



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Q.1218

(03/93)

**RECOMMANDATIONS GÉNÉRALES SUR
LA COMMUTATION ET LA SIGNALISATION
TÉLÉPHONIQUES
RÉSEAU INTELLIGENT**

**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 que les Commissions d'études de l'UIT-T doivent examiner et à propos desquels elles doivent émettre des Recommandations.

La Recommandation UIT-T Q.1218, élaborée par la Commission d'études XI (1988-1993) de l'UIT-T, a été approuvée par la CMNT (Helsinki, 1-12 mars 1993).

NOTES

1 Suite au processus de réforme entrepris au sein de l'Union internationale des télécommunications (UIT), le CCITT n'existe plus depuis le 28 février 1993. Il est remplacé par le Secteur de la normalisation des télécommunications de l'UIT (UIT-T) créé le 1^{er} mars 1993. De même, le CCIR et l'IFRB ont été remplacés par le Secteur des radiocommunications.

Afin de ne pas retarder la publication de la présente Recommandation, aucun changement n'a été apporté aux mentions contenant les sigles CCITT, CCIR et IFRB ou aux entités qui leur sont associées, comme «Assemblée plénière», «Secrétariat», etc. Les futures éditions de la présente Recommandation adopteront la terminologie appropriée reflétant la nouvelle structure de l'UIT.

2 Dans la présente Recommandation, le terme «Administration» désigne indifféremment une administration de télécommunication ou une exploitation reconnue.

© UIT 1993

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 | Méthodologie de définition..... | 1 |
| 0.2 | Exemples de scénarios physiques | 1 |
| 0.3 | Architecture du protocole d'application du réseau intelligent ou protocole INAP | 8 |
| 0.4 | Adressage dans le protocole INAP | 9 |
| 0.5 | Relations entre les Recommandations Q.1214 et la présente Recommandation | 10 |
| 1 | Règles applicables aux entités SACF/MACF..... | 12 |
| 1.1 | Réflexion du contexte d'application du sous-système TCAP | 12 |
| 1.2 | Exécution séquentielle/parallèle des opérations | 12 |
| 2 | Syntaxe abstraite du protocole d'application dans l'ensemble de capacités CS-1 du réseau intelligent | 12 |
| 2.1 | IN CS-1 Operation Types (Types d'opérations dans l'ensemble CS-1 du RI) | 13 |
| 2.2 | IN CS-1 Error Types (Types d'erreurs dans l'ensemble CS-1 du RI) | 28 |
| 2.3 | IN CS-1 data types (Types de données dans l'ensemble CS-1 du RI) | 30 |
| 2.4 | IN CS-1 application protocol (operation and error codes) [Protocole d'application de l'ensemble CS-1 du RI (codes d'opérations et d'erreurs)] | 46 |
| 3 | Procédures..... | 50 |
| 3.1 | Définition des procédures et des entités | 50 |
| Appendice I – 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 | | 102 |
| I.1 | Généralités | 102 |
| I.2 | Opérations..... | 103 |
| I.3 | Paramètres..... | 104 |
| I.4 | Modules en notation ASN.1 des opérations et des paramètres | 106 |
| I.5 | Procédures | 110 |

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.

Certaines redondances fonctionnelles peuvent exister dans l'ensemble des opérations relatives au traitement des appels et gêner alors l'interfonctionnement. Les Administrations désireuses d'utiliser le réseau intelligent et les fabricants de matériel pour ce type de réseau doivent tenir compte de cette éventualité.

Il est certes prévu d'apporter à la présente Recommandation un certain nombre d'améliorations afin de réparer les erreurs et les omissions constatées au fur et à mesure des expériences de réalisation et d'utilisation de l'ensemble CS-1. Toutefois, même si ces corrections sont importantes, elles n'auront vraisemblablement qu'une incidence mineure sur le texte actuel.

L'Appendice I contient des précisions sur le sens vers lequel les futurs ensembles de capacités pourront évoluer. Ce domaine n'a pas été suffisamment étudié et cet appendice ne doit pas être considéré comme définitif, mais comme informatif.

L'ensemble CS-1 du réseau intelligent est défini de façon à avoir l'incidence la plus faible possible sur les protocoles existants de signalisation entre commutateurs et entre réseau et utilisateur, y compris sur les systèmes sémaphores décadiques utilisés dans des réseaux analogiques pour lesquels, toutefois, certaines caractéristiques complémentaires nécessaires dépendent des applications. La conformité avec ces caractéristiques dépend du réseau et n'est pas spécifiée dans la présente Recommandation.

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

INTERFACE POUR L'ENSEMBLE DE CAPACITÉS 1 DU RÉSEAU INTELLIGENT

(Helsinki, 1993)

0 Introduction

La présente Recommandation spécifie le protocole d'application de réseau intelligent (INAP) (*intelligent network application protocol*) ou protocole INAP 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 RI:

- 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*)

0.1 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 la Recommandation X.208) 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 en 6/Q.1214 (voir 0.5 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 Recommandations 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.

Il est à noter que la sous-couche composantes du sous-système TCAP dispose d'une unité APDU supplémentaire, résultat partiel, absente de l'élément ROSE, qui ne peut être utilisée que si le protocole INAP est pris en charge par le sous-système commande des connexions sémaphores (sous-système SCCP) défini dans le *Livre bleu* (version 1988).

Le protocole INAP (en tant qu'utilisateur d'élément ROSE) et le protocole élément 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 (voir la Recommandation X.209).

0.2 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.

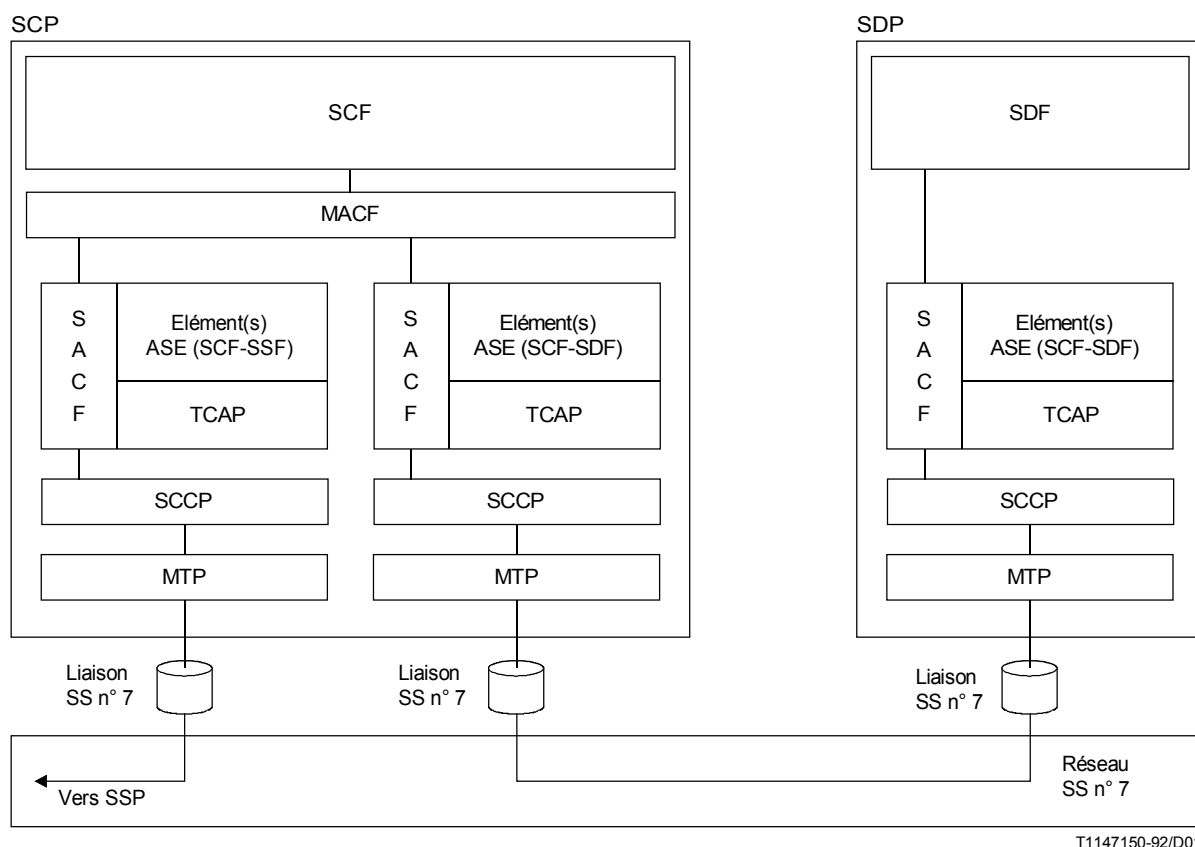
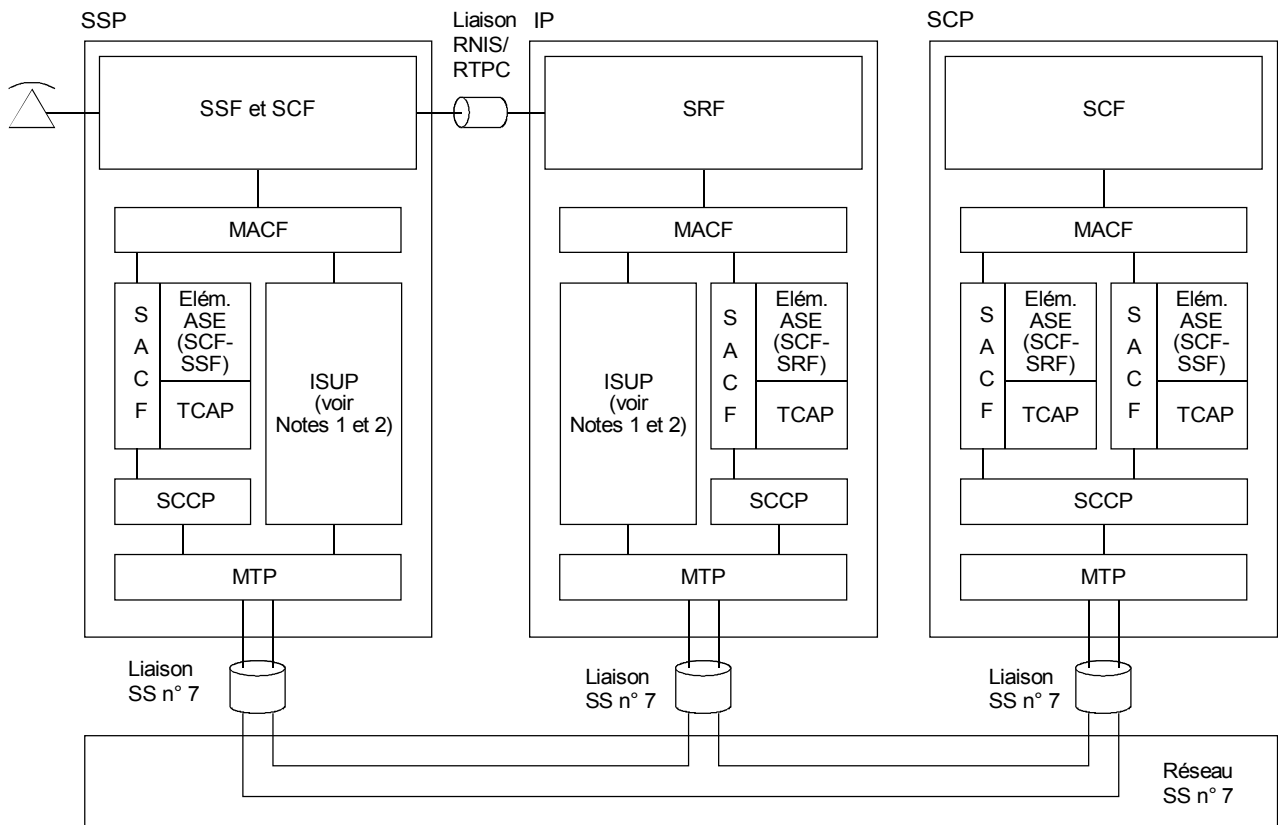


FIGURE 1/Q.1218
Interface physique entre points SCP et SDP

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.



