OBJETS "serviceUserClass".

La modification de l'attribut "regLocation" implique la fourniture du nom distinctif de la sous-entrée (voir la Figure I.6).



T1180020-96/d107

FIGURE I.6/Q.1218

Arbre d'annuaire montrant l'emplacement de la classe des sous-entrées

I.3.2.4 Attribution du contrôle d'accès

Une fois l'arbre DIT défini, l'étape suivante à franchir consiste à définir les valeurs d'attribut de contrôle d'accès pour chacune des entrées de l'arbre DIT. Un attribut de contrôle d'accès pour une entrée (entryACI, voir 16.5.2/X.501) contient les valeurs de type ACIItem qui déterminent la façon dont l'utilisateur peut accéder aux valeurs de l'attribut. Le tableau suivant définit le contrôle d'accès pour chacun des attributs définis dans l'arbre DIT.

Objet	Attribut	Classes d'utilisateurs définies par gestion X.500					Items protégés spéciaux
		Utilisateur-X	Tous utilisateurs	Gestion d'utilisateur	Service	Gestion de service	
serviceClass	serviceKey	+R+F	+F	+R+F	+R+F	+R+A+D+F	
serviceUserClass	userId	+R+F	+F	+R+F	+R+F	+R+A+D+F	
	userPassword	+R+F+C		+R+A+D+F	+R+F	+R+F	
	userName	+R+F+C		+R+A+D+F	+R+F	+R+F	
	incallScreenList	+R+F		+R+A+D+F(*)	+R+F	+R+F	maxValueCount
	currentLocation	+R+A+D+F(*)		+R+F	+R+F	+R+F	restrictedBy
	defaultLocation	+R+A+D+F(*)		+R+F	+R+F	+R+F	restrictedBy
tableHeadA	tableHead Identifier	+R+F		+R+F	+R+F	+R+A+D+F	
timeOfDay Location	tableRowIndex	+R+F		+R+A+D+F	+R+F	+R+F	
	startDay	+R+F		+R+A+D+F	+R+F	+R+F	
	startTime	+R+F		+R+A+D+F	+R+F	+R+F	
	stopDay	+R+F		+R+A+D+F	+R+F	+R+F	
	stopTime	+R+F		+R+A+D+F	+R+F	+R+F	
	location	+R+F		+R+A+D+F(*)	+R+F	+R+F	restrictedBy
tableHeadB	tableHead_ Identifier	+R+F		+R+F	+R+F	+R+A+D+F	
outcallScreenList	outcallScreen_ Location	+R+F		+R+A+D+F(*)	+R+F	+R+F	restrictedBy
	outcallScreen_ StopList	+R+F		+R+A+D+F	+R+F	+R+F	
serviceUser_ Subentry	regLocations	+R+F		+R+A+D+F	+R+F	+R+F	
+ Octroi de permission		A Adjonction d'attribut à une entrée		M Modification d'attribut			
- Refus de permission		R Lecture d'accès		C Comparaison d'attribut			
* Item protégé spécial		D Suppression d'accès		F Filtrage d'attributs pour vérifier coïncidence			

Chaque colonne du tableau sous la rubrique "Classes d'utilisateurs définies par gestion X.500" indique une valeur possible de l'élément ACIItem pour l'attribut "entryACI". En pratique, ces valeurs peuvent être combinées en un seul élément ACIItem regroupant un ensemble d'items protégés ayant les mêmes autorisations d'accès dans un ensemble de classes d'utilisateur. Chaque élément ACIItem possède un champ d'identification unique, qui peut être utilisé pour appliquer le contrôle d'accès à des valeurs individuelles de l'attribut de contrôle d'accès proprement dit. Cela permet à différents "utilisateurs" de modifier différentes valeurs de contrôle d'accès (qui peuvent contenir des limites de valeur autorisée ou de taille) sans qu'ils aient la possibilité de modifier d'autres valeurs de l'attribut de contrôle d'accès.

Lorsque des autorisations sont assorties d'un astérisque (*), cela signifie qu'il faut inclure un item protégé spécial, du type spécifié dans la dernière colonne (intitulée "Items protégés spéciaux"), dans l'élément ACIItem afin de traiter les commandes de limite relatives à des valeurs autorisées ou à un décompte de valeurs.

I.4 Définition des contextes

Ce paragraphe définit un ensemble de types de contextes qui peuvent se révéler utiles pour toute une gamme de services du réseau intelligent.

I.4.1 Contexte de valeur d'attribut d'index numérique

Le contexte de valeur d'attribut d'index numérique associe un index numérique à une valeur d'attribut.

```
numericalIndexAVC    ATTRIBUTE-VALUE-CONTEXT ::= {  
    SYNTAX            INTEGER  
    id                id-avc-numericalIndex }
```

Un exemple d'utilisation du contexte valeur d'attribut numérique est la table de conversion "numéro abrégé vers numéro complet" où l'index (la valeur du contexte) est le numéro abrégé et où la valeur d'attribut représente le numéro complet.

Noter qu'un type similaire de contexte de valeur d'attribut d'index de chaîne de caractères pourrait être également défini.

Appendice II

Aspects relatifs à l'interface du réseau intelligent devant faire l'objet d'un complément d'étude en ce qui concerne l'ensemble de capacités CS-1

(Cet appendice ne fait pas partie intégrante de la présente Recommandation)

II.1 Généralités

II.1.1 Considérations générales

Le présent appendice porte sur la gestion de participant et d'autres points qui se sont révélés être incomplets lors de l'élaboration de l'ensemble de capacités CS-1 de l'interface de réseau intelligent de la nouvelle Recommandation Q.1218. Bien que les éléments contenus dans cet appendice aient été élaborés autour de l'ensemble CS-1, les procédures applicables à ces capacités peuvent s'avérer insuffisamment définies et demander un complément d'étude. Les éléments inclus dans le présent appendice contiennent une certaine base technique destinée aux travaux futurs.

II.1.2 Relation avec les autres appendices des Recommandations de la série Q.1200

Le présent appendice ne concerne que l'ensemble de capacités CS-1 de l'interface de réseau intelligent de la Recommandation Q.1218. Chaque Recommandation de la série Q.1200 comporte le cas échéant un appendice spécifique.

II.1.3 Format du document

La présente introduction donne une explication de l'objet et du champ d'application de l'appendice.

- Le paragraphe II.2 contient la liste des opérations.
- Le paragraphe II.3 contient la liste des paramètres.
- Le paragraphe II.4 est un module ASN.1 des opérations et des paramètres.
- Le paragraphe II.5 contient les procédures applicables aux opérations.

II.2 Opérations

Les opérations citées dans le présent paragraphe viennent s'ajouter aux opérations énumérées dans l'article 2.

II.2.1 Considérations applicables à toutes les opérations de cet appendice

Les opérations ou aspects d'opérations ci-après nécessitent un complément d'étude en ce qui concerne l'ensemble de capacités CS-1. Ces opérations reposent sur les capacités de l'ensemble CS-1 pour lesquelles les procédures correspondantes ne sont pas totalement définies. Par conséquent, elles figurent dans le présent appendice afin d'être complétées.

Les options définies par défaut peuvent être incomplètes (par exemple identificateur de demi-appel) et peuvent dépendre du point PIC (*point in call*) utilisé.

II.2.2 Opération Add Party (Ajouter un participant)

II.2.2.1 Considérations générales

La différence entre cette opération et l'opération attach (attacher) doit être précisée.

II.2.2.2 Description

SCF → SSF

Cette opération est utilisée pour exécuter les actions de traitement d'appel et ajouter toutes les connexions de participant d'un appel à un autre, puis libérer le premier appel (par exemple pour créer une communication conférence). Vu de l'entité de commande, cette opération réunit effectivement deux appels.

II.2.3 Opération Attach (Attacher)

II.2.3.1 Considérations générales

La différence entre cette opération et l'opération add party (ajouter participant) doit être précisée.

II.2.3.2 Description

SCF → SSF

Cette opération permet à la fonction SCF de demander à la fonction SSF d'inclure un demi-appel dans l'instance de relation en cours. Ce demi-appel est transféré à partir d'une autre instance de relation, instance à partir duquel il a été supprimé en utilisant l'opération "detach". Il faut noter que l'opération "detach" peut être exécutée également après l'opération "attach" en utilisant le même identificateur absolu.

II.2.4 Opération Change parties (changement de participants)

II.2.4.1 Description

SCF → SSF

Cette opération est utilisée pour exécuter les actions de traitement d'appel visant à transférer une connexion particulière de correspondant d'un appel à un autre appel. Vu du participant à l'appel considéré, cette opération met effectivement le premier appel en garde et reprend l'appel associé précédemment en garde.

II.2.5 Opérations "Detach" (Détacher)

II.2.5.1 Considérations générales

La différence entre cette opération et l'opération de libération de la connexion d'un correspondant nécessite un complément d'étude.

II.2.5.2 Description

SCF → SSF

Cette opération permet à l'entité SCF de demander à l'entité SSF de supprimer un demi-appel dans une instance relationnelle et de lui attribuer un identificateur absolu (c'est-à-dire reconnu dans tout le réseau). Cet identificateur (de corrélation), permettra de transférer ce demi-appel vers une autre instance relationnelle, à laquelle il était/sera attaché au moyen de l'opération "attach" (attacher) en utilisant le même identificateur absolu.

II.2.6 Opération de mise en garde d'une connexion de correspondant

II.2.6.1 Description

SCF → SSF

Cette opération est effectuée pendant la phase active d'une communication entre deux ou plus de deux correspondants afin de mettre en garde une seule connexion de correspondant.

II.2.7 Opération de tentative d'appel au départ (dans le cas de plus d'un correspondant)

II.2.7.1 Considérations générales

Ce flux d'information est repris dans le corps de la Recommandation Q.1214 pour le cas de la création d'un appel à destination d'un correspondant unique. Ce flux d'information figure dans la liste du présent appendice pour traiter le cas de la création d'un appel à destination de plus d'un correspondant dans la même communication, ce qui, pour l'ensemble CS-1, doit faire l'objet d'un complément d'étude.

II.2.7.2 Description

SCF → SSF

Cette opération sert à demander à l'entité SSF de créer un nouvel appel à destination de plusieurs correspondants en utilisant les informations d'adresse fournies par l'entité SCF (par exemple, les adresses d'une communication de conférence prédéfinie, des informations obtenues par une opération précédente d'invitation et recueil). Toutes erreurs éventuelles associées à l'exécution de cette opération feront l'objet d'un retour de résultat.

II.2.8 Opération de reconnexion

II.2.8.1 Description

SCF → SSF

Cette opération sert à remettre en communication un correspondant préalablement mis en garde (c'est l'inverse de l'opération de mise en garde d'une connexion de correspondant).

II.2.9 Opération de libération de la connexion d'un correspondant

II.2.9.1 Considérations générales

La différence entre cette opération et l'opération de détachement doit faire l'objet d'un complément d'étude.

II.2.9.2 Description

SCF → SSF

Cette opération sert à libérer une connexion de correspondant pendant une communication entre deux ou plus de deux correspondants.

II.3 Paramètres

Les paramètres énumérés dans ce paragraphe sont des paramètres additionnels pour les opérations répertoriées à l'article 2.

II.3.1 Considérations applicables à tous les paramètres de cet appendice

Les paramètres ci-après doivent faire l'objet d'un complément d'étude en ce qui concerne l'ensemble CS-1. Ces paramètres sont fondés sur l'ensemble de capacités CS-1, pour lequel on doit définir les procédures correspondantes. Ils ne sont donc inclus dans cet appendice que pour mémoire.