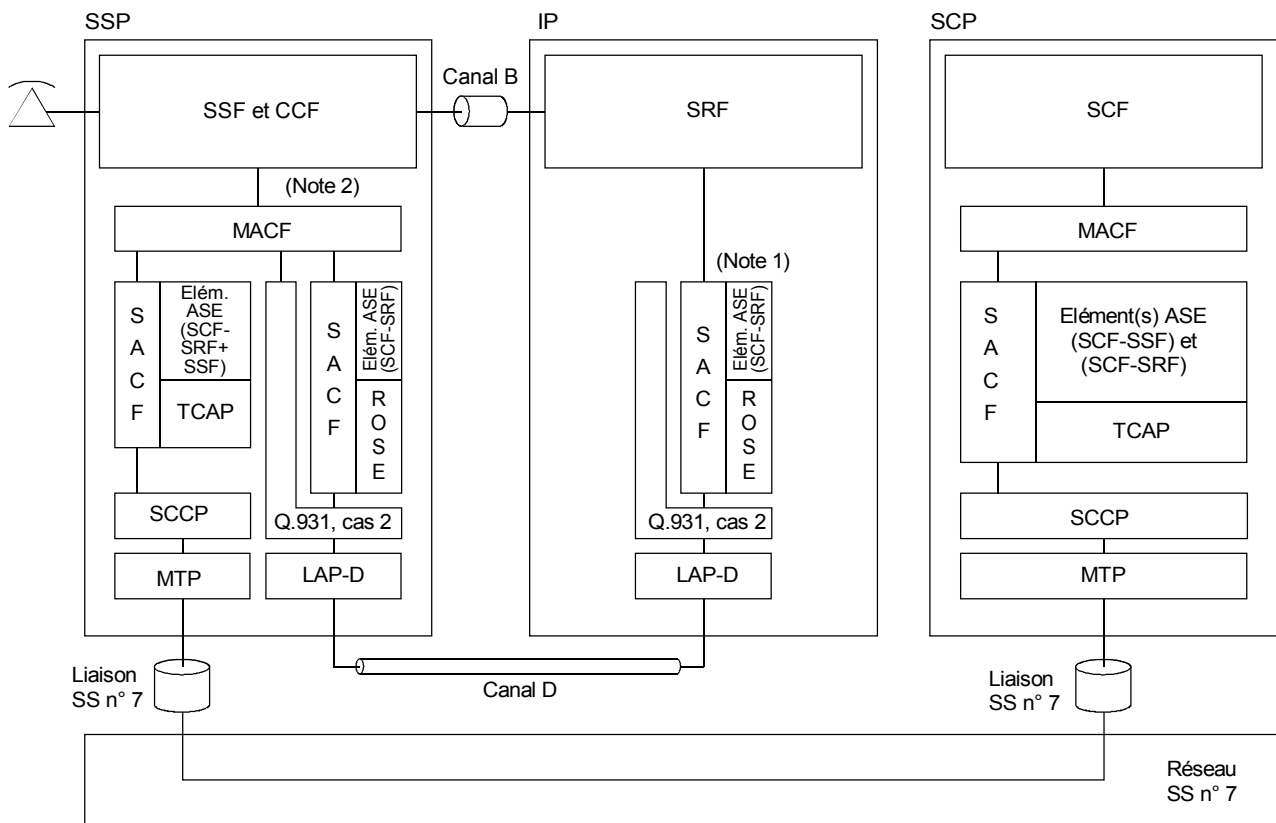
T1147160-92/D02

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)**



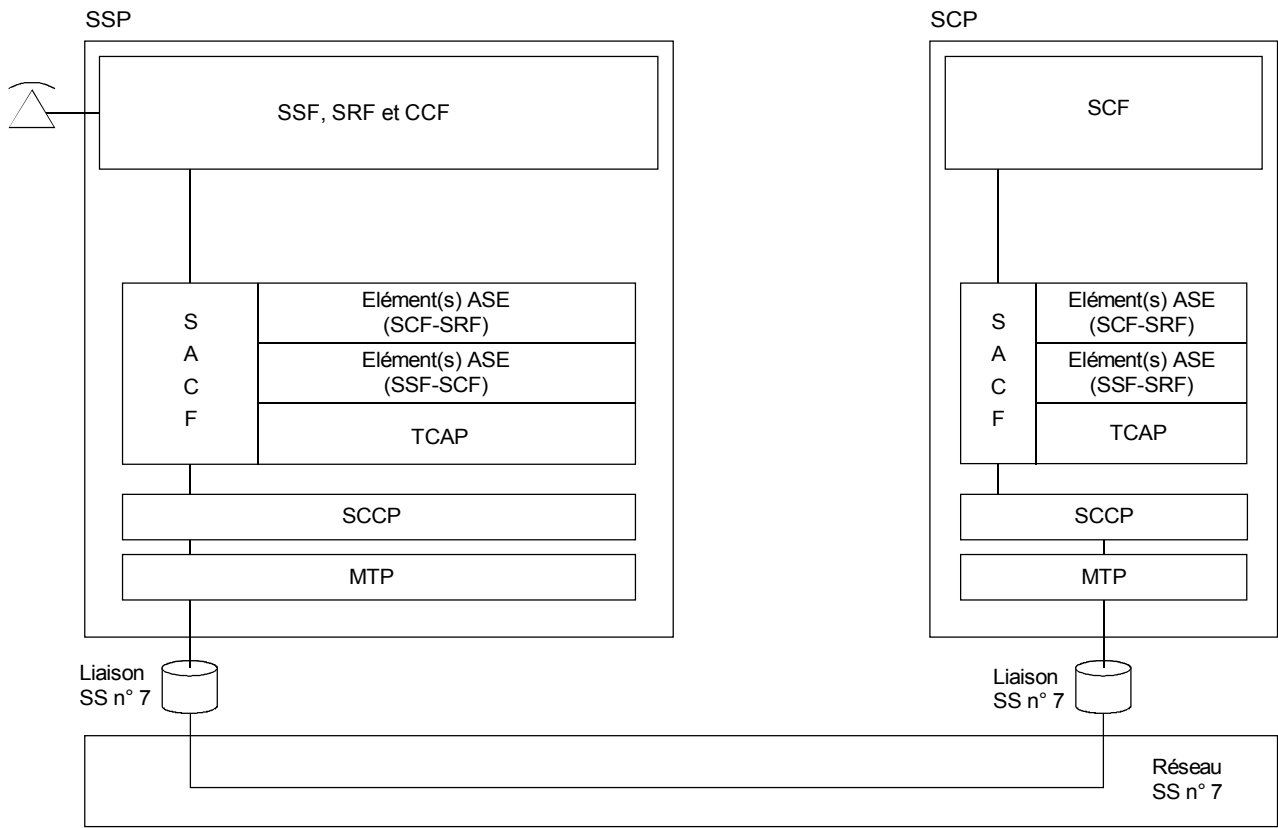
T1146670-92/D03

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

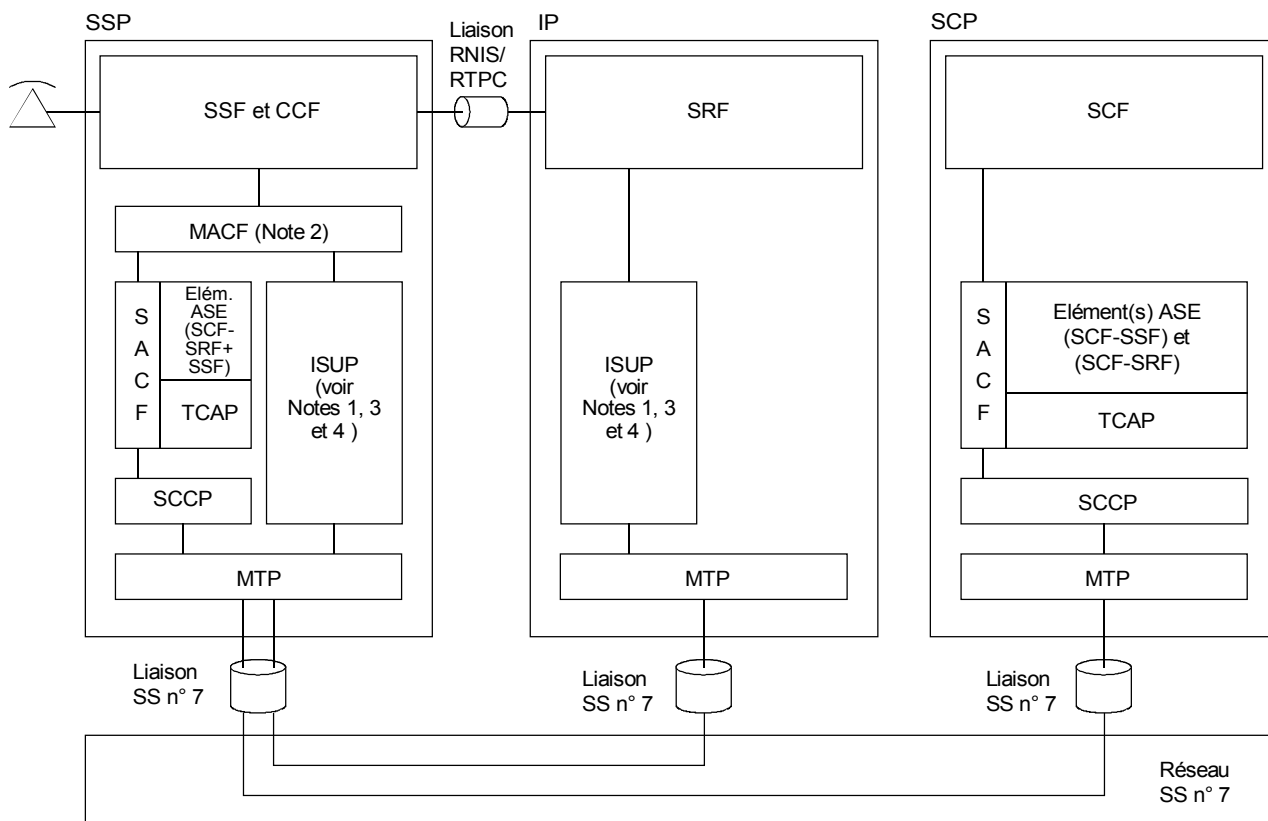
**Exemple d'architecture de prise en charge de la fonction SRF, cas 2
(fonction SRF dans un périphérique intelligent connecté au point SSP et avec accès
par le point SCP par l'intermédiaire d'un canal D via le point SSP)**



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)**



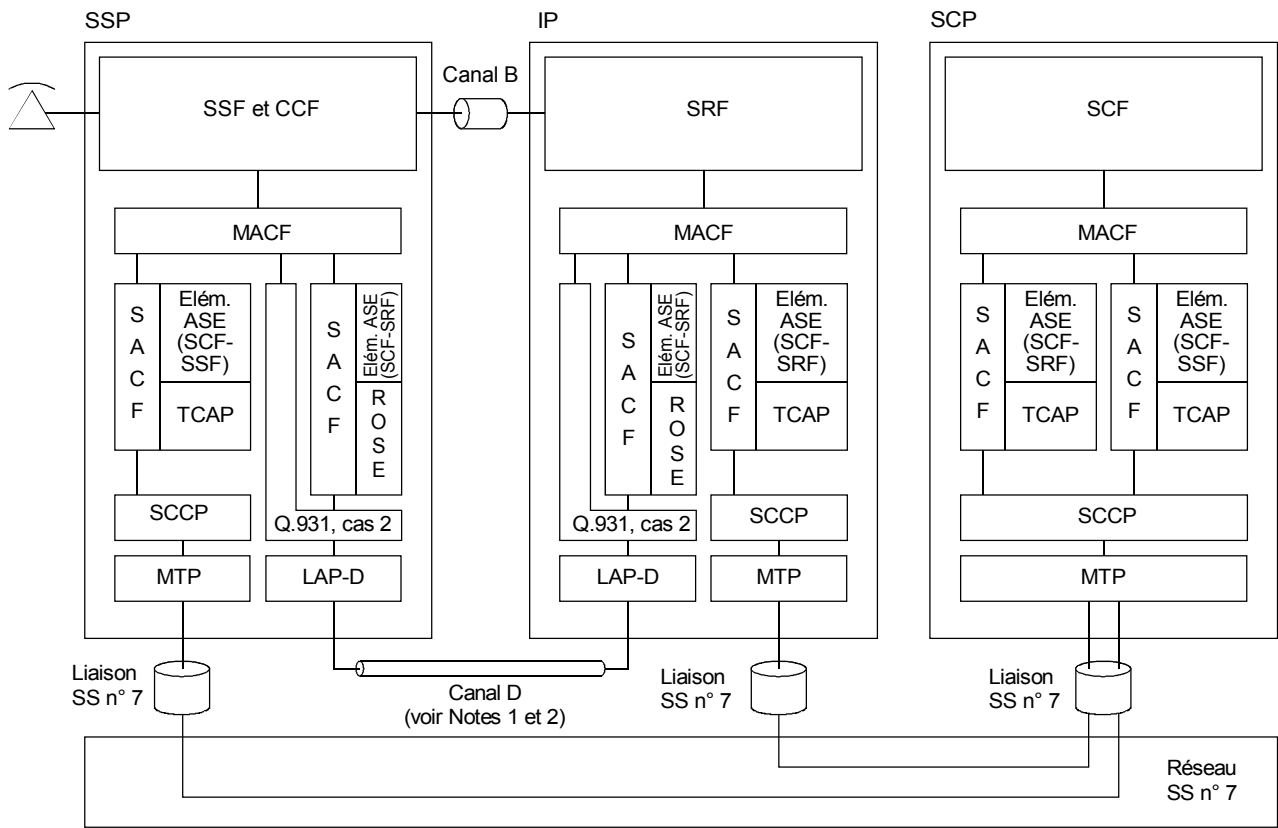
T1146690-92/D05

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> | | |

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

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

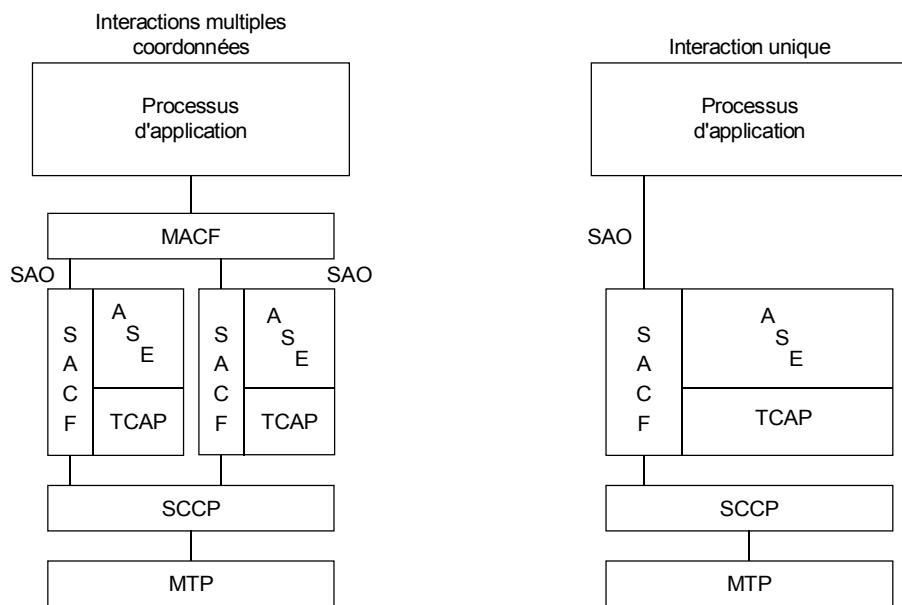
L'architecture du protocole INAP peut être celle de la Figure 7.

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.



T1146710-92/D07

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

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.

0.4 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 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.

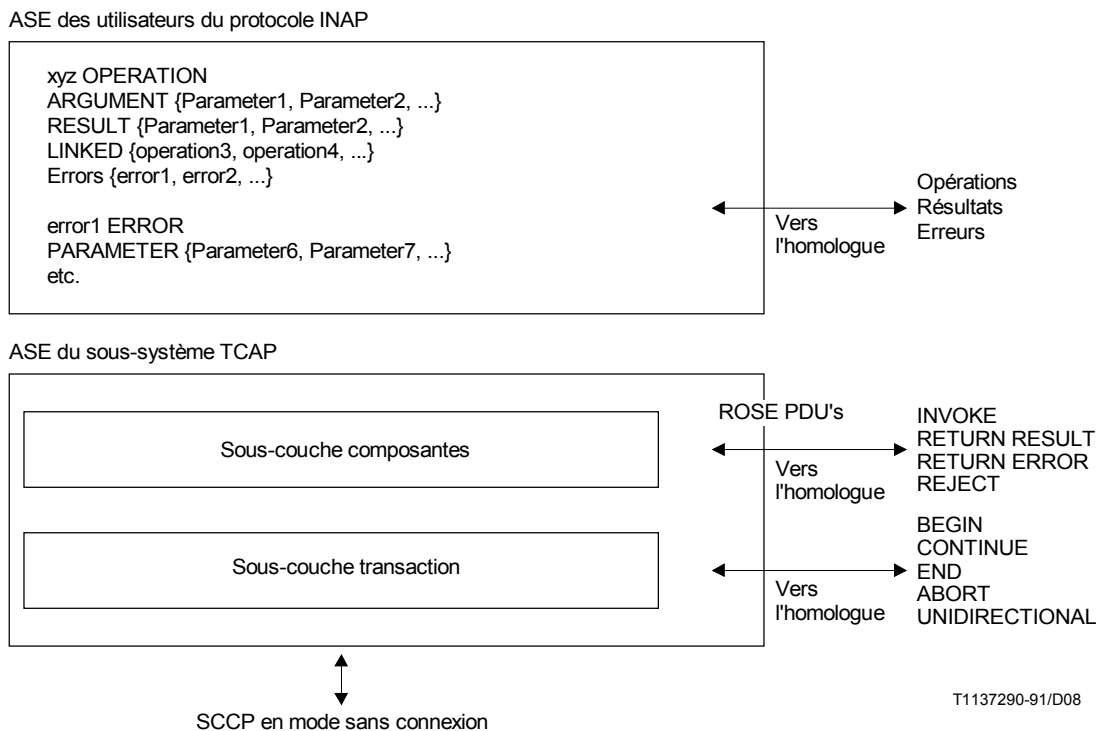


FIGURE 8/Q.1218
 Description du fonctionnement

0.5 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.

| Référence Rec. Q.1214 | Flux d'information | Nom de l'opération correspondante |
|--------------------------|--|-----------------------------------|
| 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 |
| 6.4.2.5 | Analyse des informations | AnalyseInformation |
| 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 |
| 6.4.2.12 | Annulation de demande d'informations d'appel | Cancel |
| 6.4.2.13 | Annulation de demande de compte rendu d'état | CancelStatusReportRequest |

| <i>Référence Rec. Q.1214</i> | <i>Flux d'information</i> | <i>Nom de l'opération correspondante</i> |
|----------------------------------|--|---|
| 6.4.2.14 | Informations recueillies | CollectedInformation |
| 6.4.2.15 | Recueil des informations | CollectInformation |
| 6.4.2.16 | Connexion | Connect |
| 6.4.2.17 | Connexion à la ressource | ConnectToResource |
| 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 |
| 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 |
| 6.4.2.28 | OCalledPartyBusy (occupation du demandé au départ) | OCalledPartyBusy |
| 6.4.2.29 | ODisconnect (déconnexion au départ) | ODisconnect |
| 6.4.2.30 | O_MidCall (semi-communication au départ) | O_MidCall |
| 6.4.2.31 | O_No_Answer (non-réponse au départ) | O_No_Answer |
| 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 de route | RouteSelectFailure |
| 6.4.2.39 | Sélection d'équipement | SelectFacility |
| 6.4.2.40 | Sélection de route | SelectRoute |
| 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 |
| 6.4.2.45 | TCalledPartyBusy (occupation du demandé à l'arrivée) | TCalledPartyBusy |
| 6.4.2.46 | TDisconnect (déconnexion à l'arrivée) | TDisconnect |
| 6.4.2.47 | Autorisation d'appel à l'arrivée | TermAttemptAuthorized |
| 6.4.2.48 | T_MidCall (semi-communication à l'arrivée) | T_MidCall |

| <i>Référence Rec. Q.1214</i> | <i>Flux d'information</i> | <i>Nom de l'opération correspondante</i> |
|----------------------------------|---|--|
| 6.4.2.49 | TNoAnswer (pas de réponse à l'arrivée) | TNoAnswer |
| 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 | Interrogation | Query |
| 6.6.2.2 | Résultat de l'interrogation | Return Result from Query |
| 6.6.2.3 | Réponse de la fonction SDF | SDFResponse |
| 6.6.2.4 | Confirmation de mise à jour | Return Result from Update Data |
| 6.6.2.5 | Mise à jour de données | UpdateData |

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.

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 permet simplement de justifier leur synchronisation.

2 Syntaxe abstraite du protocole d'application dans l'ensemble de capacités CS-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 la Recommandation X.208.

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).

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.

Le module contenant toutes les définitions des types des erreurs INAP est **IN-CS-1-Errors** et est décrit en 2.2.

Le module contenant toutes les définitions des types de données INAP est **IN-CS-1-DataTypes** et est décrit en 2.3.

Le module contenant les codes d'opérations et d'erreurs pour le protocole est **IN-CS-1-Codes** et est décrit en 2.4.

2.1 IN CS-1 Operation Types (Types d'opérations dans l'ensemble CS-1 du RI)

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,
ERROR

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

-- error types

Canceled,
CancelFailed,
DatabaseError,
ETCFailed,
ImproperCallerResponse,
InfoKeyError,
MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
Referral,
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,
QueryArg,
QueryResultArg,
ReceivedInformationArg,
ReleaseCallArg,
RequestCurrentStatusReportArg,
RequestCurrentStatusReportResultArg,
RequestEveryStatusChangeReportArg,
RequestFirstStatusMatchReportArg,
RequestNotificationChargingEventArg,
RequestReportBCSMEventArg,
ResetTimerArg,
RouteSelectFailureArg,
SelectFacilityArg,
SelectRouteArg,
SendChargingInformationArg,
ServiceFilteringResponseArg,
SpecializedResourceReportArg,
StatusReportArg,
TAnswerArg,
TCalledPartyBusyArg,
TDisconnectArg,
TermAttemptAuthorizedArg,
TNoAnswerArg,
UpdateDataArg,
UpdateDataResultArg

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

ERRORS {

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

-- **SCF → SSF**

-- 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 televoting 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

ARGUMENT

RESULT

-- **SCF → SSF**

-- 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,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- **SSF → SCF**

-- 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,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}**

-- **SCF → SSF**

-- 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 dialing 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,
ParameterOutOfRange,
SystemFailure,
TaskRefused
}**

-- SCF → SSF

-- This operation is used to interact with on line mechanisms that can be used for charging in the SSF.

ApplyChargingReport ::= OPERATION

ARGUMENT

ApplyChargingReportArg

-- SSF → SCF

-- This operation is used to report the results of charging in the SSF.

AssistRequestInstructions ::= OPERATION

ARGUMENT

AssistRequestInstructionsArg

ERRORS {

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

-- SSF → SCF or SRF → SCF

*-- 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 SSF or SRF to the SCF, when the SSF has set up a
-- connection to the SRF as a result of receiving a ConnectToResource or EstablishTemporaryConnection
-- operation from the SCF. Refer to clause 3 for a description of the procedures associated with this operation.*

CallGap ::= OPERATION

ARGUMENT

CallGapArg

-- SCF → SSF

*-- This operation is used to request the SSF reduce the rate at which specific service requests are sent to
-- the SSF. Use of this operation by the SCF to gap queries and updates at the SDF is for further study.*

CallInformationReport ::= OPERATION

ARGUMENT

CallInformationReportArg

-- SSF → SCF

*-- 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 {

**Canceled,
MissingParameter,
ParameterOutOfRange,
RequestedInfoError,
SystemFailure,
TaskRefused,**

```

        UnexpectedComponentSequence,
        UnexpectedParameter
    }

-- SCF → SSF
-- This operation is used to request the SSF to save specific information about a single call and report it to
-- the SCF at the end of the call (with a callInformationReport operation).

Cancel ::= OPERATION
    ARGUMENT
        CancelArg
    ERRORS {
        CancelFailed
    }

-- SCF → SSF, or SCF → SRF
-- This generic operation cancels the correlated previous operation. The following operations can be canceled:
-- PlayAnnouncement, PromptAndCollectUserInformation, and CallInformationRequest.

CancelStatusReportRequest ::= OPERATION
    ARGUMENT
        CancelStatusReportRequestArg
    ERRORS {
        CancelFailed
    }

-- SCF → SSF
-- This operation cancels the following processes: RequestFirstStatusMatchReport and
-- RequestEveryStatusChangeReport.

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

-- SSF → SCF
-- This operation is used to indicate availability of complete initial information package/dialing string from
-- originating party. (This event may have already occurred in the case of en bloc signaling, 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 4.2.2.2a)2)/Q.1214.

CollectInformation ::= OPERATION
    ARGUMENT
        CollectInformationArg
    ERRORS {
        MissingParameter,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
    }

-- SCF → SSF
-- 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,
SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- SCF → SSF

-- 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. dialed digits) and existing call setup information (e.g. route index to a list of trunk groups), depending on the information provided by the SCF.

ConnectToResource ::= OPERATION

ARGUMENT

ConnectToResourceArg

ERRORS {

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

-- SCF → SSF

-- 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

ARGUMENT

-- SCF → SSF

-- 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

ARGUMENT

ERRORS {

SystemFailure,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedParameter
}

-- SCF → SSF

-- This operation is used to disconnect a forward temporary connection and 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
}

-- SCF → SSF
-- 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

-- SSF → SCF
-- This operation is used to notify the SCF of a charging-related event previously requested
-- by the SCF in a RequestNotificationChargingEvent operation.

EventReportBCSM ::= OPERATION
ARGUMENT
 EventReportBCSMArg

-- SSF → SCF
-- 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
}

-- SCF → SSF
-- This operation is used to give some charging information to the SSF, to be used later in off line processing.

HoldCallInNetwork ::= OPERATION
ARGUMENT
 HoldCallInNetworkArg
ERRORS {
 MissingParameter,
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
}

-- SCF → SSF
-- This operation is used to provide the capability of queueing a call during the setup 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,
 SystemFailure,
 TaskRefused,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnexpectedParameter
}

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

InitiateCallAttempt ::= OPERATION

ARGUMENT

InitiateCallAttemptArg

ERRORS {

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

-- SCF → SSF

-- This operation is used to request the SSF to create a new call to one or more call parties using address information
-- provided by the SCF (e.g. wake-up call, predefined conference call, previous prompt and collect information).

OAnswer ::= OPERATION

ARGUMENT

OAnswerArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2a)4)/Q.1214.

OCalledPartyBusy ::= OPERATION

ARGUMENT

OCalledPartyBusyArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2a)4)/Q.1214.

ODisconnect ::= OPERATION

ARGUMENT

ODisconnectArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2a)5)/Q.1214.

OMidCall ::= OPERATION**ARGUMENT**

MidCallArg

ERRORS {

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

-- SSF → SCF

-- This operation is used to indicate a feature request is received from the originating party
-- (e.g. hook flash, ISDN feature activator, Q.931 HOLD or RETrieve message). (DP 8 – O_Mid_Call).
-- For additional information on this operation, refer to 4.2.2.2a)5)/Q.1214.

ONoAnswer ::= OPERATION**ARGUMENT**

ONoAnswerArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2a)4)/Q.1214.

OriginationAttemptAuthorized ::= OPERATION**ARGUMENT**

OriginationAttemptAuthorizedArg

ERRORS {

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

-- SSF → SCF

-- This operation is used to Indicate the desire to place outgoing call (e.g. offhook, Q.931 Setup 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 4.2.2.2a)1)/Q.1214.

ReleaseCall ::= OPERATION**ARGUMENT**

ReleaseCallArg

-- SCF → SSF

-- This operation is used to tear down an existing call at any phase of the call for 1, 2, or more parties
-- involved in the call.

RequestCurrentStatusReport ::= OPERATION**ARGUMENT**

RequestCurrentStatusReportArg

RESULT

RequestCurrentStatusReportResultArg

```

ERRORS {
    MissingParameter,
    ParameterOutOfRange,
    SystemFailure,
    TaskRefused,
    UnexpectedComponentSequence,
    UnexpectedParameter,
    UnknownResource
}

```

-- SCF → SSF

-- 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 {
        Canceled,
        MissingParameter,
        ParameterOutOfRange,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedParameter,
        UnknownResource
    }

```

-- SCF → SSF

-- 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 {
        Canceled,
        MissingParameter,
        ParameterOutOfRange,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedParameter,
        UnknownResource
    }

```

-- SCF → SSF

-- 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,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
    }

```

-- SCF → SSF

-- This operation is used to request the SSF to monitor for a charging-related event, then send a notification back to the SCF when the event is detected.