II.3.2 Paramètre identification de demi-appel créé (issu de l'opération d'analyse des informations)

II.3.2.1 Description

DEFAULT bPARTY (Correspondant B par défaut)

Ce paramètre indique une référence à un correspondant spécifique dans une communication. Le qualificatif OPTIONAL (facultatif) signale que l'utilisation de ce paramètre est réservée à l'exploitation du réseau, qui a le choix entre une affectation unilatérale ou bilatérale de l'identification.

II.3.3 Paramètre Identification de demi-appel créé (issu de l'opération de connexion)

II.3.3.1 Description

DEFAULT bPARTY (Correspondant B par défaut)

Ce paramètre indique une référence à un correspondant spécifique dans une communication. Le qualificatif OPTIONAL (facultatif) signale que l'utilisation de ce paramètre est réservée à l'exploitation du réseau, qui a le choix entre une affectation unilatérale ou bilatérale de l'identification.

II.3.4 Paramètre Identification de demi-appel créé (issu de l'opération de tentative d'appel au départ)

II.3.4.1 Description

DEFAULT bPARTY (Correspondant B par défaut)

Ce paramètre indique une référence à un correspondant spécifique dans une communication. Le qualificatif OPTIONAL (facultatif) signale que l'utilisation de ce paramètre est réservée à l'exploitation du réseau, qui a le choix entre une affectation unilatérale ou bilatérale de l'identification.

II.3.5 Paramètre Identification de demi-appel créé (issu de l'opération de sélection d'équipement)

II.3.5.1 Description

DEFAULT bPARTY (Correspondant B par défaut)

Ce paramètre indique une référence à un correspondant spécifique dans une communication. Le qualificatif OPTIONAL (facultatif) signale que l'utilisation de ce paramètre est réservée à l'exploitation du réseau, qui a le choix entre une affectation unilatérale ou bilatérale de l'identification.

II.3.6 Paramètre Identification de demi-appel créé (issu de l'opération de sélection d'acheminement)

II.3.6.1 Description

DEFAULT bPARTY (Correspondant B par défaut)

Ce paramètre indique une référence à un correspondant spécifique dans une communication. Le qualificatif OPTIONAL (facultatif) signale que l'utilisation de ce paramètre est réservée à l'exploitation du réseau, qui a le choix entre une affectation unilatérale ou bilatérale de l'identification.

II.3.7 Paramètre demi-appel 1 (issu de l'opération du point de détection initial)

II.3.7.1 Description

DEFAULT {aPARTY, pending} (Correspondant A par défaut, en attente)

Ce paramètre indique une information relative à un correspondant telle que définie par un objet demi-appel. Cette information comporte un élément d'identification de demi-appel (LegId) pour désigner chaque correspondant et un élément d'état du demi-appel (LegStatus) pour indiquer si le correspondant est ou non connecté.

II.3.8 Paramètre demi-appel 2 (issu de l'opération du point de détection initial)

II.3.8.1 Description

FACULTATIF

Ce paramètre indique une information relative à un correspondant telle que définie par un objet demi-appel. Cette information comporte un élément d'identification de demi-appel (LegId) pour désigner chaque correspondant et un élément d'état du demi-appel (LegStatus) pour indiquer si le correspondant est ou non connecté.

II.3.9 Paramètre identification d'appel

II.3.9.1 Description

Indique un identificateur pour désigner une instance d'appel accessible à l'entité SCF. Voir la description du segment d'appel au 4.2.2.1/Q.1214.

II.4 Modules en notation ASN.1 des opérations et des paramètres

Les modules suivants décrivent les opérations additionnelles figurant dans le présent appendice. Ces modules ne décrivent pas les modifications à apporter aux opérations de l'article 2 pour y inclure les paramètres de demi-appel étudiés au II.3.

II.4.1 Syntaxe abstraite du protocole d'application de l'ensemble CS-1 du réseau intelligent – Appendice

Le paragraphe suivant spécifie la syntaxe abstraite additionnelle pour le protocole d'application de l'ensemble CS-1 du réseau intelligent. Elle utilise la notation de syntaxe abstraite numéro un (ASN.1), qui est définie dans la Recommandation X.208.

Les données contenues dans le présent appendice sont fondées sur celles de l'appendice à la Recommandation Q.1214.

Operation types

IN-CS-1-Operations-appendix { ccitt recommendation q 1218 modules(0) cs-1-operations-app(4) version1(0) }

-- This module contains additional type definitions for IN CS-1 operations.

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

**OPERATION,
ERROR**

FROM TCAPMessages { ccitt recommendation q 773 modules(0) messages(1) version2(2) };

-- TYPE DEFINITION FOR ADDITIONAL IN CS-1 OPERATIONS

-- SCF-SSF operations

AddParty ::= OPERATION

ARGUMENT

AddPartyArg

RESULT

CallPartyHandlingResultsArg

ERRORS {

**DataAlreadyExists,
MissingParameter,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- SCF → SSF

-- This operation is used to perform the call processing actions to add all call party connections

-- from one Call to another Call, then clear the first Call (e.g. to create a conference call).

-- From the perspective of the controlling party, this operation effectively bridges two Calls.

Attach ::= OPERATION

ARGUMENT

AttachArg

ERRORS {

**DataAlreadyExists,
LegIDAlreadyAssigned,
MissingParameter,
SystemFailure,
TaskRefused,
TooLate,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- SCF → SSF

-- This operation is used to attach two Calls.

ChangeParties ::= OPERATION

ARGUMENT

ChangePartiesArg

RESULT

CallPartyHandlingResultsArg

ERRORS {

**DataAlreadyExists,
MissingParameter,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- SCF → SSF
-- This operation is used to perform the call processing actions to change a particular party
-- connection from one Call to another Call. From the perspective of the particular call party, this
-- operation effectively places the first Call on hold and retrieves the associated Call from hold.

Detach ::= OPERATION
 ARGUMENT
 DetachArg
 ERRORS {
 DataAlreadyExists,
 MissingParameter,
 SystemFailure,
 TaskRefused,
 TooLate,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter,
 UnknownLegID
 }

-- SCF → SSF
-- This operation is used to detach two Calls.

HoldCallPartyConnection ::= OPERATION
 ARGUMENT
 HoldCallPartyConnectionArg
 RESULT
 CallPartyHandlingResultsArg
 ERRORS {
 DataUnavailable,
 MissingParameter,
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
 }

-- SCF → SSF
-- This operation is used during the active phase of a call between two or more parties to put one
-- party connection on hold.

Reconnect ::= OPERATION
 ARGUMENT
 ReconnectArg
 RESULT
 CallPartyHandlingResultsArg
 ERRORS {
 DataAlreadyExists,
 MissingParameter,
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
 }

-- SCF → SSF
-- This operation is used to resume a held party to a call (inverse of HoldCallPartyConnection).

ReleaseCallPartyConnection ::= OPERATION
 ARGUMENT
 ReleaseCallPartyConnectionArg
 RESULT
 CallPartyHandlingResultsArg
 ERRORS {
 DataAlreadyExists,
 MissingParameter,
 SystemFailure,
 }

```

    TaskRefused,
    UnexpectedComponentSequence,
    UnexpectedDataValue,
    UnexpectedParameter
}
-- SCF → SSF
-- This operation is used to release a call party connection during a call between two or more parties.

```

END

```

IN-CS-1-Errors-appendix { ccitt recommendation q 1218 modules(0) cs-1-errors-app(5) version1(0) }

```

```

-- This module contains additional type definitions for the IN CS-1 errors.

```

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

ERROR

```

FROM TCAPMessages { ccitt recommendation q 773 modules(0) messages(1) version2(2) };

```

```

-- TYPE DEFINITION FOR IN CS-1 ERRORS

```

LegIDAlreadyAssigned ::= ERROR

```

-- Indicates that a legID has already been assigned with the requested value.

```

TooLate ::= ERROR

```

-- Indicates that the operation could not be performed in a timely manner.

```

UnknownLegID ::= ERROR

```

-- Indicates that the legID does not exist.

```

END

Data types

```

IN-CS-1-DataTypes-appendix { ccitt recommendation q 1218 modules(0) cs-1-datatypes-app(6) version1(0) }

```

```

-- This module contains additional type definitions for the IN CS-1 data types.

```

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

```

-- TYPE DEFINITION FOR ADDITIONAL IN CS-1 DATA TYPES

```

```

-- Argument Data Types

```

AddPartyArg ::= SEQUENCE {

```

    originalCallID          [0] CallID          OPTIONAL,
    destinationCallID      [1] CallID          OPTIONAL
}

```

```

-- OPTIONAL denotes network operator specific use.

```

AttachArg ::= SEQUENCE {

```

    newLegID                [0] LegID          OPTIONAL,
    correlationIdentifier    [1] CorrelationID  OPTIONAL
}

```

```

-- OPTIONAL denotes network operator specific use.

```

CallPartyHandlingResultsArg ::= SEQUENCE OF LegInformation

ChangePartiesArg ::= SEQUENCE {

```

    callID                  [0] CallID          OPTIONAL,
    targetCallID           [1] CallID,
    legToBeConnectedID     [2] LegID
}

```

```

-- OPTIONAL denotes network operator specific use.

```

DetachArg ::= SEQUENCE {

```

    legToBeDetached        [0] LegID          OPTIONAL,
    correlationIdentifier   [1] CorrelationID  OPTIONAL
}

```

```

-- OPTIONAL denotes network operator specific use.

```

HoldCallPartyConnectionArg ::= SEQUENCE {
 callID **[0] CallID** **OPTIONAL,**
 legID **[1] LegID**
}

-- *OPTIONAL denotes network operator specific use.*

ReconnectArg ::= SEQUENCE {
 callID **[0] CallID** **OPTIONAL,**
 heldLegID **[1] LegID**
}

-- *OPTIONAL denotes network operator specific use.*

ReleaseCallPartyConnectionArg ::= SEQUENCE {
 legToBeReleased **[0] LegID,**
 callID **[1] CallID** **OPTIONAL,**
 releaseCause **[2] Cause** **OPTIONAL**
}

-- *OPTIONAL denotes network operator specific use. Common Data Types*

CallID ::= INTEGER

-- *Indicates an identifier to reference an instance of a Call accessible to the SCF. Refer to 4.2.2.1/Q.1214 for a description of Call Segment.*

Cause ::= OCTET STRING

-- *Indicates the cause for interface related information. Refer to the Q.763 Cause parameter for encoding.*

LegInformation ::= SEQUENCE {
 legID **[0] LegID,**
 legStatus **[1] LegStatus**
}

-- *Indicates call party information, as defined by a Leg object. This includes a LegID to reference each call party, and a LegStatus to indicate whether the call party is connected or not.*

LegStatus ::= ENUMERATED {
 connected(0),
 unconnected(1),
 pending(2),
 interacting(3) -- *user connected to a resource*
}

-- *Indicates the state of the call party.*

END

Application Protocol (Operation and Error Codes)

IN-CS1-Codes-appendix { ccitt recommendation q 1218 modules(0) cs-1-codes-app(7) version1(0) }

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- *OPERATION AND ERROR CODE DEFINITION*

-- *Code point values are for further study. The operations are grouped by the identified ASEs.*

-- *Call party handling ASE*

addParty **AddParty ::= ffs**
changeParties **ChangeParties ::= ffs**
holdCallPartyConnection **HoldCallPartyConnection ::= ffs**
reconnect **Reconnect ::= ffs**
releaseCallPartyConnection **ReleaseCallPartyConnection ::= ffs**
-- *Attach ASE*
attach **Attach ::= ffs**
detach **Detach ::= ffs**

END

II.5 Procédures

Les principes utilisés sont les suivants:

Les procédures SSF sont illustrées par des machines à états finis ou machines FSM.

La machine FSM (au niveau de l'appel), actuellement décrite dans la présente Recommandation, comporte plusieurs états représentant les différents états de l'appel.