RequestReportBCSMEvent ::= OPERATION

ARGUMENT

RequestReportBCSMEventArg

ERRORS {

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

-- SCF → SSF

-- 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.

ResetTimer ::= OPERATION

ARGUMENT

ResetTimerArg

ERRORS {

MissingParameter,
TaskRefused,
UnexpectedComponentSequence,
UnexpectedDataValue,
UnexpectedParameter
}

-- SCF → SSF

-- This operation is used to request the SSF to refresh an application timer in the SSF.

RouteSelectFailure ::= OPERATION

ARGUMENT

RouteSelectFailureArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2a)4)/Q.1214.

SelectFacility ::= OPERATION

ARGUMENT

SelectFacilityArg

ERRORS {

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

-- SCF → SSF

-- 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,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
    }
-- SCF → SSF
-- 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
    }
-- SCF → SSF
-- This operation is used to request the SSF to generate specific charging messages.

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

StatusReport ::= OPERATION
    ARGUMENT
        StatusReportArg
-- SSF → SCF
-- This operation is used as a response to RequestFirstStatusMatchReport or
-- RequestEveryStatusChangeReport operations.

TAnswer ::= OPERATION
    ARGUMENT
        TAnswerArg
    ERRORS {
        MissingCustomerRecord,
        MissingParameter,
        SystemFailure,
        TaskRefused,
        UnexpectedComponentSequence,
        UnexpectedDataValue,
        UnexpectedParameter
    }
-- SSF → SCF
-- 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 – T_Answer). For additional information on this operation, refer to 4.2.2.2b)8)/Q.1214.

```

TCalledPartyBusy ::= OPERATION

ARGUMENT

TCalledPartyBusyArg

ERRORS {

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

-- SSF → SCF

-- This operation is used to indicate all resources in group busy (DP 13 – TCalledPartyBusy).

-- For additional information on this operation, refer to 4.2.2.2b)8)/Q.1214.

TDisconnect ::= OPERATION

ARGUMENT

TDisconnectArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2b)10)/Q.1214.

TermAttemptAuthorized ::= OPERATION

ARGUMENT

TermAttemptAuthorizedArg

ERRORS {

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

-- SSF → SCF

-- 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 4.2.2.2b)7)/Q.1214.

TMidCall ::= OPERATION

ARGUMENT

MidCallArg

ERRORS {

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

-- SSF → SCF
 -- This operation is used to indicate that a feature request is received from the terminating party (e.g. hook flash, ISDN feature activator, Q.931 HOLD or RETrieve message). (DP 16 – T_Mid_Call).
 -- For additional information on this operation, refer to 4.2.2.2a)10)/Q.1214.

TNoAnswer ::= OPERATION

ARGUMENT

TNoAnswerArg

ERRORS {

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

}

-- SSF → SCF
 -- 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 4.2.2.2b)9)/Q.1214.

-- SCF-SRF operations

-- AssistRequestInstructions

-- SRF → SCF

-- Refer to previous description of this operation in the SCF-SSF operations section Cancel

-- SCF → SRF

-- Refer to previous description of this operation in the SCF-SSF operations section.

PlayAnnouncement ::= OPERATION

ARGUMENT

PlayAnnouncementArg

ERRORS {

Canceled,
 MissingParameter,
 SystemFailure,
 UnexpectedComponentSequence,
 UnexpectedDataValue,
 UnavailableResource

}

LINKED {

SpecializedResourceReport

}

-- SCF → SRF
 -- 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 analog 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 {

Canceled,
 ImproperCallerResponse,
 MissingParameter,
 SystemFailure,
 TaskRefused,

```

        UnavailableResource,
        UnexpectedDataValue
    }

-- SCF → SRF
-- This operation is used to interact with a user to collect information.

SpecializedResourceReport ::= OPERATION
    ARGUMENT
        SpecializedResourceReportArg

-- SRF → SCF
-- This operation is used as the response to a PromptAndCollectUserInformation operation or as the response to
-- a PlayAnnouncement operation when the announcement completed report indication is set.
-- SCF-SDF operations

Query ::= OPERATION
    ARGUMENT
        QueryArg
    RESULT
        QueryResultArg
    ERRORS {
        DatabaseError,
        InfoKeyError,
        MissingParameter,
        Referral,
        RequestedInfoError,
        UnexpectedDataValue,
        UnexpectedParameter
    }
    LINKED {
        SdfResponse
    }

-- SCF → SDF
-- This operation is used to query an item of data held in the SDF (e.g. a translation of a Freephone number).
-- The timer associated with this operation must be of a sufficient duration to allow its linked operation
-- to be correctly correlated.

SdfResponse ::= OPERATION
    ARGUMENT

-- SDF → SCF
-- This operation is used to acknowledge the start of a Query or Update operation.

UpdateData ::= OPERATION
    ARGUMENT
        UpdateDataArg
    RESULT
        UpdateDataResultArg
    ERRORS {
        DatabaseError,
        InfoKeyError,
        MissingParameter,
        Referral,
        RequestedInfoError,
        UnexpectedDataValue,
        UnexpectedParameter
    }
    LINKED {
        SdfResponse
    }

-- SCF → SDF
-- This operation is used to update an item of data held in the SDF.

```

END

2.2 IN CS-1 Error Types (Types d'erreurs dans l'ensemble CS-1 du RI)

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.

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

ERROR

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

**AttributeID,
DatabaseID,
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

Canceled ::= ERROR

-- The operation has been canceled.

CancelFailed ::= ERROR

```
PARAMETER SEQUENCE {
    problem [0] ENUMERATED {
        unknownOperation(0),
        tooLate(1),
        operationNotCancelable(2)
    },
    operation [1] InvokeID
}
```

-- The operation failed to be canceled.

DatabaseError ::= ERROR

```
PARAMETER ENUMERATED {
    invalidDatabaseID(1),
    databaseCurrentlyUnavailable(2),
    databaseDenied(3),
    databaseRequestDenied(4)
    -- other values FFS
}
```

-- The SDF database could not be accessed.

ETCFailed ::= ERROR

-- The establish temporary connection failed.

ImproperCallerResponse ::= ERROR

-- The caller response was not as expected.

InfoKeyError ::= ERROR

```
PARAMETER SEQUENCE {
    errorType [0] ENUMERATED {
        invalidInfoKey(1),
        infoKeyNotUnique(2)
        -- other values FFS
    },
    attributeID [1] AttributeID }
}
```

-- The information key is invalid or ambiguous.

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).

Referral ::= ERROR -- reference to other database.

```
PARAMETER SEQUENCE {
    databaseAddress [0] OCTET STRING,
    databaseID[1] DatabaseID OPTIONAL
}
```

-- The SDF accessed does not have the data requested and instead refers the requesting
-- entity to another SDF.

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

2.3 IN CS-1 data types (Types de données dans l'ensemble CS-1 du RI)

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.

-- The following parameters map onto bearer protocol (i.e. Q.931, case 2 and ISUP) parameters:

-- CallingPartyBusinessGroupID, CallingPartySubaddress, CalledPartyNumber,
-- CalledPartyLineID, Prefix (derived from dialed digits), DestinationRoutingAddress,
-- DialedDigits, AccessTransport, CallingPartyCategory, LocationNumber,
-- TravellingClassMark, AssistingSSPIPRoutingAddress, AlertingPattern (Q.931 only),
-- ReleaseCause (and other Cause parameters), and SPID (Q.931 only).

-- 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: FacilityGroup,
-- FacilityGroupMember, RouteList, LegID, SSIPCapabilities, IPAvailable, CGEncountered,
-- ForwardingCondition, CorrelationID, Timers, TerminalType, MiscCallInfo, 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

ExtensionField

FROM { ccitt recommendation q 1400 modules(0) extension-example(1) version1(0) };

-- 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 | OPTIONAL, |
| startTime | [4] | DateAndTime | OPTIONAL, |
| extensions | [5] | SEQUENCE SIZE(0..MAX) | OF ExtensionField OPTIONAL |

}

-- filteringCriteria is required if this operation is not in the context of a call as determined by the SCF.

-- Depending on implementation choices, the filteringCriteria parameter can be used to correlate the

-- ServiceFilteringResponse with ActivateServiceFiltering. In this case, filteringCriteria parameter

-- is always required.

AnalysedInformationArg ::= SEQUENCE {

| | | | |
|------------------------------------|-------------|------------------------------------|-----------------------------------|
| dpSpecificCommonParameters | [0] | DpSpecificCommonParameters, | |
| dialedDigits | [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(0..MAX) | 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.

AnalyseInformationArg ::= SEQUENCE {
 destinationRoutingAddress **[0] DestinationRoutingAddress,**
 alertingPattern **[1] AlertingPattern** **OPTIONAL,**
 iSDNAccessRelatedInformation **[2] ISDNAccessRelatedInformation** **OPTIONAL,**
 originalCalledPartyID **[3] OriginalCalledPartyID** **OPTIONAL,**
 extensions **[4] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

ApplyChargingArg ::= SEQUENCE {
 billingChargingCharacteristics **[0] BillingChargingCharacteristics,**
 sendCalculationToSCFIndication **[1] BOOLEAN** **DEFAULT FALSE,**
 partyToCharge **[2] LegID** **OPTIONAL,**
 extensions **[3] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

ApplyChargingReportArg ::= CallResult

AssistRequestInstructionsArg ::= SEQUENCE {
 correlationID **[0] CorrelationID,**
 iPAvailable **[1] iPAvailable** **OPTIONAL,**
 iPSSPCapabilities **[2] iPSSPCapabilities** **OPTIONAL,**
 extensions **[3] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

-- OPTIONAL denotes network operator specific use. The value of the correlationID may be the Called Party Number supplied by the initiating SSF.

CallGapArg ::= SEQUENCE {
 gapCriteria **[0] GapCriteria,**
 gapIndicators **[1] GapIndicators,**
 controlType **[2] ControlType** **OPTIONAL,**
 gapTreatment **[3] GapTreatment** **OPTIONAL,**
 extensions **[4] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

-- OPTIONAL denotes network operator optional. If gapTreatment is not present, the SSF will use a default treatment depending on network operator implementation.

CallInformationReportArg ::= SEQUENCE {
 requestedInformationList **[0] RequestedInformationList,**
 correlationID **[1] CorrelationID** **OPTIONAL,**
 extensions **[2] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

-- OPTIONAL denotes network operator optional.

CallInformationRequestArg ::= SEQUENCE {
 requestedInformationTypeList **[0] RequestedInformationTypeList,**
 correlationID **[1] CorrelationID** **OPTIONAL,**
 extensions **[2] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

-- OPTIONAL denotes network operator optional.

CancelArg ::= invokeID[0] InvokeID

-- The InvokeID has the same value as that which was used for the operation to be cancelled.

CancelStatusReportRequestArg ::= SEQUENCE {
 resourceID **[0] ResourceID** **OPTIONAL,**
 extensions **[1] SEQUENCE SIZE(0..MAX)** **OF ExtensionField OPTIONAL**
}

CollectedInformationArg ::= 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, |
| travellingClassMark | [10] TravellingClassMark | OPTIONAL, |
| extensions | [11] SEQUENCE SIZE(0..MAX) | 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.