Dans certains de ces états, il existe un autre niveau machine(s) à états finis qu'il ne faut pas nécessairement décrire et qui représente l'état d'une connexion avec l'un des correspondants. Cette machine FSM est appelée "FSM du niveau branche".

Cette machine "FSM du niveau branche" est "créée" ou "disparaît" pendant une transition FSM de l'étape appel. Tout "événement" (réception d'opération par exemple) provoque un changement d'état de toutes les machines FSM concernées.

Les états des branches (Leg states) utilisés sont les valeurs que peut prendre le paramètre "LegStatus", à savoir:

- "IDLE": pas de connexion avec le correspondant;
- "PENDING": correspondant non connecté, en cours de création;
- "UNCONNECTED": correspondant non connecté, état stable;
- "CONNECTED": correspondant connecté à un autre correspondant;
- "INTERACTING": correspondant connecté à une fonction SRF.

Appendice III

ASN.1 étendu

(Cet appendice ne fait pas partie intégrante de la présente Recommandation)

Cet appendice contient la description ASN.1 source étendue de l'interface SCF/SSF/SRF. Pour les définitions des types ASN.1, on se reportera au point 2.

NOTES

- 1 "??" en borne d'intervalle signifie que la valeur est propre au réseau.
- 2 Les opérations sont déposées alphabétiquement.

activateServiceFiltering OPERATION

ARGUMENT

SEQUENCE {

filteredCallTreatment [0] IMPLICIT SEQUENCE {

sFBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),

informationToSend [1] CHOICE {

inbandInfo [0] IMPLICIT SEQUENCE {

messageID [0] CHOICE {

elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),

text [1] IMPLICIT SEQUENCE {

messageContent [0] IMPLICIT IA5String (SIZE (??..??)),

attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},

```

    elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF
      INTEGER (0..2147483647),
    variableMessage [30] IMPLICIT SEQUENCE {
      elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
      variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
        CHOICE {
          integer [0] IMPLICIT INTEGER (0..2147483647),
          number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
          time [2] IMPLICIT OCTET STRING (SIZE (2)),
          date [3] IMPLICIT OCTET STRING (SIZE (3)),
          price [4] IMPLICIT OCTET STRING (SIZE (4))}},
      numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
      duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
      interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL,
      tone [1] IMPLICIT SEQUENCE {
      toneID [0] IMPLICIT INTEGER (0..2147483647),
      duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
      displayInformation [2] IMPLICIT IA5String (SIZE (??..??)) OPTIONAL,
      maximumNumberOfCounters [2] IMPLICIT INTEGER (1..100) OPTIONAL,
      releaseCause [3] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
      filteringCharacteristics [1] CHOICE {
        interval [0] IMPLICIT INTEGER (-1..32000),
        numberOfCalls [1] IMPLICIT INTEGER (0..2147483647)},
      filteringTimeOut [2] CHOICE {
        duration [0] IMPLICIT INTEGER (-2..86400),
        stopTime [1] IMPLICIT OCTET STRING (SIZE (6))},
      filteringCriteria [3] CHOICE {
        dialledNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        callingLineID [1] IMPLICIT OCTET STRING (SIZE (??..??)),
        serviceKey [2] IMPLICIT INTEGER (0..2147483647),
        addressAndService [30] IMPLICIT SEQUENCE {
          calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
          serviceKey [1] IMPLICIT INTEGER (0..2147483647),
          callingAddressValue [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
          locationNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}},
      startTime [4] IMPLICIT OCTET STRING (SIZE (6)) OPTIONAL,
      extensions [5] IMPLICIT SEQUENCE SIZE (1..?) OF
        SEQUENCE {
          type INTEGER,
          criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
          value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

```

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedParameter -- localValue 16}
 ::= localValue 42

```

activityTest OPERATION

```
 ::= localValue 55
```

analysedInformation OPERATION

ARGUMENT

SEQUENCE {

```

  dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
    serviceAddressInformation [0] IMPLICIT SEQUENCE {
      serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
      miscCallInfo [1] IMPLICIT SEQUENCE {
        messageType [0] IMPLICIT ENUMERATED {
          request (0),
          notification (1)},

```

dpAssignment [1] IMPLICIT ENUMERATED {
 individualLine (0),
 groupBased (1),
 officeBased (2) OPTIONAL},
triggerType [2] IMPLICIT ENUMERATED {
 featureActivation (0),
 verticalServiceCode (1),
 customizedAccess (2),
 customizedIntercom (3),
 emergencyService (12),
 aFR (13),
 sharedIOTrunk (14),
 offHookDelay (17),
 channelSetupPRI (18),
 tNoAnswer (25),
 tBusy (26),
 oCalledPartyBusy (27),
 oNoAnswer (29),
 originationAttemptAuthorized (30),
 oAnswer (31),
 oDisconnect (32),
 termAttemptAuthorized (33),
 tAnswer (34),
 tDisconnect (35) OPTIONAL},
bearerCapability [1] CHOICE {
 bearerCap[0] IMPLICIT OCTET STRING (SIZE (2..??)),
 tmr [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
 noCGencountered (0),
 manualCGencountered (1),
 scpOverload (2) OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
 unknown (0),
 dialPulse (1),
 dtmf (2),
 isdn (3),
 isdnNoDtmf (4),
 spare (16) OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
 type INTEGER,
 criticality ENUMERATED {
 ignore (0),
 abort (1) DEFAULT ignore ,
 value [1] ANY DEFINED BY type } OPTIONAL,
 chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
dialledDigits [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyBusinessGroupID [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [3] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [4] CHOICE {
 trunkGroupID [0] IMPLICIT INTEGER,
 privateFacilityID [1] IMPLICIT INTEGER,
 huntGroup [2] IMPLICIT OCTET STRING,
 routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [5] IMPLICIT INTEGER OPTIONAL,
originalCalledPartyID [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
prefix [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectingPartyID [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [9] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [10] IMPLICIT SEQUENCE SIZE (1..3) OF

OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [11] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL,
featureCode [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
accessCode [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
carrier [15] IMPLICIT OCTET STRING OPTIONAL}

ERRORS {

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 3

analyseInformation OPERATION

ARGUMENT

SEQUENCE {

destinationRoutingAddress [0] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..??)),
alertingPattern [1] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
iSDNAccessRelatedInformation [2] IMPLICIT OCTET STRING OPTIONAL,
originalCalledPartyID [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL,
callingPartyNumber [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [6] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
calledPartyNumber [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
chargeNumber [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
carrier [10] IMPLICIT OCTET STRING OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 28

applyCharging OPERATION

ARGUMENT

SEQUENCE {

aChBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),
partyToCharge [2] CHOICE {
sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL}

ERRORS {

-- *missingParameter* -- *localValue* 7,
 -- *unexpectedComponentSequence* -- *localValue* 14,
 -- *unexpectedParameter* -- *localValue* 16,
 -- *unexpectedDataValue* -- *localValue* 15,
 -- *parameterOutOfRange* -- *localValue* 8,
 -- *systemFailure* -- *localValue* 11,
 -- *taskRefused* -- *localValue* 12}
 ::= **localValue** 35

applyChargingReport OPERATION**ARGUMENT**

OCTET STRING (SIZE (??..??))

ERRORS {

-- *missingParameter* -- *localValue* 7,
 -- *unexpectedComponentSequence* -- *localValue* 14,
 -- *unexpectedParameter* -- *localValue* 16,
 -- *unexpectedDataValue* -- *localValue* 15,
 -- *parameterOutOfRange* -- *localValue* 8,
 -- *systemFailure* -- *localValue* 11,
 -- *taskRefused* -- *localValue* 12}
 ::= **localValue** 36

assistRequestInstructions OPERATION**ARGUMENT****SEQUENCE {**

correlationID [0] **IMPLICIT OCTET STRING (SIZE (??..??))**,
iPAvailable [1] **IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL**,
iPSSPCapabilities [2] **IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL**,
extensions [3] **IMPLICIT SEQUENCE SIZE (1..??) OF**

SEQUENCE {

type **INTEGER**,
criticality **ENUMERATED {**
ignore (0),
abort (1) **DEFAULT ignore** ,
value [1] **ANY DEFINED BY type** } **OPTIONAL**}

ERRORS {

-- *missingCustomerRecord* -- *localValue* 6,
 -- *missingParameter* -- *localValue* 7,
 -- *taskRefused* -- *localValue* 12,
 -- *unexpectedComponentSequence* -- *localValue* 14,
 -- *unexpectedDataValue* -- *localValue* 15,
 -- *unexpectedParameter* -- *localValue* 16}
 ::= **localValue** 16

callGap OPERATION**ARGUMENT****SEQUENCE {**

gapCriteria [0] **CHOICE {**
calledAddressValue [0] **IMPLICIT OCTET STRING (SIZE (??..??))**,
gapOnService [2] **IMPLICIT SEQUENCE {**
serviceKey [0] **IMPLICIT INTEGER (0..2147483647)**,
dpCriteria [1] **IMPLICIT ENUMERATED {**
origAttemptAuthorized (1),
collectedInfo (2),
analysedInformation (3),
routeSelectFailure (4),
oCalledPartyBusy (5),
oNoAnswer (6),
oAnswer (7),
oMidCall (8),
oDisconnect (9),
oAbandon (10),
termAttemptAuthorized (12),
tBusy (13),
tNoAnswer (14),


```

tone [1] IMPLICIT SEQUENCE {
    toneID [0] IMPLICIT INTEGER (0..2147483647),
    duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    displayInformation [2] IMPLICIT IA5String (SIZE (??..??)),
    releaseCause [1] IMPLICIT OCTET STRING (SIZE (2..??))} OPTIONAL,
extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}
 ::= localValue 41

```

callInformationReport OPERATION

ARGUMENT

```

SEQUENCE {
    requestedInformationList [0] IMPLICIT SEQUENCE SIZE (1..5) OF
    SEQUENCE {
        requestedInformationType [0] IMPLICIT ENUMERATED {
            callAttemptElapsedTime (0),
            callStopTime (1),
            callConnectedElapsedTime (2),
            calledAddress (3),
            releaseCause (30)},
        requestedInformationValue [1] CHOICE {
            callAttemptElapsedTimeValue [0] IMPLICIT INTEGER (0..255),
            callStopTimeValue [1] IMPLICIT OCTET STRING (SIZE (6)),
            callConnectedElapsedTimeValue [2] IMPLICIT INTEGER (0..2147483647),
            calledAddressValue [3] IMPLICIT OCTET STRING (SIZE (??..??)),
            releaseCauseValue [30] IMPLICIT OCTET STRING (SIZE (2..??))},
    correlationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
        value [1] ANY DEFINED BY type } OPTIONAL}
 ::= localValue 44

```

callInformationRequest OPERATION

ARGUMENT

```

SEQUENCE {
    requestedInformationTypeList [0] IMPLICIT SEQUENCE SIZE (1..5) OF
    ENUMERATED {
        callAttemptElapsedTime (0),
        callStopTime (1),
        callConnectedElapsedTime (2),
        calledAddress (3),
        releaseCause (30)},
    correlationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
        value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

```

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- requestedInfoError -- localValue 10,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 45

```

cancel OPERATION

ARGUMENT

CHOICE {

invokeID [0] IMPLICIT INTEGER (-128..127),
allRequests [1] IMPLICIT NULL}

ERRORS {

-- *cancelFailed* -- *localValue 1,*
-- *missingParameter* -- *localValue 7,*
-- *taskRefused* -- *localValue 12}*

::= *localValue 53*

cancelStatusReportRequest OPERATION

ARGUMENT

SEQUENCE {

resourceID [0] CHOICE {
lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),
facilityGroupID [1] CHOICE {
trunkGroupID [0] IMPLICIT INTEGER,
privateFacilityID [1] IMPLICIT INTEGER,
huntGroup [2] IMPLICIT OCTET STRING,
routeIndex [3] IMPLICIT OCTET STRING},
facilityGroupMemberID [2] IMPLICIT INTEGER,
trunkGroupID [3] IMPLICIT INTEGER} OPTIONAL,
extensions [1] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL}

ERRORS {

-- *cancelFailed* -- *localValue 1,*
-- *missingParameter* -- *localValue 7,*
-- *taskRefused* -- *localValue 12}*

::= *localValue 54*

collectedInformation OPERATION

ARGUMENT

SEQUENCE {

dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
serviceAddressInformation [0] IMPLICIT SEQUENCE {
serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
miscCallInfo [1] IMPLICIT SEQUENCE {
messageType [0] IMPLICIT ENUMERATED {
request (0),
notification (1)},
dpAssignment [1] IMPLICIT ENUMERATED {
individualLine (0),
groupBased (1),
officeBased (2)} OPTIONAL},
triggerType [2] IMPLICIT ENUMERATED {
featureActivation (0),
verticalServiceCode (1),
customizedAccess (2),
customizedIntercom (3),
emergencyService (12),
aFR (13),
sharedIOTrunk (14),
offHookDelay (17),
channelSetupPRI (18),
tNoAnswer (25),
tBusy (26),
oCalledPartyBusy (27),
oNoAnswer (29),
originationAttemptAuthorized (30),
oAnswer (31),
oDisconnect (32),

```

    termAttemptAuthorized (33),
    tAnswer (34),
    tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
    tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAavailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
dialledDigits [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyBusinessGroupID [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [3] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [4] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [5] IMPLICIT INTEGER OPTIONAL,
originalCalledPartyID [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
prefix [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectingPartyID [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [9] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
travellingClassMark [10] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [11] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
featureCode [12] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
accessCode [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
carrier [14] IMPLICIT OCTET STRING OPTIONAL}
ERRORS {
-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 2

```

collectInformation OPERATION

ARGUMENT

SEQUENCE {

alertingPattern [0] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
numberingPlan [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
originalCalledPartyID [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL,
callingPartyNumber [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
dialledDigits [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 27

connect OPERATION

ARGUMENT

SEQUENCE {

destinationRoutingAddress [0] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..??)),
alertingPattern [1] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
correlationID [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
cutAndPaste [3] IMPLICIT INTEGER (0..22) OPTIONAL,
forwardingCondition [4] IMPLICIT ENUMERATED {
busy (0),
noanswer (1),
any (2)} OPTIONAL,
isdNAccessRelatedInformation [5] IMPLICIT OCTET STRING OPTIONAL,
originalCalledPartyID [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
routeList [7] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..??)) OPTIONAL,
scfID [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [10] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL,
carrier [11] IMPLICIT OCTET STRING OPTIONAL,
serviceInteractionIndicators [26] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [27] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [28] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
redirectingPartyID [29] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [30] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 20

connectToResource OPERATION**ARGUMENT****SEQUENCE {**

resourceAddress CHOICE {

ipRoutingAddress [0] IMPLICIT OCTET STRING (SIZE (??..??)),

legID [1] CHOICE {

sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),

receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))},

both [2] IMPLICIT SEQUENCE {

ipRoutingAddress [0] IMPLICIT OCTET STRING (SIZE (??..??)),

legID [1] CHOICE {

sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),

receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))},

none [3] IMPLICIT NULL},

extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF

SEQUENCE {

type INTEGER,

criticality ENUMERATED {

ignore (0),

abort (1)} DEFAULT ignore ,

value [1] ANY DEFINED BY type } OPTIONAL,

serviceInteractionIndicators [30] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}

ERRORS {*-- missingParameter -- localValue 7,**-- systemFailure -- localValue 11,**-- taskRefused -- localValue 12,**-- unexpectedComponentSequence -- localValue 14,**-- unexpectedDataValue -- localValue 15,**-- unexpectedParameter -- localValue 16}***::= localValue 19****continue OPERATION****::= localValue 31****disconnectForwardConnection OPERATION****ERRORS {***-- systemFailure -- localValue 11,**-- taskRefused -- localValue 12,**-- unexpectedComponentSequence -- localValue 14}***::= localValue 18****establishTemporaryConnection OPERATION****ARGUMENT****SEQUENCE {**

assistingSSPIPRoutingAddress [0] IMPLICIT OCTET STRING (SIZE (??..??)),

correlationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,

legID [2] CHOICE {

sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),

receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,

scfID [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,

extensions [4] IMPLICIT SEQUENCE SIZE (1..??) OF

SEQUENCE {

type INTEGER,

criticality ENUMERATED {

ignore (0),

abort (1)} DEFAULT ignore ,

value [1] ANY DEFINED BY type } OPTIONAL,

carrier [5] IMPLICIT OCTET STRING OPTIONAL,

serviceInteractionIndicators [30] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}

ERRORS {*-- eTCFailed -- localValue 3,**-- missingParameter -- localValue 7,**-- systemFailure -- localValue 11,**-- taskRefused -- localValue 12,**-- unexpectedComponentSequence -- localValue 14,**-- unexpectedDataValue -- localValue 15,**-- unexpectedParameter -- localValue 16}***::= localValue 17**

eventNotificationCharging OPERATION

ARGUMENT

SEQUENCE {

eventTypeCharging [0] IMPLICIT OCTET STRING (SIZE (??..??)),
eventSpecificInformationCharging [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
legID [2] CHOICE {
 sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
 receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF
 SEQUENCE {
 type INTEGER,
 criticality ENUMERATED {
 ignore (0),
 abort (1)} DEFAULT ignore ,
 value [1] ANY DEFINED BY type } OPTIONAL,
 monitorMode [30] IMPLICIT ENUMERATED {
 interrupted (0),
 notifyAndContinue (1),
 transparent (2)} DEFAULT notifyAndContinue }

::= localValue 26

eventReportBCSM OPERATION

ARGUMENT

SEQUENCE {

eventTypeBCSM [0] IMPLICIT ENUMERATED {
 origAttemptAuthorized (1),
 collectedInfo (2),
 analysedInformation (3),
 routeSelectFailure (4),
 oCalledPartyBusy (5),
 oNoAnswer (6),
 oAnswer (7),
 oMidCall (8),
 oDisconnect (9),
 oAbandon (10),
 termAttemptAuthorized (12),
 tBusy (13),
 tNoAnswer (14),
 tAnswer (15),
 tMidCall (16),
 tDisconnect (17),
 tAbandon (18)},
bcsmEventCorrelationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
eventSpecificInformationBCSM [2] CHOICE {
 collectedInfoSpecificInfo [0] IMPLICIT SEQUENCE {
 calledPartyNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),
 analyzedInfoSpecificInfo [1] IMPLICIT SEQUENCE {
 calledPartyNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),
 routeSelectFailureSpecificInfo [2] IMPLICIT SEQUENCE {
 failureCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL},
 oCalledPartyBusySpecificInfo [3] IMPLICIT SEQUENCE {
 busyCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL},
 oNoAnswerSpecificInfo [4] IMPLICIT SEQUENCE {},
 oAnswerSpecificInfo [5] IMPLICIT SEQUENCE {},
 oMidCallSpecificInfo [6] IMPLICIT SEQUENCE {
 connectTime [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
 oDisconnectSpecificInfo [7] IMPLICIT SEQUENCE {
 releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
 connectTime [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
 tBusySpecificInfo [8] IMPLICIT SEQUENCE {
 busyCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL},
 tNoAnswerSpecificInfo [9] IMPLICIT SEQUENCE {},
 tAnswerSpecificInfo [10] IMPLICIT SEQUENCE {},
 tMidCallSpecificInfo [11] IMPLICIT SEQUENCE {
 connectTime [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
 tDisconnectSpecificInfo [12] IMPLICIT SEQUENCE {
 releaseCause [0] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
 connectTime [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL}} OPTIONAL,

```

legID [3] CHOICE {
    sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
    receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
miscCallInfo [4] IMPLICIT SEQUENCE {
    messageType [0] IMPLICIT ENUMERATED {
        request (0),
        notification (1)},
    dpAssignment [1] IMPLICIT ENUMERATED {
        individualLine (0),
        groupBased (1),
        officeBased (2)} OPTIONAL} DEFAULT {
    messageType request },
extensions [5] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}
 ::= localValue 24

```

furnishChargingInformation OPERATION

ARGUMENT

OCTET STRING (SIZE (??..??))

ERRORS {

```

-- missingParameter -- localValue 7,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 34

```

holdCallInNetwork OPERATION

ARGUMENT

CHOICE {

```

    holdcause [0] IMPLICIT OCTET STRING,
    empty [1] IMPLICIT NULL}

```

ERRORS {

```

-- missingParameter -- localValue 7,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 21

```

initialDP OPERATION

ARGUMENT

SEQUENCE {

```

    serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
    dialledDigits [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    callingPartyBusinessGroupID [4] IMPLICIT OCTET STRING OPTIONAL,
    callingPartysCategory [5] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
    callingPartySubaddress [6] IMPLICIT OCTET STRING OPTIONAL,
    cGEncountered [7] IMPLICIT ENUMERATED {
        noCGencountered (0),
        manualCGencountered (1),
        scpOverload (2)} OPTIONAL,
    iPSSPCapabilities [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    iPAavailable [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    locationNumber [10] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    miscCallInfo [11] IMPLICIT SEQUENCE {
        messageType [0] IMPLICIT ENUMERATED {
            request (0),
            notification (1)},

```

dpAssignment [1] IMPLICIT ENUMERATED {
 individualLine (0),
 groupBased (1),
 officeBased (2) } OPTIONAL,
originalCalledPartyID [12] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [13] IMPLICIT OCTET STRING OPTIONAL,
 terminalType [14] IMPLICIT ENUMERATED {
 unknown (0),
 dialPulse (1),
 dtmf (2),
 isdn (3),
 isdnNoDtmf (4),
 spare (16) } OPTIONAL,
extensions [15] IMPLICIT SEQUENCE SIZE (1..?) OF
 SEQUENCE {
 type INTEGER,
 criticality ENUMERATED {
 ignore (0),
 abort (1) } DEFAULT ignore ,
 value [1] ANY DEFINED BY type } OPTIONAL,
triggerType [16] IMPLICIT ENUMERATED {
 featureActivation (0),
 verticalServiceCode (1),
 customizedAccess (2),
 customizedIntercom (3),
 emergencyService (12),
 aFR (13),
 sharedIOTrunk (14),
 offHookDelay (17),
 channelSetupPRI (18),
 tNoAnswer (25),
 tBusy (26),
 oCalledPartyBusy (27),
 oNoAnswer (29),
 originationAttemptAuthorized (30),
 oAnswer (31),
 oDisconnect (32),
 termAttemptAuthorized (33),
 tAnswer (34),
 tDisconnect (35) } OPTIONAL,
highLayerCompatibility [23] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
serviceInteractionIndicators [24] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
additionalCallingPartyNumber [25] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
forwardCallIndicators [26] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
bearerCapability [27] CHOICE {
 bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
 tmr [1] IMPLICIT OCTET STRING (SIZE (1)) } OPTIONAL,
eventTypeBCSM [28] IMPLICIT ENUMERATED {
 origAttemptAuthorized (1),
 collectedInfo (2),
 analysedInformation (3),
 routeSelectFailure (4),
 oCalledPartyBusy (5),
 oNoAnswer (6),
 oAnswer (7),
 oMidCall (8),
 oDisconnect (9),
 oAbandon (10),
 termAttemptAuthorized (12),
 tBusy (13),
 tNoAnswer (14),
 tAnswer (15),
 tMidCall (16),
 tDisconnect (17),
 tAbandon (18) } OPTIONAL,
redirectingPartyID [29] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [30] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL }

```

ERRORS {
-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 0

initiateCallAttempt OPERATION
ARGUMENT
SEQUENCE {
  destinationRoutingAddress [0] IMPLICIT SEQUENCE SIZE (1..3) OF
    OCTET STRING (SIZE (??..??)),
  alertingPattern [1] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
  iSDNAccessRelatedInformation [2] IMPLICIT OCTET STRING OPTIONAL,
  travellingClassMark [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  extensions [4] IMPLICIT SEQUENCE SIZE (1..?) OF
    SEQUENCE {
      type INTEGER,
      criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
      value [1] ANY DEFINED BY type } OPTIONAL,
  serviceInteractionIndicators [29] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  callingPartyNumber [30] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}

ERRORS {
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 32

oAnswer OPERATION
ARGUMENT
SEQUENCE {
  dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
    serviceAddressInformation [0] IMPLICIT SEQUENCE {
      serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
      miscCallInfo [1] IMPLICIT SEQUENCE {
        messageType [0] IMPLICIT ENUMERATED {
          request (0),
          notification (1)},
        dpAssignment [1] IMPLICIT ENUMERATED {
          individualLine (0),
          groupBased (1),
          officeBased (2)} OPTIONAL},
        triggerType [2] IMPLICIT ENUMERATED {
          featureActivation (0),
          verticalServiceCode (1),
          customizedAccess (2),
          customizedIntercom (3),
          emergencyService (12),
          aFR (13),
          sharedIOTrunk (14),
          offHookDelay (17),
          channelSetupPRI (18),
          tNoAnswer (25),
          tBusy (26),
          oCalledPartyBusy (27),
          oNoAnswer (29),