```
CollectInformationArg ::= SEQUENCE {
    alertingPattern          [0] AlertingPattern          OPTIONAL,
    numberingPlan           [1] NumberingPlan           OPTIONAL,
    originalCalledPartyID   [2] OriginalCalledPartyID   OPTIONAL,
    travellingClassMark     [3] TravellingClassMark     OPTIONAL,
    extensions              [4] SEQUENCE SIZE(0..MAX)   OF ExtensionField 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(0..MAX)   OF ExtensionField OPTIONAL
}
```

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

```
ConnectToResourceArg ::= SEQUENCE {
    CHOICE {
        ipRoutingAddress     [0] IPRoutingAddress,
        legID                 [1] LegID,
        both                  [2] SEQUENCE {
            ipRoutingAddress [0] IPRoutingAddress,
            legID             [1] LegID
        },
        none                  [3] NULL
    },
    extensions              [4] SEQUENCE SIZE(0..MAX)   OF ExtensionField OPTIONAL
}
```

```
DpSpecificCommonParameters ::= SEQUENCE {
    serviceAddressInformation [0] ServiceAddressInformation,
    bearerCapability          [1] BearerCapability          OPTIONAL,
    calledPartyNumber        [2] CalledPartyNumber        OPTIONAL,
    callingLineID            [3] CallingPartyNumber        OPTIONAL,
    callingPartysCategory    [4] CallingPartysCategory    OPTIONAL,
    iPSSPCapabilities        [5] IPSSPCapabilities        OPTIONAL,
    iPAvailable              [6] IPAvailable              OPTIONAL,
    iSDNAccessRelatedInformation [7] ISDNAccessRelatedInformation OPTIONAL,
    cGEncountered            [8] CGEncountered            OPTIONAL,
    locationNumber           [9] LocationNumber           OPTIONAL,
    serviceProfileIdentifier [10] ServiceProfileIdentifier  OPTIONAL,
    terminalType              [11] TerminalType              OPTIONAL,
    extensions              [12] SEQUENCE SIZE(0..MAX)   OF ExtensionField OPTIONAL
}
```

-- OPTIONAL for iPSSPCapabilities, iPAvailable, and cGEncountered denotes network operator specific use.
-- OPTIONAL for dialledDigits, callingLineID, and callingLineCategory 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(0..MAX) OF ExtensionField OPTIONAL
}

```

```

EventNotificationChargingArg ::= SEQUENCE {
    eventTypeCharging [0] EventTypeCharging,
    eventSpecificInformationCharging [1] EventSpecificInformationCharging OPTIONAL,
    legID [2] LegID OPTIONAL,
    extensions [3] SEQUENCE SIZE(0..MAX) OF ExtensionField OPTIONAL
}

```

-- 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 OPTIONAL,
    extensions [5] SEQUENCE SIZE(0..MAX) OF ExtensionField OPTIONAL
}

```

-- OPTIONAL denotes network operator specific use.

FurnishChargingInformationArg ::= BillingChargingCharacteristics

```

HoldCallInNetworkArg ::= CHOICE {
    holdcause [0] HoldCause,
    empty [1] NULL
}

```

-- holdcause is optional and denotes network operator specific use.

```

InitialDPArg ::= SEQUENCE {
    serviceKey [0] ServiceKey,
    dialledDigits [1] CalledPartyNumber OPTIONAL,
    calledPartyNumber [2] CalledPartyNumber OPTIONAL,
    callingLineID [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] IPAavailable 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(0..MAX) OF ExtensionField OPTIONAL
}

```

-- OPTIONAL for iPSSPCapabilities, iPAvailable, cGEncountered, and miscCallInfo denotes network operator specific use.

-- OPTIONAL for dialledDigits, callingLineID, and callingLineCategory 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(0..MAX) OF ExtensionField 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(0..MAX) | 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.

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(0..MAX) | 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(0..MAX) | 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.

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(0..MAX) | 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.

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(0..MAX) | 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.

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(0..MAX) | 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.

PlayAnnouncementArg ::= SEQUENCE {

| | | | |
|-----------------------------|-----|-----------------------|----------------------------|
| informationToSend | [0] | InformationToSend, | |
| disconnectFromIPForbidden | [1] | BOOLEAN | DEFAULT TRUE, |
| requestAnnouncementComplete | [2] | BOOLEAN | DEFAULT TRUE, |
| extensions | [3] | SEQUENCE SIZE(0..MAX) | OF ExtensionField OPTIONAL |

PromptAndCollectUserInformationArg ::= SEQUENCE {

| | | | |
|---------------------------|-----|-----------------------|----------------------------|
| collectedInfo | [0] | CollectedInfo, | |
| disconnectFromIPForbidden | [1] | BOOLEAN | DEFAULT TRUE, |
| informationToSend | [2] | InformationToSend | OPTIONAL, |
| extensions | [3] | SEQUENCE SIZE(0..MAX) | OF ExtensionField OPTIONAL |

QueryArg ::= SEQUENCE {

| | | | |
|-------------------|-----|-----------------------|----------------------------|
| informationKey | [0] | SET OF Attribute, | |
| databaseID | [1] | DatabaseID | OPTIONAL, |
| requestedInfoType | [2] | SET OF AttributeID | OPTIONAL, |
| extensions | [3] | SEQUENCE SIZE(0..MAX) | OF ExtensionField OPTIONAL |

-- requestedInfoType is used when the requested info type is known by the SDF.

QueryResultArg ::= SET OF Attribute

ReceivedInformationArg ::= CHOICE {

| | | |
|----------------|-----|---------------|
| digitsResponse | [0] | OCTET STRING, |
| 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(0..MAX) | OF ExtensionField OPTIONAL |

```

RequestEveryStatusChangeReportArg ::= SEQUENCE {
    resourceID          [0] ResourceID,
    correlationID       [1] CorrelationID          OPTIONAL,
    monitorDuration     [2] Duration              OPTIONAL,
    extensions          [3] SEQUENCE SIZE(0..MAX) 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,
    resourceStatus     [1] ResourceStatus,
    correlationID       [2] CorrelationID          OPTIONAL,
    monitorDuration     [3] Duration              OPTIONAL,
    extensions          [4] SEQUENCE SIZE(0..MAX) 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.

```

RequestNotificationChargingEventArg ::= SEQUENCE SIZE(0..MAX) OF ChargingEvent

```

```

RequestReportBCSMEventArg ::= SEQUENCE {
    bcsmEvents         [0] SEQUENCE SIZE(0..MAX) OF BCSMEvent,
    bcsmEventCorrelationID [1] CorrelationID          OPTIONAL,
    extensions         [2] SEQUENCE SIZE(0..MAX) 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(0..MAX) 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(0..MAX) 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.

```

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

```

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

```

SendChargingInformationArg ::= SEQUENCE {
    billingChargingCharacteristics [0] BillingChargingCharacteristics,
    legID [1] LegID,
    extensions [2] SEQUENCE SIZE(0..MAX) OF ExtensionField OPTIONAL
}

```

```

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

```

SpecializedResourceReportArg ::= NULL

```

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

```

```

TCalledPartyBusyArg ::= 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(0..MAX) 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.

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(0..MAX) OF ExtensionField 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(0..MAX) OF ExtensionField 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(0..MAX) OF ExtensionField OPTIONAL
}

UpdateDataArg ::= SEQUENCE {
 informationKey [0] SET OF Attribute,
 updatedInfo [1] SET OF Attribute,
 databaseID [2] DatabaseID OPTIONAL,
 extensions [3] SEQUENCE SIZE(0..MAX) OF ExtensionField OPTIONAL
}

-- The function type is coded in the informationKey.

UpdateDataResultArg ::= SEQUENCE SIZE (0..MAX) OF Attribute

-- The Definition of Common Data Types Follows

AlertingPattern ::= OCTET STRING

-- 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

-- 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.

Attribute ::= SEQUENCE {
 attributeID [0] AttributeID,
 attributeValues [1] AttributeValue
}

AttributeID ::= INTEGER

AttributeValue ::= ANY DEFINED BY attributeID

-- currently defined attribute types:

| | AttributeID | Type of Attribute value |
|----------------------------|-------------|--|
| -- freeAttribute | 0 | OCTET STRING (SIZE(0..MAX)) |
| -- functionType | 1 | ENUMERATED (update(0), increment(1), decrement(2)) |
| -- calledAddress | 2 | Digits |
| -- callingAddress | 3 | Digits |
| -- personalIDNumber | 4 | Digits |
| -- accountNumber | 5 | Digits |
| -- origAdministrationID | 6 | Digits |
| -- bearerCapability | 7 | BearerCapability |
| -- callingTermCapabilities | 8 | TermCapability |
| -- onNetOffNetIndicator | 9 | ENUMERATED { on(0), off(1), undetermined(2) } |

-- other types FFS

BCSMEvent ::= SEQUENCE {

| | | |
|------------------|-----------------------------|------------------|
| eventTypeBCSM | [0] EventTypeBCSM, | |
| monitorMode | [1] MonitorMode, | |
| legID | [2] LegID | OPTIONAL, |
| applicationTimer | [3] ApplicationTimer | OPTIONAL |

}

-- Indicates the BCSM Event information for monitoring. The SCF may set a timer in the SSF for the
-- No Answer event. If the user doesn't answer the call within the allotted time, the SSF reports the event
-- to the SCF.

BearerCapability ::= CHOICE {

| | |
|------------------|---|
| bearerCapability | [0] OCTET STRING (SIZE(2..MAX)), |
| 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.

BillingChargingCharacteristics ::= OCTET STRING

-- Indicates the billing and/or charging calculation characteristics. The value of this octet string is
-- network operator specific.

CalledPartyBusinessGroupID ::= OCTET STRING

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

CalledPartyNumber ::= OCTET STRING

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

CalledPartySubaddress ::= OCTET STRING

-- Indicates the Called Party Subaddress. Refer to 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

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

CallingPartySubaddress ::= OCTET STRING

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

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

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

CallResult ::= OCTET STRING

-- Indicates call result for billing purposes. The value of this octet string is network operator specific.

Cause ::= OCTET STRING

-- Indicates the cause for interface related information. Refer to the Q.763 Cause parameter for encoding.

CGEncountered ::= ENUMERATED {
 noCGencountered(0),
 manualCGencountered(1),
 scpOverload(2)
}

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

ChargingEvent ::= SEQUENCE {
 eventTypeCharging **[0] EventTypeCharging,**
 monitorMode **[1] MonitorMode,**
 legID **[2] LegID** **OPTIONAL**
}

-- Indicates the charging event information for monitoring.

CollectedDigits ::= SEQUENCE {
 minimumNbOfDigits **[0] INTEGER** **DEFAULT 1,**
 maximumNbOfDigits **[1] INTEGER,**
 endOfReplyDigit **[2] OCTET STRING (SIZE (1))** **OPTIONAL,**
 cancelDigit **[3] OCTET STRING (SIZE (1))** **OPTIONAL,**
 startDigit **[4] OCTET STRING (SIZE (1))** **OPTIONAL,**
 firstDigitTimeOut **[5] INTEGER** **OPTIONAL,**
 interDigitTimeOut **[6] INTEGER** **OPTIONAL,**
 errortreatment **[7] ErrorTreatment** **DEFAULT stdErrorAndInfo,**
 interruptableAnnInd **[8] BOOLEAN** **DEFAULT TRUE,**
 voiceInformation **[9] BOOLEAN** **OPTIONAL,**
 voiceBack **[10] BOOLEAN** **OPTIONAL**
}

-- The use of voiceBack is network operator specific.

-- The endOfReplyDigit, cancelDigit, and startDigit parameters have been designated as OCTET STRING,

-- and are to be encoded as Q.763 address signal digits in the Called Party Number. The endOfReplyDigit,

-- cancelDigit, and startDigit are one digit only with: code 11 = *, code 12 = #.

CollectedInfo ::= CHOICE {
 collectedDigits **[0] CollectedDigits,**
 iA5Information **[1] BOOLEAN (TRUE)**
}

ControlType ::= ENUMERATED {
 sCPOverloaded(0),
 manuallyInitiated(1),
 destinationOverload(2)
 -- other values FFS
}

CorrelationID ::= Digits

-- used by SCF for correlation with a previous operation. Refer to clause 3 for a description of the procedures

-- associated with this parameter.

CounterAndValue ::= SEQUENCE {
 counterID **[0] CounterID,**
 counterValue **[1] INTEGER**
}

CounterID ::= INTEGER (0..99)

-- Indicates the counters to be incremented.

-- The counterIDs can be addressed by using the last digits of the dialed number.

CountersValue ::= SEQUENCE SIZE(0..99) OF CounterAndValue

CutAndPaste ::= INTEGER

-- Indicates the number of digits to be deleted. Refer to 6.4.2.16/Q.1214 for additional information.

DatabaseID ::= OCTET STRING

-- Indicates the database to query. The value of this octet string is network operator specific.

DateAndTime ::= OCTET STRING (SIZE(6))

-- Indicates the start time for activate service filtering. Coded as YYMMDDHHMMSS with each digit coded BCD.

-- The first octet contains YY and the remaining items are sequenced following.

DestinationRoutingAddress ::= SEQUENCE SIZE(1..3) OF CalledPartyNumber

-- Indicates the list of Called Party Numbers (primary and alternates).

Digits ::= OCTET STRING

-- Indicates the address signalling digits. Refer to the Q.763 Generic Number and Generic Digits parameters

-- for encoding. The following parameter should use Generic Number:

-- CorrelationID for AssistRequestInstructions. The following parameters should use Generic Digits: prefix, all

-- other CorrelationID occurrences, personalIDNumber attribute, accountNumber attribute, origAdministrationID

-- attribute, calledAddress attribute, callingAddress attribute, dialledNumber filtering criteria, callingLineID

-- filtering criteria, lineID ResourceID type.

DisplayInformation ::= IA5String

-- Indicates the display information.

Duration ::= INTEGER (0..2047)

-- Values are seconds

ErrorTreatment ::= ENUMERATED {

stdErrorAndInfo(0),

help(1),

repeatPrompt(2) }

EventSpecificInformationBCSM ::= OCTET STRING -- defined by network operator.

-- Indicates the call related information specific to the event.

EventSpecificInformationCharging ::= OCTET STRING -- defined by network operator.