```

```

    originationAttemptAuthorized (30),
    oAnswer (31),
    oDisconnect (32),
    termAttemptAuthorized (33),
    tAnswer (34),
    tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..?)),
    tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
callingPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [3] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
originalCalledPartyID [5] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
redirectingPartyID [6] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
redirectionInformation [7] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [8] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..?)) OPTIONAL,
travellingClassMark [9] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
extensions [10] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}
ERRORS {
-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 7

```

oCalledPartyBusy OPERATION

ARGUMENT

SEQUENCE {

dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
 serviceAddressInformation [0] IMPLICIT SEQUENCE {
 serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
 miscCallInfo [1] IMPLICIT SEQUENCE {
 messageType [0] IMPLICIT ENUMERATED {
 request (0),
 notification (1),
 dpAssignment [1] IMPLICIT ENUMERATED {
 individualLine (0),
 groupBased (1),
 officeBased (2)} OPTIONAL},
 triggerType [2] IMPLICIT ENUMERATED {
 featureActivation (0),
 verticalServiceCode (1),
 customizedAccess (2),
 customizedIntercom (3),
 emergencyService (12),
 aFR (13),
 sharedIOTrunk (14),
 offHookDelay (17),
 channelSetupPRI (18),
 tNoAnswer (25),
 tBusy (26),
 oCalledPartyBusy (27),
 oNoAnswer (29),
 originationAttemptAuthorized (30),
 oAnswer (31),
 oDisconnect (32),
 termAttemptAuthorized (33),
 tAnswer (34),
 tDisconnect (35)} OPTIONAL},
 bearerCapability [1] CHOICE {
 bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..?)),
 tmr [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
 calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL,
 callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL,
 callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
 iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL,
 iPAvailable [6] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL,
 iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
 cGEncountered [8] IMPLICIT ENUMERATED {
 noCGencountered (0),
 manualCGencountered (1),
 scpOverload (2)} OPTIONAL,
 locationNumber [9] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL,
 serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
 terminalType [11] IMPLICIT ENUMERATED {
 unknown (0),
 dialPulse (1),
 dtmf (2),
 isdn (3),
 isdnNoDtmf (4),
 spare (16)} OPTIONAL,
 extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
 SEQUENCE {
 type INTEGER,
 criticality ENUMERATED {
 ignore (0),
 abort (1)} DEFAULT ignore ,
 value [1] ANY DEFINED BY type } OPTIONAL,
 chargeNumber [13] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL,
 servingAreaID [14] IMPLICIT OCTET STRING (SIZE (?..?)) OPTIONAL},
 busyCause [1] IMPLICIT OCTET STRING (SIZE (2..?)) OPTIONAL,
 callingPartyBusinessGroupID [2] IMPLICIT OCTET STRING OPTIONAL,
 callingPartySubaddress [3] IMPLICIT OCTET STRING OPTIONAL,
 callingFacilityGroup [4] CHOICE {

```

trunkGroupID [0] IMPLICIT INTEGER,
privateFacilityID [1] IMPLICIT INTEGER,
huntGroup [2] IMPLICIT OCTET STRING,
routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [5] IMPLICIT INTEGER OPTIONAL,
originalCalledPartyID [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
prefix [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectingPartyID [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [9] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [10] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [11] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
type INTEGER,
criticality ENUMERATED {
ignore (0),
abort (1)} DEFAULT ignore ,
value [1] ANY DEFINED BY type } OPTIONAL,
carrier [13] IMPLICIT OCTET STRING OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 5

```

oDisconnect OPERATION

ARGUMENT

```

SEQUENCE {
dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
serviceAddressInformation [0] IMPLICIT SEQUENCE {
serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
miscCallInfo [1] IMPLICIT SEQUENCE {
messageType [0] IMPLICIT ENUMERATED {
request (0),
notification (1)},
dpAssignment [1] IMPLICIT ENUMERATED {
individualLine (0),
groupBased (1),
officeBased (2)} OPTIONAL},
triggerType [2] IMPLICIT ENUMERATED {
featureActivation (0),
verticalServiceCode (1),
customizedAccess (2),
customizedIntercom (3),
emergencyService (12),
aFR (13),
sharedIOTrunk (14),
offHookDelay (17),
channelSetupPRI (18),
tNoAnswer (25),
tBusy (26),
oCalledPartyBusy (27),
oNoAnswer (29),
originationAttemptAuthorized (30),
oAnswer (31),
oDisconnect (32),
termAttemptAuthorized (33),
tAnswer (34),
tDisconnect (35)} OPTIONAL},

```

```

bearerCapability [1] CHOICE {
  bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
  tmr [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
  calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
  iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
  cGEncountered [8] IMPLICIT ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
  locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
  terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
  extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
  SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
      ignore (0),
      abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
  chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
  servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
callingPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [3] CHOICE {
  trunkGroupID [0] IMPLICIT INTEGER,
  privateFacilityID [1] IMPLICIT INTEGER,
  huntGroup [2] IMPLICIT OCTET STRING,
  routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
releaseCause [5] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
routeList [6] IMPLICIT SEQUENCE SIZE (1..3) OF
  OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [7] IMPLICIT SEQUENCE SIZE (1..??) OF
  SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
      ignore (0),
      abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
carrier [8] IMPLICIT OCTET STRING OPTIONAL,
connectTime [9] IMPLICIT INTEGER (0..2147483647) OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 8

```

oMidCall OPERATION

ARGUMENT

SEQUENCE {

```

  dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
    serviceAddressInformation [0] IMPLICIT SEQUENCE {

```

serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
miscCallInfo [1] IMPLICIT SEQUENCE {
 messageType [0] IMPLICIT ENUMERATED {
 request (0),
 notification (1)},
 dpAssignment [1] IMPLICIT ENUMERATED {
 individualLine (0),
 groupBased (1),
 officeBased (2)} OPTIONAL},
 triggerType [2] IMPLICIT ENUMERATED {
 featureActivation (0),
 verticalServiceCode (1),
 customizedAccess (2),
 customizedIntercom (3),
 emergencyService (12),
 aFR (13),
 sharedIOTrunk (14),
 offHookDelay (17),
 channelSetupPRI (18),
 tNoAnswer (25),
 tBusy (26),
 oCalledPartyBusy (27),
 oNoAnswer (29),
 originationAttemptAuthorized (30),
 oAnswer (31),
 oDisconnect (32),
 termAttemptAuthorized (33),
 tAnswer (34),
 tDisconnect (35)} OPTIONAL},
 bearerCapability [1] CHOICE {
 bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
 tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
 calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
 ipSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
 cGEncountered [8] IMPLICIT ENUMERATED {
 noCGencountered (0),
 manualCGencountered (1),
 scpOverload (2)} OPTIONAL,
 locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
 terminalType [11] IMPLICIT ENUMERATED {
 unknown (0),
 dialPulse (1),
 dtmf (2),
 isdn (3),
 isdnNoDtmf (4),
 spare (16)} OPTIONAL,
 extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
 SEQUENCE {
 type INTEGER,
 criticality ENUMERATED {
 ignore (0),
 abort (1)} DEFAULT ignore ,
 value [1] ANY DEFINED BY type } OPTIONAL,
 chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
 servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
calledPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartyBusinessGroupID [3] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [4] IMPLICIT OCTET STRING OPTIONAL,
featureRequestIndicator [5] IMPLICIT ENUMERATED {
 hold (0),
 retrieve (1),
 featureActivation (2),
 spare1 (3),

```

    sparen (127)} OPTIONAL,
    extensions [6] IMPLICIT SEQUENCE SIZE (1..??) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
        value [1] ANY DEFINED BY type } OPTIONAL,
    carrier [7] IMPLICIT OCTET STRING OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 14

```

oNoAnswer OPERATION

ARGUMENT

SEQUENCE {

```

    dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
        serviceAddressInformation [0] IMPLICIT SEQUENCE {
            serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
            miscCallInfo [1] IMPLICIT SEQUENCE {
                messageType [0] IMPLICIT ENUMERATED {
                    request (0),
                    notification (1)},
                dpAssignment [1] IMPLICIT ENUMERATED {
                    individualLine (0),
                    groupBased (1),
                    officeBased (2)} OPTIONAL},
                triggerType [2] IMPLICIT ENUMERATED {
                    featureActivation (0),
                    verticalServiceCode (1),
                    customizedAccess (2),
                    customizedIntercom (3),
                    emergencyService (12),
                    aFR (13),
                    sharedIOTrunk (14),
                    offHookDelay (17),
                    channelSetupPRI (18),
                    tNoAnswer (25),
                    tBusy (26),
                    oCalledPartyBusy (27),
                    oNoAnswer (29),
                    originationAttemptAuthorized (30),
                    oAnswer (31),
                    oDisconnect (32),
                    termAttemptAuthorized (33),
                    tAnswer (34),
                    tDisconnect (35)} OPTIONAL},
                bearerCapability [1] CHOICE {
                    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
                    tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
                calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
                callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
                callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
                iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
                iPAavailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
                iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
                cGEncountered [8] IMPLICIT ENUMERATED {
                    noCGencountered (0),
                    manualCGencountered (1),
                    scpOverload (2)} OPTIONAL,
            }
        }
    }

```

```

locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
callingPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [3] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
originalCalledPartyID [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
prefix [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectingPartyID [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [8] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [9] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [10] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [11] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
carrier [12] IMPLICIT OCTET STRING OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 6

```

originationAttemptAuthorized OPERATION

ARGUMENT

```

SEQUENCE {
    dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
        serviceAddressInformation [0] IMPLICIT SEQUENCE {
            serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
            miscCallInfo [1] IMPLICIT SEQUENCE {
                messageType [0] IMPLICIT ENUMERATED {
                    request (0),
                    notification (1)},
                dpAssignment [1] IMPLICIT ENUMERATED {
                    individualLine (0),
                    groupBased (1),
                    officeBased (2)} OPTIONAL},