-- Indicates the charging related information specific to the event.

EventTypeBCSM ::= ENUMERATED {

origAttemptAuthorized(1),

collectInfo(2),

analysedInformation(3),

routeSelectFailure(4),

oCalledPartyBusy(5),

oNoAnswer(6),

oAnswer(7),

oMidCall(8),

oDisconnect(9),

oAbandon(10),

reserved(11),

termAttemptAuthorized(12),

tCalledPartyBusy(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.

EventTypeCharging ::= OCTET STRING

-- Indicates the charging event. Network operator specific based on Q.763 Charging Indicators message.

```

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.

```

FeatureRequestIndicator ::= ENUMERATED {
    hold(0),
    retrieve(1),
    featureActivator(2),
    spare1(3),
    sparen(127)
}

```

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

```

FilteredCallTreatment ::= SEQUENCE {
    billingChargingCharacteristics [0] BillingChargingCharacteristics,
    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.

```

FilteringCharacteristics ::= CHOICE {
    interval [0] INTEGER,
    numberOfCalls [1] INTEGER
}

```

-- 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. If = NumberOfCalls, every N calls the Nth call leads to an InitialDP and a ServiceFilteringResponse
-- is sent to the SCF.

```

FilteringCriteria ::= CHOICE {
    dialledNumber [0] Digits,
    callingLineID [1] Digits,
    serviceKey [2] ServiceKey
}

```

```

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.

```

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

```

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

GapCriteria ::= CHOICE {
 calledPartyNumber **[0] Digits,**
 callingPartyNumber **[1] Digits,**
 gapOnService **[2] GapOnService**
}

-- Both calledPartyNumber and callingPartyNumber can be incomplete numbers, in the sense that a limited amount of digits can be given. In that case the gapping will apply to all numbers which start with the same string of digits.

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.

HoldCause ::= OCTET STRING -- defined by network operator.

-- Indicates the cause for holding the call.

InbandInfo ::= SEQUENCE {
 messageID **[0] MessageID,**
 numberOfRepetitions **[1] INTEGER** **DEFAULT 1,**
 duration **[2] INTEGER** **OPTIONAL,**
 interval **[3] INTEGER** **OPTIONAL**
}

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

Interval ::= INTEGER (-1..60000)

-- Units are milliseconds. A -1 value denotes infinite.

InvokeID ::= INTEGER

-- Operation invoke identifier.

IPAvailable ::= OCTET STRING -- defined by network operator.

-- Indicates that the resource is available.

IPRoutingAddress ::= CalledPartyNumber

-- Indicates the routing address for the IP.

IPSSPCapabilities ::= OCTET STRING -- 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] OCTET STRING,
    receivingSideID        [1] OCTET STRING,
    both                    [2] SEQUENCE {
        sendingSideID [0] OCTET STRING,
        receivingSideID [1] OCTET STRING
    }
}

```

-- 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.

LocationNumber ::= OCTET STRING

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

MaximumNumberOfCounters ::= INTEGER (0..99)

```

MessageID ::= CHOICE {
    elementaryMessageID    [0] INTEGER,
    text                   [1] SEQUENCE {
        messageContent [0] IA5String,
        attributes      [1] OCTET STRING OPTIONAL
    }
}

```

-- 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.

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

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

OriginalCalledPartyID ::= OCTET STRING

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

RedirectingPartyID ::= OCTET STRING

-- 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.

RequestedInformationList ::= SEQUENCE OF RequestedInformation

RequestedInformationTypeList ::= SEQUENCE OF RequestedInformationType

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

RequestedInformationType ::= ENUMERATED {
 callAttemptElapsedTime(0),
 callStopTime(1),
 callConnectedElapsedTime(2),
 calledAddress(3),
 callingAddress(4),
 bearerCapability(5)
}

RequestedInformationValue ::= CHOICE {
 callAttemptElapsedTimeValue **[0] INTEGER,**
 callStopTimeValue **[1] DateAndTime,**
 callConnectedElapsedTimeValue **[2] INTEGER,**
 calledAddressValue **[3] Digits,**
 callingAddressValue **[4] Digits,**
 bearerCapabilityValue **[5] BearerCapability**
}

-- Units for the time values are network operator specific.

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)
}

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

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

ScfID ::= OCTET STRING *-- defined by network operator.*

-- Indicates the SCF identifier.

ServiceAddressInformation ::= SEQUENCE {
 serviceKey **[0] ServiceKey,**
 miscCallInfo **[1] MiscCallInfo**
}

*-- Information that allows the SCF to choose the appropriate service logic with additional DP
-- information that is optional for network operator specific use.*

ServiceKey ::= INTEGER

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

ServiceProfileIdentifier ::= OCTET STRING

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

TerminalType ::= ENUMERATED {
 unknown(0),
 dialPulse(1),
 dtmf(2),
 isdn(3),
 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 ::= INTEGER
```

-- Indicates the timer value (in seconds).

```
Tone ::= SEQUENCE {
    toneID [0] INTEGER,
    duration [1] INTEGER OPTIONAL
}
```

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

-- Indicates travelling class mark information. The value of this octet string is network operator specific.

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

-- Indicates the network resource that failed.

END

2.4 IN CS-1 application protocol (operation and error codes) [Protocole d'application de l'ensemble CS-1 du RI (codes d'opérations et d'erreurs)]

-- 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 FOLLOWS

IMPORTS

-- 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,
Query,
ReleaseCall,
RequestCurrentStatusReport,
RequestEveryStatusChangeReport,
RequestFirstStatusMatchReport,
RequestNotificationChargingEvent,
RequestReportBCSMEvent,
ResetTimer,
RouteSelectFailure,
SdfResponse,
SelectFacility,
SelectRoute,
SendChargingInformation,
ServiceFilteringResponse,
SpecializedResourceReport,
StatusReport,
TAnswer,
TCalledPartyBusy,
TDisconnect,
TermAttemptAuthorized,
TMidCall,
TNoAnswer,
UpdateData

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

-- error types

Canceled,
CancelFailed,
DatabaseError,
ETCFailed,
ImproperCallerResponse,
InfoKeyError,
MissingCustomerRecord,
MissingParameter,
ParameterOutOfRange,
Referral,
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) };

-- the operations are grouped by the identified ASEs.

-- **SCF activation ASE**

initialDP InitialDP ::= localValue 0

-- **Basic BCP DP ASE**

originationAttemptAuthorized OriginationAttemptAuthorized ::= localValue 1
collectedInformation CollectedInformation ::= localValue 2
analysedInformation AnalysedInformation ::= localValue 3
routeSelectFailure RouteSelectFailure ::= localValue 4
oCalledPartyBusy OCalledPartyBusy ::= localValue 5
oNoAnswer ONoAnswer ::= localValue 6
oAnswer OAnswer ::= localValue 7
oDisconnect ODisconnect ::= localValue 8
termAttemptAuthorized TermAttemptAuthorized ::= localValue 9
tCalledPartyBusy TCalledPartyBusy ::= localValue 10
tNoAnswer TNoAnswer ::= localValue 11
tAnswer TAnswer ::= localValue 12
tDisconnect TDisconnect ::= localValue 13

-- **Advanced BCP DP ASE**

oMidCall OMidCall ::= localValue 14
tMidCall TMidCall ::= localValue 15

-- **SCF/SRF activation of assist ASE**

assistRequestInstructions AssistRequestInstructions ::= localValue 16

-- **Assist connection establishment ASE**

establishTemporaryConnection EstablishTemporaryConnection ::= localValue 17

-- **Generic disconnect resource ASE**

disconnectForwardConnection DisconnectForwardConnection ::= localValue 18

-- **Non-assisted connection establishment ASE**

connectToResource ConnectToResource ::= localValue 19

-- **Connect ASE (elementary SSF function)**

connect Connect ::= localValue 20

-- **Call handling ASE (elementary SSF function)**

holdCallInNetwork HoldCallInNetwork ::= localValue 21
releaseCall ReleaseCall ::= localValue 22

-- **BCSM Event handling ASE**

requestReportBCSMEvent RequestReportBCSMEvent ::= localValue 23
eventReportBCSM EventReportBCSM ::= localValue 24

-- **Charging Event handling ASE**

requestNotificationChargingEvent RequestNotificationChargingEvent ::= localValue 25
eventNotificationCharging EventNotificationCharging ::= localValue 26

-- **SSF call processing ASE**

collectInformation CollectInformation ::= localValue 27
analyseInformation AnalyseInformation ::= localValue 28
selectRoute SelectRoute ::= localValue 29
selectFacility SelectFacility ::= localValue 30
continueContinue ::= localValue 31

-- **SCF call initiation ASE**

initiateCallAttempt InitiateCallAttempt ::= localValue 32

-- **Timer ASE**

resetTimer ResetTimer ::= localValue 33

-- **Billing ASE**

furnishChargingInformation FurnishChargingInformation ::= localValue 34

-- Charging ASE

| | | |
|----------------------------|----------------------------|-------------------|
| <i>applyCharging</i> | <i>ApplyCharging</i> | ::= localValue 35 |
| <i>applyChargingReport</i> | <i>ApplyChargingReport</i> | ::= localValue 36 |

-- Status reporting ASE

| | | |
|---------------------------------------|---------------------------------------|-------------------|
| <i>requestCurrentStatusReport</i> | <i>RequestCurrentStatusReport</i> | ::= localValue 37 |
| <i>requestEveryStatusChangeReport</i> | <i>RequestEveryStatusChangeReport</i> | ::= localValue 38 |
| <i>requestFirstStatusMatchReport</i> | <i>RequestFirstStatusMatchReport</i> | ::= localValue 39 |
| <i>statusReport</i> | <i>StatusReport</i> | ::= localValue 40 |

-- Traffic management ASE

| | | |
|----------------|----------------|-------------------|
| <i>callGap</i> | <i>CallGap</i> | ::= localValue 41 |
|----------------|----------------|-------------------|

-- Service management ASE

| | | |
|---------------------------------|---------------------------------|-------------------|
| <i>activateServiceFiltering</i> | <i>ActivateServiceFiltering</i> | ::= localValue 42 |
| <i>serviceFilteringResponse</i> | <i>ServiceFilteringResponse</i> | ::= localValue 43 |

-- Call report ASE

| | | |
|-------------------------------|-------------------------------|-------------------|
| <i>callInformationReport</i> | <i>CallInformationReport</i> | ::= localValue 44 |
| <i>callInformationRequest</i> | <i>CallInformationRequest</i> | ::= localValue 45 |

-- Signalling control ASE

| | | |
|--------------------------------|--------------------------------|-------------------|
| <i>sendChargingInformation</i> | <i>SendChargingInformation</i> | ::= localValue 46 |
|--------------------------------|--------------------------------|-------------------|

-- Specialized resource control ASE

| | | |
|--|--|-------------------|
| <i>playAnnouncement</i> | <i>PlayAnnouncement</i> | ::= localValue 47 |
| <i>promptAndCollectUserInformation</i> | <i>PromptAndCollectUserInformation</i> | ::= localValue 48 |
| <i>specializedResourceReport</i> | <i>SpecializedResourceReport</i> | ::= localValue 49 |

-- User data manipulation ASE

| | | |
|--------------------|--------------------|-------------------|
| <i>query</i> | <i>Query</i> | ::= localValue 50 |
| <i>sdfResponse</i> | <i>SdfResponse</i> | ::= localValue 51 |
| <i>updateData</i> | <i>UpdateData</i> | ::= localValue 52 |

-- Cancel ASE

| | | |
|----------------------------------|----------------------------------|-------------------|
| <i>cancel</i> | <i>Cancel</i> | ::= localValue 53 |
| <i>cancelStatusReportRequest</i> | <i>CancelStatusReportRequest</i> | ::= localValue 54 |

-- Activity Test ASE

| | | |
|---------------------|---------------------|-------------------|
| <i>activityTest</i> | <i>ActivityTest</i> | ::= localValue 55 |
|---------------------|---------------------|-------------------|

-- ERROR codes

| | | |
|------------------------------------|------------------------------------|-------------------|
| <i>canceled</i> | <i>Canceled</i> | ::= localValue 0 |
| <i>cancelFailed</i> | <i>CancelFailed</i> | ::= localValue 1 |
| <i>databaseError</i> | <i>DatabaseError</i> | ::= localValue 2 |
| <i>eTCFailed</i> | <i>ETCFailed</i> | ::= localValue 3 |
| <i>improperCallerResponse</i> | <i>ImproperCallerResponse</i> | ::= localValue 4 |
| <i>infoKeyError</i> | <i>InfoKeyError</i> | ::= localValue 5 |
| <i>missingCustomerRecord</i> | <i>MissingCustomerRecord</i> | ::= localValue 6 |
| <i>missingParameter</i> | <i>MissingParameter</i> | ::= localValue 7 |
| <i>parameterOutOfRange</i> | <i>ParameterOutOfRange</i> | ::= localValue 8 |
| <i>referral</i> | <i>Referral</i> | ::= localValue 9 |
| <i>requestedInfoError</i> | <i>RequestedInfoError</i> | ::= localValue 10 |
| <i>systemFailure</i> | <i>SystemFailure</i> | ::= localValue 11 |
| <i>taskRefused</i> | <i>TaskRefused</i> | ::= localValue 12 |
| <i>unavailableResource</i> | <i>UnavailableResource</i> | ::= localValue 13 |
| <i>unexpectedComponentSequence</i> | <i>UnexpectedComponentSequence</i> | ::= localValue 14 |
| <i>unexpectedDataValue</i> | <i>UnexpectedDataValue</i> | ::= localValue 15 |
| <i>unexpectedParameter</i> | <i>UnexpectedParameter</i> | ::= localValue 16 |
| <i>unknownLegID</i> | <i>UnknownLegID</i> | ::= localValue 17 |
| <i>unknownResource</i> | <i>UnknownResource</i> | ::= localValue 18 |

END

3 Procédures

3.1 Définition des procédures et des entités

3.1.1 Procédures relatives à l'entité application de l'entité SSF

3.1.1.1 Considérations générales

Le présent paragraphe définit les procédures relatives à l'entité application (AE) (*application entity*) de l'entité SSF et utilisées à l'interface SSP-SCP. Ces procédures reposent sur le système de signalisation par canal sémaphore n° 7 (système CCSS n° 7); d'autres systèmes de signalisation peuvent être utilisés (par exemple, la couche 3 du système DSS 1).

Les capacités non explicitement concernées par ces procédures peuvent être prises en charge dans le point SSP de diverses manières selon la réalisation, mais en conformité avec l'article 2.

Selon l'architecture définie dans les Recommandations Q.700, Q.771 et Q.1400, les entités application comportent un sous-système application de gestion des transactions (TCAP) (*transaction capabilities application part*) et un ou plusieurs éléments ASE appelés utilisateurs TC. Les éléments ASE d'utilisateur TC assurant l'interface avec le sous-système TCAP à l'aide des primitives spécifiées dans la Recommandation Q.771 sont définis dans les paragraphes ci-après; d'autres systèmes de signalisation comme le système DSS 1 (couche 3) peuvent être utilisés.

Cette procédure peut également s'utiliser avec d'autres systèmes de transport de message de signalisation prenant en charge les structures définies pour la couche application.

3.1.1.2 Modèle et interfaces

La Figure 9 représente le modèle fonctionnel de l'entité application de l'entité SSF; les éléments ASE sont reliés au sous-système TCAP pour communiquer avec l'entité SCF et assurent l'interface avec la fonction commande d'appel (CCF) (*call control function*) et les fonctions de maintenance déjà définies pour les systèmes de commutation. Le champ d'application de la présente Recommandation est limité à la partie ombrée sur la Figure 9.

Les interfaces représentées sur la Figure 9 utilisent les primitives d'élément ASE d'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 paramètres du protocole INAP sont définis en 2.

3.1.1.3 Relations entre la machine à états finis FSM de l'entité SSF et les fonctions CCF et de maintenance

L'interface de base entre la machine FSM de l'entité SSF et les fonctions CCF et de maintenance est une interface interne et n'a pas à être normalisée dans l'ensemble CS-1. Toutefois, elle doit être conforme au modèle BCSM défini en 4.2.1.2/Q.1214.

La relation entre le modèle BCSM et la machine FSM de l'entité SSF peut être décrite comme suit dans les cas de tentative d'appel effectuée par un utilisateur ou par la logique de service de réseau intelligent:

- lorsqu'une tentative d'appel est effectuée par un utilisateur et traitée dans un commutateur, un exemplaire du modèle BCSM est créé. Au fur et à mesure du déroulement du modèle BCSM, il rencontre des points de détection (DP) (*detection points*) (voir 4.2/Q.1214). Si un point DP est armé comme point de détection à déclenchement (TDP) (*trigger detection point*), un exemplaire de machine FSM de l'entité SSF est créé;
- lorsqu'une opération de lancement d'appel *InitiateCallAttempt* est reçue en provenance de l'entité SCF, un exemplaire du modèle BCSM est créé, ainsi qu'un exemplaire de la machine FSM de l'entité SSF.

Les fonctions de gestion relatives à l'exécution des opérations reçues de l'entité SCF sont réalisées par l'entité gestion de l'entité SSF (SSME) (*SSF management entity*). L'entité SSME est reliée aux différentes machines FSM de l'entité SSF et au gestionnaire d'accès à l'entité fonctionnelle (FEAM) (*functional entity access manager*). La Figure 10 représente les interfaces de l'entité SSF.

Le gestionnaire d'accès à l'entité fonctionnelle (FEAM) assure les fonctions de maintenance de l'interface de niveau inférieur, à savoir:

- 1) établissement et maintenance des interfaces avec les entités SCF et SRF;
- 2) transmission à l'entité gestion de l'entité SSF (entité SSME) et mise en file d'attente (si nécessaire) des messages reçus en provenance des entités SCF et SRF;
- 3) formatage, mise en file d'attente (si nécessaire) et envoi aux entités SCF et SRF des messages reçus en provenance de l'entité SSME.

L'entité SSME continue à dialoguer avec les entités SCF et SRF pour le compte de toutes les instances de la machine à états finis (FSM) (*finite state machine*) de l'entité SSF. Ces instances apparaissent concurremment et de manière asynchrone, avec les appels, c'est pourquoi il est nécessaire de disposer d'une entité unique chargée de la création, de l'invocation et de la maintenance des machines FSM de l'entité SSF (machine SSF FSM). L'entité SSME exécute notamment les opérations suivantes:

- 1) interprétation des messages entrants provenant d'autres entités fonctionnelles et leur traduction en événements correspondants pour la machine FSM de l'entité SSF;
- 2) traduction des sorties de la machine SSF FSM en messages correspondants pour d'autres entités fonctionnelles;
- 3) exécution des tâches asynchrones (avec traitement de l'appel) relatives aux fonctions de gestion ou de supervision de l'entité SSF. Par exemple, l'entité SSME effectue des traitements non associés aux appels résultant de modifications de filtrage des services, de l'espacement des appels ou du descripteur d'état renvoyé par le contrôle des ressources; elle fournit aussi à l'entité SCF des messages sur l'état des ressources. L'entité SSME sépare ainsi la machine SSF FSM des fonctions d'espacement des appels, de filtrage des services et de contrôle des ressources.