```

```

triggerType [2] IMPLICIT ENUMERATED {
    featureActivation (0),
    verticalServiceCode (1),
    customizedAccess (2),
    customizedIntercom (3),
    emergencyService (12),
    aFR (13),
    sharedIOTrunk (14),
    offHookDelay (17),
    channelSetupPRI (18),
    tNoAnswer (25),
    tBusy (26),
    oCalledPartyBusy (27),
    oNoAnswer (29),
    originationAttemptAuthorized (30),
    oAnswer (31),
    oDisconnect (32),
    termAttemptAuthorized (33),
    tAnswer (34),
    tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
    tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
dialledDigits [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyBusinessGroupID [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [3] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [4] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [5] IMPLICIT INTEGER OPTIONAL,
travellingClassMark [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [7] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
carrier [8] IMPLICIT OCTET STRING OPTIONAL}

```

ERRORS {

-- *missingCustomerRecord* -- *localValue* 6,
 -- *missingParameter* -- *localValue* 7,
 -- *parameterOutOfRange* -- *localValue* 8,
 -- *systemFailure* -- *localValue* 11,
 -- *taskRefused* -- *localValue* 12,
 -- *unexpectedComponentSequence* -- *localValue* 14,
 -- *unexpectedDataValue* -- *localValue* 15,
 -- *unexpectedParameter* -- *localValue* 16}

::= *localValue* 1

playAnnouncement OPERATION**ARGUMENT****SEQUENCE {**

informationToSend [0] CHOICE {

inbandInfo [0] IMPLICIT SEQUENCE {

messageID [0] CHOICE {

elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),

text [1] IMPLICIT SEQUENCE {

messageContent [0] IMPLICIT IA5String (SIZE (??..??)),

attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},

elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF

INTEGER (0..2147483647),

variableMessage [30] IMPLICIT SEQUENCE {

elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),

variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF

CHOICE {

integer [0] IMPLICIT INTEGER (0..2147483647),

number [1] IMPLICIT OCTET STRING (SIZE (??..??)),

time [2] IMPLICIT OCTET STRING (SIZE (2)),

date [3] IMPLICIT OCTET STRING (SIZE (3)),

price [4] IMPLICIT OCTET STRING (SIZE (4))}}},

numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,

duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,

interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL},

tone [1] IMPLICIT SEQUENCE {

toneID [0] IMPLICIT INTEGER (0..2147483647),

duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},

displayInformation [2] IMPLICIT IA5String (SIZE (??..??)),

disconnectFromIPForbidden [1] IMPLICIT BOOLEAN DEFAULT TRUE,

requestAnnouncementComplete [2] IMPLICIT BOOLEAN DEFAULT TRUE,

extensions [3] IMPLICIT SEQUENCE SIZE (1..?) OF

SEQUENCE {

type INTEGER,

criticality ENUMERATED {

ignore (0),

abort (1)} DEFAULT ignore ,

value [1] ANY DEFINED BY type } OPTIONAL}

ERRORS {

-- *canceled* -- *localValue* 0,
 -- *missingParameter* -- *localValue* 7,
 -- *parameterOutOfRange* -- *localValue* 8,
 -- *systemFailure* -- *localValue* 11,
 -- *taskRefused* -- *localValue* 12,
 -- *unexpectedComponentSequence* -- *localValue* 14,
 -- *unexpectedDataValue* -- *localValue* 15,
 -- *unexpectedParameter* -- *localValue* 16,
 -- *unavailableResource* -- *localValue* 13}

LINKED {

-- *specializedResourceReport* -- *localValue* 49}

::= *localValue* 47

promptAndCollectUserInformation OPERATION**ARGUMENT****SEQUENCE {**

collectedInfo [0] CHOICE {

```

collectedDigits [0] IMPLICIT SEQUENCE {
  minimumNbOfDigits [0] IMPLICIT INTEGER (1..127) DEFAULT 1,
  maximumNbOfDigits [1] IMPLICIT INTEGER (1..127),
  endOfReplyDigit [2] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
  cancelDigit [3] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
  startDigit [4] IMPLICIT OCTET STRING (SIZE (1..2)) OPTIONAL,
  firstDigitTimeOut [5] IMPLICIT INTEGER (1..127) OPTIONAL,
  interDigitTimeOut [6] IMPLICIT INTEGER (1..127) OPTIONAL,
  errorTreatment [7] IMPLICIT ENUMERATED {
    reportErrorToScf (0),
    help (1),
    repeatPrompt (2)} DEFAULT reportErrorToScf ,
  interruptableAnnInd [8] IMPLICIT BOOLEAN DEFAULT TRUE,
  voiceInformation [9] IMPLICIT BOOLEAN DEFAULT FALSE,
  voiceBack [10] IMPLICIT BOOLEAN DEFAULT FALSE},
iA5Information [1] IMPLICIT BOOLEAN},
disconnectFromIPForbidden [1] IMPLICIT BOOLEAN DEFAULT TRUE,
informationToSend [2] CHOICE {
  inbandInfo [0] IMPLICIT SEQUENCE {
    messageID [0] CHOICE {
      elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
      text [1] IMPLICIT SEQUENCE {
        messageContent [0] IMPLICIT IA5String (SIZE (??..??)),
        attributes [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
      elementaryMessageIDs [29] IMPLICIT SEQUENCE SIZE (1..?) OF
        INTEGER (0..2147483647),
      variableMessage [30] IMPLICIT SEQUENCE {
        elementaryMessageID [0] IMPLICIT INTEGER (0..2147483647),
        variableParts [1] IMPLICIT SEQUENCE SIZE (1..5) OF
          CHOICE {
            integer [0] IMPLICIT INTEGER (0..2147483647),
            number [1] IMPLICIT OCTET STRING (SIZE (??..??)),
            time [2] IMPLICIT OCTET STRING (SIZE (2)),
            date [3] IMPLICIT OCTET STRING (SIZE (3)),
            price [4] IMPLICIT OCTET STRING (SIZE (4))}},
        numberOfRepetitions [1] IMPLICIT INTEGER (1..127) OPTIONAL,
        duration [2] IMPLICIT INTEGER (0..32767) OPTIONAL,
        interval [3] IMPLICIT INTEGER (0..32767) OPTIONAL},
    tone [1] IMPLICIT SEQUENCE {
      toneID [0] IMPLICIT INTEGER (0..2147483647),
      duration [1] IMPLICIT INTEGER (0..2147483647) OPTIONAL},
      displayInformation [2] IMPLICIT IA5String (SIZE (??..??)) OPTIONAL,
    extensions [3] IMPLICIT SEQUENCE SIZE (1..?) OF
      SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
          ignore (0),
          abort (1)} DEFAULT ignore ,
        value [1] ANY DEFINED BY type } OPTIONAL}
}

```

RESULT

```

CHOICE {
  digitsResponse [0] IMPLICIT OCTET STRING (SIZE (??..??)),
  iA5Response [1] IMPLICIT IA5String }

```

ERRORS {

```

-- canceled -- localValue 0,
-- improperCallerResponse -- localValue 4,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unavailableResource -- localValue 13,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 48

```

releaseCall OPERATION

ARGUMENT

OCTET STRING (SIZE (2..??))

::= localValue 22

requestCurrentStatusReport OPERATION

ARGUMENT

CHOICE {

lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),

facilityGroupID [1] CHOICE {

trunkGroupID [0] IMPLICIT INTEGER,

privateFacilityID [1] IMPLICIT INTEGER,

huntGroup [2] IMPLICIT OCTET STRING,

routeIndex [3] IMPLICIT OCTET STRING},

facilityGroupMemberID [2] IMPLICIT INTEGER,

trunkGroupID [3] IMPLICIT INTEGER}

RESULT

SEQUENCE {

resourceStatus [0] IMPLICIT ENUMERATED {

busy (0),

idle (1)},

resourceID [1] CHOICE {

lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),

facilityGroupID [1] CHOICE {

trunkGroupID [0] IMPLICIT INTEGER,

privateFacilityID [1] IMPLICIT INTEGER,

huntGroup [2] IMPLICIT OCTET STRING,

routeIndex [3] IMPLICIT OCTET STRING},

facilityGroupMemberID [2] IMPLICIT INTEGER,

trunkGroupID [3] IMPLICIT INTEGER} OPTIONAL,

extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF

SEQUENCE {

type INTEGER,

criticality ENUMERATED {

ignore (0),

abort (1) DEFAULT ignore ,

value [1] ANY DEFINED BY type } OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,

-- parameterOutOfRange -- localValue 8,

-- systemFailure -- localValue 11,

-- taskRefused -- localValue 12,

-- unexpectedComponentSequence -- localValue 14,

-- unexpectedParameter -- localValue 16,

-- unknownResource -- localValue 18}

::= localValue 37

requestEveryStatusChangeReport OPERATION

ARGUMENT

SEQUENCE {

resourceID [0] CHOICE {

lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),

facilityGroupID [1] CHOICE {

trunkGroupID [0] IMPLICIT INTEGER,

privateFacilityID [1] IMPLICIT INTEGER,

huntGroup [2] IMPLICIT OCTET STRING,

routeIndex [3] IMPLICIT OCTET STRING},

facilityGroupMemberID [2] IMPLICIT INTEGER,

trunkGroupID [3] IMPLICIT INTEGER},

correlationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,

monitorDuration [2] IMPLICIT INTEGER (-2..86400) OPTIONAL,

extensions [3] IMPLICIT SEQUENCE SIZE (1..??) OF

SEQUENCE {

type INTEGER,

criticality ENUMERATED {

ignore (0),

abort (1) DEFAULT ignore ,

value [1] ANY DEFINED BY type } OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedParameter -- localValue 16,
-- unknownResource -- localValue 18}
::= localValue 38

requestFirstStatusMatchReport OPERATION**ARGUMENT****SEQUENCE {**

resourceID [0] CHOICE {
 lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),
 facilityGroupID [1] CHOICE {
 trunkGroupID [0] IMPLICIT INTEGER,
 privateFacilityID [1] IMPLICIT INTEGER,
 huntGroup [2] IMPLICIT OCTET STRING,
 routeIndex [3] IMPLICIT OCTET STRING},
 facilityGroupMemberID [2] IMPLICIT INTEGER,
 trunkGroupID [3] IMPLICIT INTEGER} OPTIONAL,
resourceStatus [1] IMPLICIT ENUMERATED {
 busy (0),
 idle (1)} OPTIONAL,
correlationID [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
monitorDuration [3] IMPLICIT INTEGER (-2..86400) OPTIONAL,
extensions [4] IMPLICIT SEQUENCE SIZE (1..?) OF
 SEQUENCE {
 type INTEGER,
 criticality ENUMERATED {
 ignore (0),
 abort (1)} DEFAULT ignore ,
 value [1] ANY DEFINED BY type } OPTIONAL,
 bearerCapability [5] CHOICE {
 bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
 tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedParameter -- localValue 16,
-- unknownResource -- localValue 18}
::= localValue 39

requestNotificationChargingEvent OPERATION**ARGUMENT****SEQUENCE SIZE (1..?) OF****SEQUENCE {**

eventTypeCharging [0] IMPLICIT OCTET STRING (SIZE (??..??)),
monitorMode [1] IMPLICIT ENUMERATED {
 interrupted (0),
 notifyAndContinue (1),
 transparent (2)},
legID [2] CHOICE {
 sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
 receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL}

ERRORS {

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,

```

-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 25

requestReportBCSMEvent OPERATION
  ARGUMENT
    SEQUENCE {
      bcsmEvents [0] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
          eventTypeBCSM [0] IMPLICIT ENUMERATED {
            origAttemptAuthorized (1),
            collectedInfo (2),
            analysedInformation (3),
            routeSelectFailure (4),
            oCalledPartyBusy (5),
            oNoAnswer (6),
            oAnswer (7),
            oMidCall (8),
            oDisconnect (9),
            oAbandon (10),
            termAttemptAuthorized (12),
            tBusy (13),
            tNoAnswer (14),
            tAnswer (15),
            tMidCall (16),
            tDisconnect (17),
            tAbandon (18)},
          monitorMode [1] IMPLICIT ENUMERATED {
            interrupted (0),
            notifyAndContinue (1),
            transparent (2)},
          legID [2] CHOICE {
            sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
            receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
          dpSpecificCriteria [30] CHOICE {
            numberOfDigits [0] IMPLICIT INTEGER (1..255),
            applicationTimer [1] IMPLICIT INTEGER (0..2047)} OPTIONAL},
      bcsmEventCorrelationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
      extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
          type INTEGER,
          criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
          value [1] ANY DEFINED BY type } OPTIONAL}

```

```

ERRORS {
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 23

```

```

resetTimer OPERATION
  ARGUMENT
    SEQUENCE {
      timerID [0] IMPLICIT ENUMERATED {
        tssf (0)} DEFAULT tssf ,
      timervalue [1] IMPLICIT INTEGER (0..2147483647),
      extensions [2] IMPLICIT SEQUENCE SIZE (1..??) OF
        SEQUENCE {
          type INTEGER,
          criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
          value [1] ANY DEFINED BY type } OPTIONAL}

```

```

ERRORS {
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 33

routeSelectFailure OPERATION
  ARGUMENT
    SEQUENCE {
      dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
        serviceAddressInformation [0] IMPLICIT SEQUENCE {
          serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
          miscCallInfo [1] IMPLICIT SEQUENCE {
            messageType [0] IMPLICIT ENUMERATED {
              request (0),
              notification (1),
            }
            dpAssignment [1] IMPLICIT ENUMERATED {
              individualLine (0),
              groupBased (1),
              officeBased (2)} OPTIONAL},
            triggerType [2] IMPLICIT ENUMERATED {
              featureActivation (0),
              verticalServiceCode (1),
              customizedAccess (2),
              customizedIntercom (3),
              emergencyService (12),
              aFR (13),
              sharedIOTrunk (14),
              offHookDelay (17),
              channelSetupPRI (18),
              tNoAnswer (25),
              tBusy (26),
              oCalledPartyBusy (27),
              oNoAnswer (29),
              originationAttemptAuthorized (30),
              oAnswer (31),
              oDisconnect (32),
              termAttemptAuthorized (33),
              tAnswer (34),
              tDisconnect (35)} OPTIONAL},
            bearerCapability [1] CHOICE {
              bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
              tnr [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
            }
            calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
            callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
            callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
            iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
            iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
            iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
            cGEncountered [8] IMPLICIT ENUMERATED {
              noCGencountered (0),
              manualCGencountered (1),
              scpOverload (2)} OPTIONAL,
            locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
            serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
            terminalType [11] IMPLICIT ENUMERATED {
              unknown (0),
              dialPulse (1),
              dtmf (2),
              isdn (3),
              isdnNoDtmf (4),
              spare (16)} OPTIONAL,
            extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
              SEQUENCE {
                type INTEGER,
                criticality ENUMERATED {

```

```

        ignore (0),
        abort (1)} DEFAULT ignore ,
        value [1] ANY DEFINED BY type } OPTIONAL,
        chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
dialledDigits [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyBusinessGroupID [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [3] IMPLICIT OCTET STRING OPTIONAL,
callingFacilityGroup [4] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
callingFacilityGroupMember [5] IMPLICIT INTEGER OPTIONAL,
failureCause [6] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
originalCalledPartyID [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
prefix [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectingPartyID [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [10] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [11] IMPLICIT SEQUENCE SIZE (1..3) OF
    OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [12] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [13] IMPLICIT SEQUENCE SIZE (1..?) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore (0),
            abort (1)} DEFAULT ignore ,
        value [1] ANY DEFINED BY type } OPTIONAL,
carrier [14] IMPLICIT OCTET STRING OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 4

```

selectFacility OPERATION

ARGUMENT

SEQUENCE {

```

    alertingPattern [0] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
    destinationNumberRoutingAddress [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    iSDNAccessRelatedInformation [2] IMPLICIT OCTET STRING OPTIONAL,
    calledFacilityGroup [3] CHOICE {
        trunkGroupID [0] IMPLICIT INTEGER,
        privateFacilityID [1] IMPLICIT INTEGER,
        huntGroup [2] IMPLICIT OCTET STRING,
        routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
    calledFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
    originalCalledPartyID [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    extensions [6] IMPLICIT SEQUENCE SIZE (1..?) OF
        SEQUENCE {
            type INTEGER,
            criticality ENUMERATED {
                ignore (0),
                abort (1)} DEFAULT ignore ,
            value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

```

-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,

```

```
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 30
```

selectRoute OPERATION

ARGUMENT

SEQUENCE {

```
destinationRoutingAddress [0] IMPLICIT SEQUENCE SIZE (1..3) OF
  OCTET STRING (SIZE (??..??)),
alertingPattern [1] IMPLICIT OCTET STRING (SIZE (3)) OPTIONAL,
correlationID [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [3] IMPLICIT OCTET STRING OPTIONAL,
originalCalledPartyID [4] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
routeList [5] IMPLICIT SEQUENCE SIZE (1..3) OF
  OCTET STRING (SIZE (??..??)) OPTIONAL,
scfID [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [7] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
extensions [8] IMPLICIT SEQUENCE SIZE (1..?) OF
  SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
      ignore (0),
      abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
  carrier [9] IMPLICIT OCTET STRING OPTIONAL}
```

ERRORS {

```
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 29
```

sendChargingInformation OPERATION

ARGUMENT

SEQUENCE {

```
sCIBillingChargingCharacteristics [0] IMPLICIT OCTET STRING (SIZE (??..??)),
partyToCharge [1] CHOICE {
  sendingSideID [0] IMPLICIT OCTET STRING (SIZE (1)),
  receivingSideID [1] IMPLICIT OCTET STRING (SIZE (1))},
extensions [2] IMPLICIT SEQUENCE SIZE (1..?) OF
  SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
      ignore (0),
      abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}
```

ERRORS {

```
-- missingParameter -- localValue 7,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedParameter -- localValue 16,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unknownLegID -- localValue 17}
::= localValue 46
```

serviceFilteringResponse OPERATION

ARGUMENT

SEQUENCE {

```
countersValue [0] IMPLICIT SEQUENCE SIZE (0..100) OF
  SEQUENCE {
    counterID [0] IMPLICIT INTEGER (0..99),
    counterValue [1] IMPLICIT INTEGER (0..2147483647)},
```

```

filteringCriteria [1] CHOICE {
    dialledNumber [0] IMPLICIT OCTET STRING (SIZE (??..??)),
    callingLineID [1] IMPLICIT OCTET STRING (SIZE (??..??)),
    serviceKey [2] IMPLICIT INTEGER (0..2147483647),
    addressAndService [30] IMPLICIT SEQUENCE {
        calledAddressValue [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        serviceKey [1] IMPLICIT INTEGER (0..2147483647),
        callingAddressValue [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        locationNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL}},
extensions [2] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore    (0),
        abort     (1)} DEFAULT ignore ,
    value        [1] ANY DEFINED BY type } OPTIONAL,
responseCondition [3] IMPLICIT ENUMERATED {
    intermediateResponse (0),
    lastResponse (1)} OPTIONAL}

```

::= localValue 43

specializedResourceReport OPERATION

ARGUMENT

NULL

::= localValue 49

statusReport OPERATION

ARGUMENT

SEQUENCE {

```

    resourceStatus [0] IMPLICIT ENUMERATED {
        busy (0),
        idle (1)} OPTIONAL,
    correlationID [1] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
    resourceID [2] CHOICE {
        lineID [0] IMPLICIT OCTET STRING (SIZE (??..??)),
        facilityGroupID [1] CHOICE {
            trunkGroupID [0] IMPLICIT INTEGER,
            privateFacilityID [1] IMPLICIT INTEGER,
            huntGroup [2] IMPLICIT OCTET STRING,
            routeIndex [3] IMPLICIT OCTET STRING},
        facilityGroupMemberID [2] IMPLICIT INTEGER,
        trunkGroupID [3] IMPLICIT INTEGER} OPTIONAL,
    extensions [3] IMPLICIT SEQUENCE SIZE (1..?) OF
    SEQUENCE {
        type INTEGER,
        criticality ENUMERATED {
            ignore    (0),
            abort     (1)} DEFAULT ignore ,
        value        [1] ANY DEFINED BY type } OPTIONAL,
    reportCondition [4] IMPLICIT ENUMERATED {
        statusReport (0),
        timerExpired (1),
        canceled (2)} OPTIONAL}

```

::= localValue 40

tAnswer OPERATION

ARGUMENT

SEQUENCE {

```

    dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
        serviceAddressInformation [0] IMPLICIT SEQUENCE {
            serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
            miscCallInfo [1] IMPLICIT SEQUENCE {
                messageType [0] IMPLICIT ENUMERATED {
                    request (0),
                    notification (1)},
                dpAssignment [1] IMPLICIT ENUMERATED {
                    individualLine (0),
                    groupBased (1),
                    officeBased (2)} OPTIONAL},

```

```

triggerType [2] IMPLICIT ENUMERATED {
    featureActivation (0),
    verticalServiceCode (1),
    customizedAccess (2),
    customizedIntercom (3),
    emergencyService (12),
    aFR (13),
    sharedIOTrunk (14),
    offHookDelay (17),
    channelSetupPRI (18),
    tNoAnswer (25),
    tBusy (26),
    oCalledPartyBusy (27),
    oNoAnswer (29),
    originationAttemptAuthorized (30),
    oAnswer (31),
    oDisconnect (32),
    termAttemptAuthorized (33),
    tAnswer (34),
    tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
    tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
calledPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
calledFacilityGroup [3] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
calledFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
extensions [5] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 12

tBusy OPERATION**ARGUMENT****SEQUENCE {**

dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
 serviceAddressInformation [0] IMPLICIT SEQUENCE {
 serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
 miscCallInfo [1] IMPLICIT SEQUENCE {
 messageType [0] IMPLICIT ENUMERATED {
 request (0),
 notification (1)},
 dpAssignment [1] IMPLICIT ENUMERATED {
 individualLine (0),
 groupBased (1),
 officeBased (2)} OPTIONAL},
 triggerType [2] IMPLICIT ENUMERATED {
 featureActivation (0),
 verticalServiceCode (1),
 customizedAccess (2),
 customizedIntercom (3),
 emergencyService (12),
 aFR (13),
 sharedIOTrunk (14),
 offHookDelay (17),
 channelSetupPRI (18),
 tNoAnswer (25),
 tBusy (26),
 oCalledPartyBusy (27),
 oNoAnswer (29),
 originationAttemptAuthorized (30),
 oAnswer (31),
 oDisconnect (32),
 termAttemptAuthorized (33),
 tAnswer (34),
 tDisconnect (35)} OPTIONAL},
 bearerCapability [1] CHOICE {
 bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..?)),
 tmr [1] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
 calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
 callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
 callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
 iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
 iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
 iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
 cGEncountered [8] IMPLICIT ENUMERATED {
 noCGencountered (0),
 manualCGencountered (1),
 scpOverload (2)} OPTIONAL,
 locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
 serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
 terminalType [11] IMPLICIT ENUMERATED {
 unknown (0),
 dialPulse (1),
 dtmf (2),
 isdn (3),
 isdnNoDtmf (4),
 spare (16)} OPTIONAL,

```

extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (?.?.?)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (?.?.?)) OPTIONAL},
busyCause [1] IMPLICIT OCTET STRING (SIZE (2..?)) OPTIONAL,
calledPartyBusinessGroupID [2] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [3] IMPLICIT OCTET STRING OPTIONAL,
originalCalledPartyID [4] IMPLICIT OCTET STRING (SIZE (?.?.?)) OPTIONAL,
redirectingPartyID [5] IMPLICIT OCTET STRING (SIZE (?.?.?)) OPTIONAL,
redirectionInformation [6] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [7] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (?.?.?)) OPTIONAL,
travellingClassMark [8] IMPLICIT OCTET STRING (SIZE (?.?.?)) OPTIONAL,
extensions [9] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 10

```

tDisconnect OPERATION

ARGUMENT

```

SEQUENCE {
    dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
        serviceAddressInformation [0] IMPLICIT SEQUENCE {
            serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
            miscCallInfo [1] IMPLICIT SEQUENCE {
                messageType [0] IMPLICIT ENUMERATED {
                    request (0),
                    notification (1)},
                dpAssignment [1] IMPLICIT ENUMERATED {
                    individualLine (0),
                    groupBased (1),
                    officeBased (2)} OPTIONAL},
            triggerType [2] IMPLICIT ENUMERATED {
                featureActivation (0),
                verticalServiceCode (1),
                customizedAccess (2),
                customizedIntercom (3),
                emergencyService (12),
                aFR (13),
                sharedIOTrunk (14),
                offHookDelay (17),
                channelSetupPRI (18),
                tNoAnswer (25),
                tBusy (26),
                oCalledPartyBusy (27),
                oNoAnswer (29),
                originationAttemptAuthorized (30),
                oAnswer (31),
                oDisconnect (32),
                termAttemptAuthorized (33),

```

```

    tAnswer (34),
    tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
    bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
    tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
    noCGencountered (0),
    manualCGencountered (1),
    scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
calledPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
calledFacilityGroup [3] CHOICE {
    trunkGroupID [0] IMPLICIT INTEGER,
    privateFacilityID [1] IMPLICIT INTEGER,
    huntGroup [2] IMPLICIT OCTET STRING,
    routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
calledFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
releaseCause [5] IMPLICIT OCTET STRING (SIZE (2..??)) OPTIONAL,
extensions [6] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
connectTime [7] IMPLICIT INTEGER (0..2147483647) OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
::= localValue 13

```

termAttemptAuthorized OPERATION

ARGUMENT

```

SEQUENCE {
    dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
        serviceAddressInformation [0] IMPLICIT SEQUENCE {
            serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,

```

```

miscCallInfo [1] IMPLICIT SEQUENCE {
  messageType [0] IMPLICIT ENUMERATED {
    request (0),
    notification (1)},
  dpAssignment [1] IMPLICIT ENUMERATED {
    individualLine (0),
    groupBased (1),
    officeBased (2)} OPTIONAL},
triggerType [2] IMPLICIT ENUMERATED {
  featureActivation (0),
  verticalServiceCode (1),
  customizedAccess (2),
  customizedIntercom (3),
  emergencyService (12),
  aFR (13),
  sharedIOTrunk (14),
  offHookDelay (17),
  channelSetupPRI (18),
  tNoAnswer (25),
  tBusy (26),
  oCalledPartyBusy (27),
  oNoAnswer (29),
  originationAttemptAuthorized (30),
  oAnswer (31),
  oDisconnect (32),
  termAttemptAuthorized (33),
  tAnswer (34),
  tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
  bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
  tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
  noCGencountered (0),
  manualCGencountered (1),
  scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
  unknown (0),
  dialPulse (1),
  dtmf (2),
  isdn (3),
  isdnNoDtmf (4),
  spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..??) OF
SEQUENCE {
  type INTEGER,
  criticality ENUMERATED {
    ignore (0),
    abort (1)} DEFAULT ignore ,
  value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
calledPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartyBusinessGroupID [3] IMPLICIT OCTET STRING OPTIONAL,
originalCalledPartyID [4] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectingPartyID [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
redirectionInformation [6] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
routeList [7] IMPLICIT SEQUENCE SIZE (1..3) OF
OCTET STRING (SIZE (??..??)) OPTIONAL,
travellingClassMark [8] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL
extensions [9] IMPLICIT SEQUENCE SIZE (1..??) OF

```

```

SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 9

```

tMidCall OPERATION

ARGUMENT

SEQUENCE {

```

dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
    serviceAddressInformation [0] IMPLICIT SEQUENCE {
        serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
        miscCallInfo [1] IMPLICIT SEQUENCE {
            messageType [0] IMPLICIT ENUMERATED {
                request (0),
                notification (1)},
            dpAssignment [1] IMPLICIT ENUMERATED {
                individualLine (0),
                groupBased (1),
                officeBased (2)} OPTIONAL},
            triggerType [2] IMPLICIT ENUMERATED {
                featureActivation (0),
                verticalServiceCode (1),
                customizedAccess (2),
                customizedIntercom (3),
                emergencyService (12),
                aFR (13),
                sharedIOTrunk (14),
                offHookDelay (17),
                channelSetupPRI (18),
                tNoAnswer (25),
                tBusy (26),
                oCalledPartyBusy (27),
                oNoAnswer (29),
                originationAttemptAuthorized (30),
                oAnswer (31),
                oDisconnect (32),
                termAttemptAuthorized (33),
                tAnswer (34),
                tDisconnect (35)} OPTIONAL},
        bearerCapability [1] CHOICE {
            bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..??)),
            tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
        calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
        iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
        cGEncountered [8] IMPLICIT ENUMERATED {
            noCGencountered (0),
            manualCGencountered (1),
            scpOverload (2)} OPTIONAL,
        locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
        serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,

```

```

terminalType [11] IMPLICIT ENUMERATED {
    unknown (0),
    dialPulse (1),
    dtmf (2),
    isdn (3),
    isdnNoDtmf (4),
    spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..??)) OPTIONAL},
calledPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
callingPartyBusinessGroupID [3] IMPLICIT OCTET STRING OPTIONAL,
callingPartySubaddress [4] IMPLICIT OCTET STRING OPTIONAL,
featureRequestIndicator [5] IMPLICIT ENUMERATED {
    hold (0),
    retrieve (1),
    featureActivation (2),
    spare1 (3),
    sparen (127)} OPTIONAL,
extensions [6] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
    type INTEGER,
    criticality ENUMERATED {
        ignore (0),
        abort (1)} DEFAULT ignore ,
    value [1] ANY DEFINED BY type } OPTIONAL,
carrier [7] IMPLICIT OCTET STRING OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,
-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 15

```

tNoAnswer OPERATION

ARGUMENT

SEQUENCE {

```

dpSpecificCommonParameters [0] IMPLICIT SEQUENCE {
    serviceAddressInformation [0] IMPLICIT SEQUENCE {
        serviceKey [0] IMPLICIT INTEGER (0..2147483647) OPTIONAL,
        miscCallInfo [1] IMPLICIT SEQUENCE {
            messageType [0] IMPLICIT ENUMERATED {
                request (0),
                notification (1)},
            dpAssignment [1] IMPLICIT ENUMERATED {
                individualLine (0),
                groupBased (1),
                officeBased (2)} OPTIONAL},
        triggerType [2] IMPLICIT ENUMERATED {
            featureActivation (0),
            verticalServiceCode (1),
            customizedAccess (2),
            customizedIntercom (3),
            emergencyService (12),
            aFR (13),
            sharedIOTrunk (14),
            offHookDelay (17),

```

```

channelSetupPRI (18),
tNoAnswer (25),
tBusy (26),
oCalledPartyBusy (27),
oNoAnswer (29),
originationAttemptAuthorized (30),
oAnswer (31),
oDisconnect (32),
termAttemptAuthorized (33),
tAnswer (34),
tDisconnect (35)} OPTIONAL},
bearerCapability [1] CHOICE {
  bearerCap [0] IMPLICIT OCTET STRING (SIZE (2..?)),
  tmr [1] IMPLICIT OCTET STRING (SIZE (1))} OPTIONAL,
calledPartyNumber [2] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
callingPartyNumber [3] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
callingPartysCategory [4] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
iPSSPCapabilities [5] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
iPAvailable [6] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
iSDNAccessRelatedInformation [7] IMPLICIT OCTET STRING OPTIONAL,
cGEncountered [8] IMPLICIT ENUMERATED {
  noCGencountered (0),
  manualCGencountered (1),
  scpOverload (2)} OPTIONAL,
locationNumber [9] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
serviceProfileIdentifier [10] IMPLICIT OCTET STRING OPTIONAL,
terminalType [11] IMPLICIT ENUMERATED {
  unknown (0),
  dialPulse (1),
  dtmf (2),
  isdn (3),
  isdnNoDtmf (4),
  spare (16)} OPTIONAL,
extensions [12] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
  type INTEGER,
  criticality ENUMERATED {
    ignore (0),
    abort (1)} DEFAULT ignore ,
  value [1] ANY DEFINED BY type } OPTIONAL,
chargeNumber [13] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
servingAreaID [14] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL},
calledPartyBusinessGroupID [1] IMPLICIT OCTET STRING OPTIONAL,
calledPartySubaddress [2] IMPLICIT OCTET STRING OPTIONAL,
calledFacilityGroup [3] CHOICE {
  trunkGroupID [0] IMPLICIT INTEGER,
  privateFacilityID [1] IMPLICIT INTEGER,
  huntGroup [2] IMPLICIT OCTET STRING,
  routeIndex [3] IMPLICIT OCTET STRING} OPTIONAL,
calledFacilityGroupMember [4] IMPLICIT INTEGER OPTIONAL,
originalCalledPartyID [5] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
redirectingPartyID [6] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
redirectionInformation [7] IMPLICIT OCTET STRING (SIZE (2)) OPTIONAL,
travellingClassMark [8] IMPLICIT OCTET STRING (SIZE (??..?)) OPTIONAL,
extensions [9] IMPLICIT SEQUENCE SIZE (1..?) OF
SEQUENCE {
  type INTEGER,
  criticality ENUMERATED {
    ignore (0),
    abort (1)} DEFAULT ignore ,
  value [1] ANY DEFINED BY type } OPTIONAL}

```

ERRORS {

```

-- missingCustomerRecord -- localValue 6,
-- missingParameter -- localValue 7,
-- parameterOutOfRange -- localValue 8,
-- systemFailure -- localValue 11,
-- taskRefused -- localValue 12,
-- unexpectedComponentSequence -- localValue 14,

```

```

-- unexpectedDataValue -- localValue 15,
-- unexpectedParameter -- localValue 16}
 ::= localValue 11

canceled ERROR
 ::= localValue 0

cancelFailed ERROR
  PARAMETER
    SEQUENCE {
      problem [0] IMPLICIT ENUMERATED {
        unknownOperation (0),
        tooLate (1),
        operationNotCancellable (2)},
      operation [1] IMPLICIT INTEGER (-128..127)}
 ::= localValue 1

eTCFailed ERROR
 ::= localValue 3

improperCallerResponse ERROR
 ::= localValue 4

missingCustomerRecord ERROR
 ::= localValue 6

missingParameter ERROR
 ::= localValue 7

parameterOutOfRange ERROR
 ::= localValue 8

requestedInfoError ERROR
  PARAMETER
    ENUMERATED {
      unknownRequestedInfo (1),
      requestedInfoNotAvailable (2)}
 ::= localValue 10

systemFailure ERROR
  PARAMETER
    unavailableNetworkResource ENUMERATED {
      unavailableResources (0),
      componentFailure (1),
      basicCallProcessingException (2),
      resourceStatusFailure (3),
      endUserFailure (4)}
 ::= localValue 11

taskRefused ERROR
  PARAMETER
    ENUMERATED {
      generic (0),
      unobtainable (1),
      congestion (2)}
 ::= localValue 12

unavailableResource ERROR
 ::= localValue 13

unexpectedComponentSequence ERROR
 ::= localValue 14

unexpectedDataValue ERROR
 ::= localValue 15

unexpectedParameter ERROR
 ::= localValue 16

unknownLegID ERROR
 ::= localValue 17

unknownResource ERROR
 ::= localValue 18

```