La machine SSF FSM transmet les instructions concernant le traitement des appels aux instances correspondantes de la machine BCSM. Les points DP peuvent être armés dynamiquement comme des points de détection d'événement, ce qui oblige la machine FSM de l'entité SSF à rester active. A un moment donné, l'interaction avec l'entité SCF n'est plus nécessaire et il peut être mis fin à l'instance de la machine SSF FSM tandis que la machine BCSM poursuit le traitement de l'appel si nécessaire. Un point TDP ultérieur de la machine BCSM peut provoquer la création d'une nouvelle instance de la machine SSF FSM pour le même appel.

Conformément à la caractéristique de commande de type mono-utilisateur des éléments de service du réseau intelligent dans l'ensemble CS-1, la machine SSF FSM ne concerne qu'une partie de l'appel fonctionnellement distincte (par exemple, la machine BCSM émettrice ou réceptrice d'un appel à deux correspondants, mais non les deux).

3.1.1.4 Diagramme d'état de l'entité gestion de l'entité SSF (entité SSME)

Le diagramme d'état de l'entité gestion de l'entité SSF (entité SSME) est représenté sur la Figure 11.

L'entité SSME est indépendante des machines FSM individuelles de l'entité SSF.

L'entité SSME passe de l'état IdleManagement (gestion inactive) à l'état non-call associated treatment (traitement non associé aux appels) à la réception de l'une des opérations non associées aux appels énumérés ci-après (transition em1):

RequestCurrentStatusReport
RequestEveryStatusChangeReport
RequestFirstStatusMatchReport,
ActivateServiceFiltering
CallGap
Activity Test

Les opérations ci-dessus peuvent être reçues tant à l'intérieur qu'à l'extérieur d'une transaction de contexte d'appel.

Pendant l'état non-call associated treatment, les événements suivants peuvent se produire:

- si la fonction de filtrage des services est activée, l'entité SSF doit envoyer un réponse de filtrage de service à l'entité SCF: l'entité SSF reste dans le même état (transition em3);
- si la fonction de filtrage des services est activée mais arrive à expiration: l'entité SSME doit passer à l'état IdleManagement (transition em2), sauf si une autre activité de gestion est en cours, auquel cas l'entité SSF reste dans le même état (transition em3);

- si la fonction de compte rendu d'état est activée, l'entité SSF doit lancer une opération StatusReport lorsque l'attribut «contrôle d'état» a la valeur «continuous monitor» (contrôle permanent): l'entité SSF reste alors dans le même état (transition em3);
- si la fonction de compte rendu d'état est activée, l'entité SSF doit lancer une opération StatusReport lorsque l'attribut «contrôle d'état» a la valeur «monitor for the first match» (détection de la première occurrence): l'entité SSF doit alors passer à l'état IdleManagement (transition em2), sauf si une autre activité de gestion est en cours, auquel cas elle doit rester dans le même état (transition em3);
- si la fonction de compte rendu d'état est activée, que l'entité SSF reçoive l'opération Cancel (StatusReport) ou que la fonction de compte rendu d'état arrive à expiration et que l'attribut «contrôle d'état» ait les valeurs «monitor for the first match» (détection de la première occurrence) ou «continuous monitor» (contrôle permanent), l'entité SSME doit alors passer à l'état IdleManagement (transition em2), sauf si une autre activité de gestion est en cours, auquel cas elle doit rester dans le même état (transition em3);
- si la fonction de compte rendu d'état est activée, que l'entité SSF lance l'opération StatusReport et que l'attribut «contrôle d'état» ait la valeur «poll resource status» (interrogation du descripteur d'état des ressources), l'entité SSME doit alors passer à l'état IdleManagement (transition em2), sauf si une autre activité de gestion est en cours, auquel cas elle doit rester dans le même état (transition em3);
- si la fonction d'espacement des appels relative à la durée de temporisation arrive à expiration, l'entité SSME doit passer à l'état IdleManagement (transition em2), sauf si une autre activité de gestion est en cours, auquel cas elle doit rester dans le même état (transition em3);
- si la fonction d'espacement des appels ou de filtrage des services est activée, l'entité SSF peut recevoir une autre opération de type call gap ou ActivateServiceFiltering ayant les mêmes critères d'espacement ou de filtrage: le second «filtre» ou «espace» remplace le premier (transition em3), sauf si la valeur de temporisation est nulle, auquel cas l'entité SSF doit passer à l'état IdleManagement (transition em2). Mais si une autre activité de gestion est en cours, l'entité SSF restera dans le même état (transition em3).

Toutes les autres opérations n'ont aucune influence sur l'état (transition em3); les opérations sont transmises à la machine FSM correspondante de l'entité SSF.

3.1.1.5 Diagramme de transition d'état de l'entité SSF

La Figure 12 représente le diagramme d'état de la partie SSF du point SSP pendant le traitement d'un appel ou d'une tentative d'appel dans le réseau RI.

Chaque état est étudié dans l'un des paragraphes ci-après. Les règles générales applicables à plus d'un état sont traitées ici.

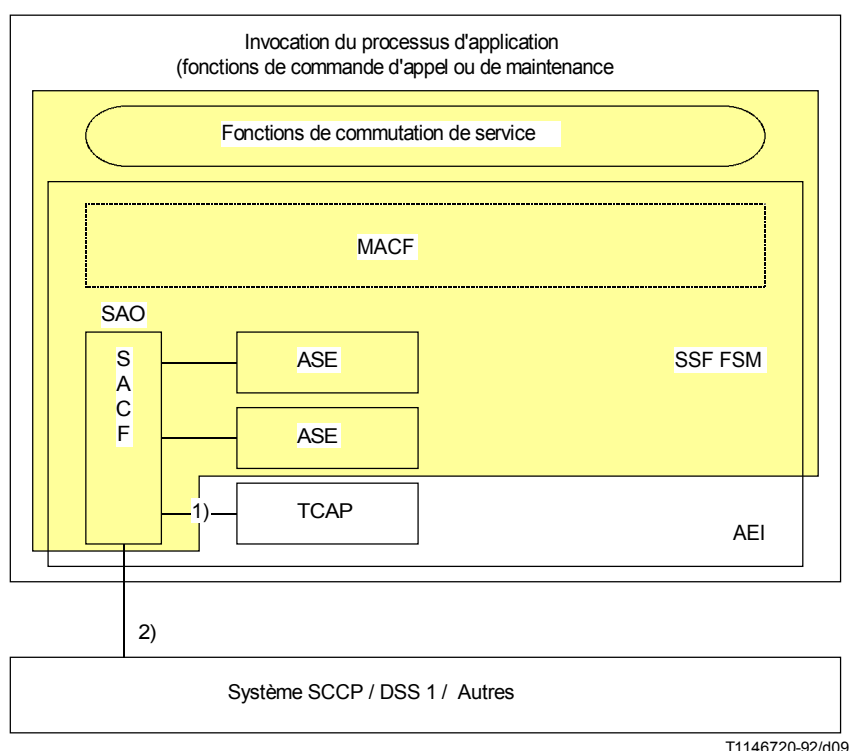
Une composante, ou une séquence de composantes, reçue dans un ou plusieurs messages du sous-système TCAP peut comporter une ou plusieurs opérations et est traitée selon les règles suivantes:

- Les opérations sont traitées dans l'ordre dans lequel elles sont reçues.
- Chaque opération entraîne une transition d'état, indépendamment du nombre d'opérations (une ou plusieurs) reçues dans un message.
- l'entité SSF examine les opérations suivantes dans la séquence. Tant que l'exécution séquentielle de ces opérations ne modifie pas l'état de la machine SSF FSM, elle poursuit l'exécution des opérations (par exemple, RequestReportBCSMEEvent). Si l'une des opérations entraîne une transition d'état, les opérations suivantes doivent être mises en mémoire tampon jusqu'à la fin de l'exécution de l'opération en cours. Dans tous les autres cas, il convient d'attendre un événement entraînant une transition à partir de l'état en cours (un tel événement sera l'achèvement de l'opération en cours d'exécution, ou la réception d'un événement extérieur). On peut en donner l'exemple suivant:

L'entité SSF reçoit les opérations FurnishChargingInformation, ConnectToResource et PlayAnnouncement dans une séquence de composantes à l'intérieur d'un seul message TCAP. A réception de ce message, ces opérations sont exécutées jusqu'à l'opération ConnectToResource incluse, l'entité SSF étant dans l'état waiting for instruction (attente d'instruction). Au moment de l'exécution de l'opération ConnectToResource (et après l'achèvement de l'opération FurnishChargingInformation), la machine SSF FSM doit passer à l'état waiting for end of user interaction (attente de fin de l'interaction utilisateur). L'opération PlayAnnouncement est transmise à l'entité SRF pendant que l'entité SSF est dans l'état waiting for end of user interaction.

- En cas d'erreur pendant le traitement de l'une des opérations de la séquence, la machine SSF FSM traite cette erreur (voir ci-après) et annule toutes les autres opérations restantes de cette séquence.
- Si une opération n'est pas comprise ou sort du contexte (par exemple si elle transgresse les règles de la fonction SACF définies par la machine FSM de l'entité SSF) décrit précédemment, ABANDONNER cette interaction. Par exemple, lorsqu'on applique la machine FSM de l'entité SSF à une machine BCSM émettrice, la réception de l'opération SelectFacility est hors contexte, cela ne s'appliquant qu'à la moitié réceptrice de la machine BCSM.

Quel que soit l'état, si une opération reçue comporte une erreur, les fonctions de maintenance en sont informées et la machine SSF FSM conserve l'état dans lequel elle se trouvait au moment de la réception de l'opération comportant l'erreur; selon la classe de cette opération, l'entité SSF signale cette erreur à l'entité SCF à l'aide de la composante appropriée (Recommandation Q.774).



T1146720-92/d09

- 1) Primitives TC (gestion des transactions)
- 2) Primitives N (Réseau)

AEI Invocation de l'entité application (*application entity invocation*)
 SSF Fonctions commutation de service (*service switching functions*)
 FSM Machine à états finis (*finite state machine*)
 MACF Fonction commande d'association multiple (*multiple association control function*)
 SACF Fonction commande d'association unique (*single association control function*)
 SAO Objet d'association unique (*single association object*)

NOTE – Il est à noter que le modèle FSM de l'entité SSF comprend plusieurs machines à états finis.

FIGURE 9/Q.1218

Modèle fonctionnel d'entité application de l'entité SSF

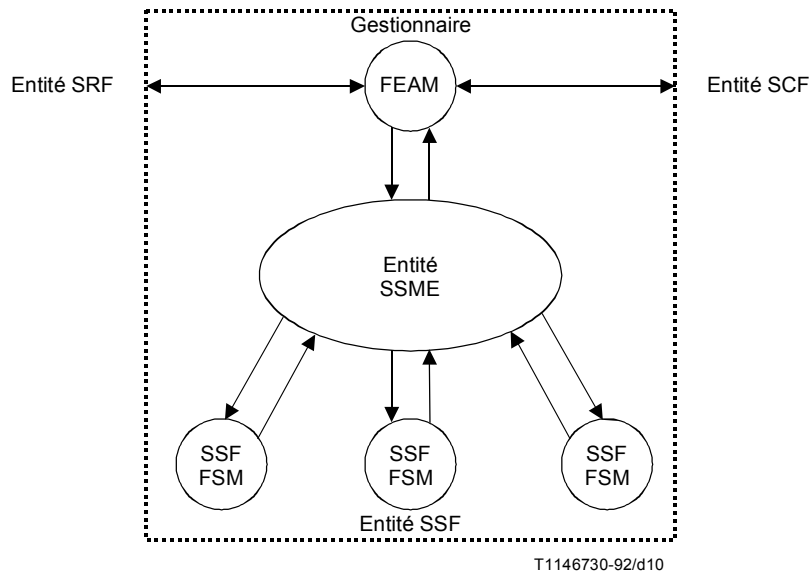


FIGURE 10/Q.1218
Interfaces de l'entité SSF

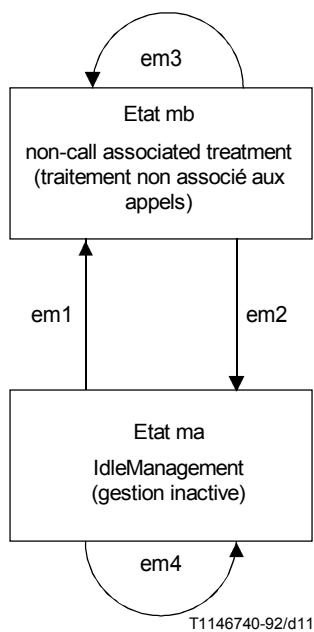
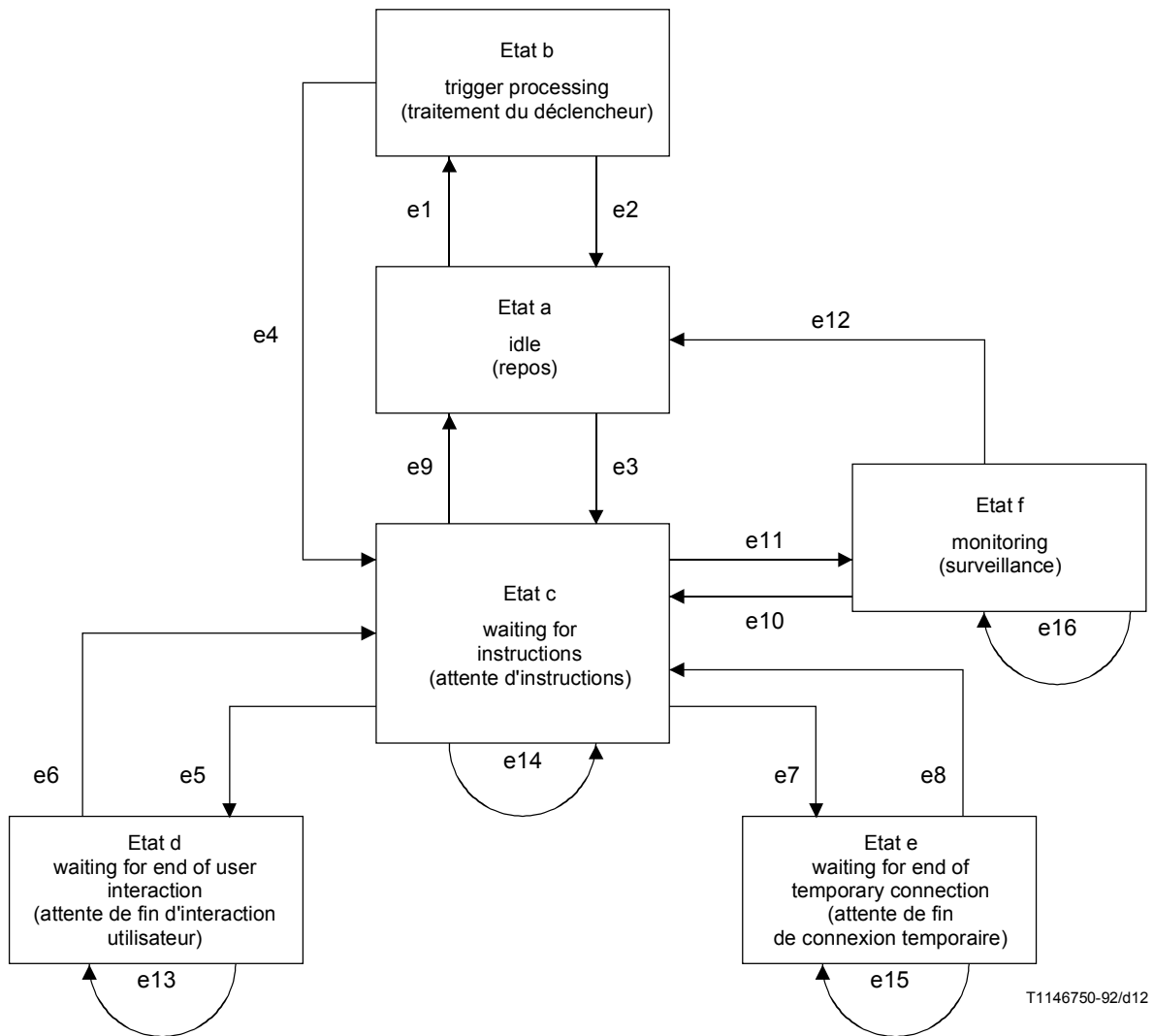


FIGURE 11/Q.1218
Diagramme d'état de l'entité gestion de l'entité SSF (entité SSME)



NOTE – Les transitions abandon et déconnexion ne sont pas représentées.

FIGURE 12/Q.1218
Machine à états finis de l'entité SSF

Quel que soit l'état (à l'exception de l'état idle – repos), si le demandeur abandonne l'appel avant d'avoir reçu une réponse (c'est-à-dire avant le point PIC actif dans la machine BCSM), la machine SSF FSM ordonne à l'entité CCF de libérer l'appel et de s'assurer que toutes les ressources qu'elle avait affectées à cet appel ont été «désaffectées», puis elle poursuit le traitement comme suit¹⁾:

- Si le point DP abandon n'est pas armé et s'il n'y a pas de demande call information request en attente, passer à l'état idle.
- Si le point DP abandon n'est pas armé et s'il y a une demande call information request en attente, envoyer un rapport CallInformationReport puis passer à l'état idle.

¹⁾ Les autres demandes en attente devant être traitées comme la demande CallInformationRequest sont RequestNotificationChargingEvent et ApplyCharging lorsque son paramètre «SendCalculationToSCFIndication» est mis à vrai (true).

- Si le point DP abandon est armé comme point EDP-R, envoyer une opération de type EventReportBCSM ou une opération spécifique de point DP puis passer à l'état waiting for instructions (qu'il y ait ou non une autre demande call information request en attente).
- Si le point DP abandon est armé comme point EDP-N et s'il n'y a pas de demande call information request en attente, envoyer une opération de type EventReportBCSM ou une opération spécifique de point DP puis passer à l'état idle.
- Si le point DP est armé comme point EDP-N et s'il y a une demande call information request en attente, envoyer une opération de type EventReportBCSM ou une opération spécifique de point DP et un rapport CallInformationReport puis passer à l'état idle.

Quel que soit l'état (à l'exception de idle), si l'un des correspondants se déconnecte d'un appel «stable» (c'est-à-dire du point PIC actif de la machine BCSM), la machine SSF FSM doit traiter cet événement comme suit²⁾:

- Si le point DP déconnexion n'est pas armé et s'il n'y a pas de demande call information request en attente, passer à l'état idle s'il ne reste pas d'autre correspondant.
- Si le point DP déconnexion n'est pas armé et s'il y a une demande call information request en attente, envoyer un rapport CallInformationReport et passer à l'état idle.
- Si le point DP déconnexion est armé comme point EDP-R, envoyer une opération de type EventReportBCSM ou une opération spécifique de point DP puis passer à l'état waiting for instructions (qu'il y ait ou non une demande call information request en attente). S'il ne reste alors plus de correspondant, continuer la libération de l'appel.
- Si le point DP déconnexion est armé comme point EDP-N et s'il n'y a pas de demande call information request en attente, envoyer une opération de type EventReportBCSM ou une opération spécifique de point DP puis passer à l'état idle.
- Si le point DP déconnexion est armé comme point EDP-N et s'il y a une demande call information request en attente, envoyer une opération de type EventReportBCSM ou une opération spécifique de point DP et un rapport CallInformationReport puis passer à l'état idle.

L'entité SSF est dotée d'un temporisateur d'application, T_{SSF} , destiné à prévenir une durée excessive de suspension de l'appel et à protéger l'association établie entre les entités SSF et SCF.

Le temporisateur T_{SSF} est déclenché dans les cas suivants:

- Lorsque l'entité SSF envoie une opération de type InitialDP ou AssistRequestInstructions ou une opération spécifique de point DP (voir 3.1.1.5.3, état c: «waiting for instructions»). Dans l'attente de la première réponse de l'entité SCF, le temporisateur T_{SSF} ne peut être réinitialisé qu'une seule fois par une opération reset timer. Après la première réponse, il peut l'être un nombre quelconque de fois.
- Lorsque l'entité SSF reçoit une opération de type HoldCallInNetwork (voir 3.1.1.5.3, état c: «waiting for instructions»). Dans ce cas, l'entité SCF peut réinitialiser le temporisateur T_{SSF} à l'aide de l'opération reset timer un nombre quelconque de fois.
- Lorsque l'entité SSF prend l'état «waiting for end of user interaction» ou «waiting for end of temporary connection» (voir 3.1.1.5.4 et 3.1.1.5.5). Dans ces cas, l'entité SCF peut réinitialiser le temporisateur T_{SSF} à l'aide de l'opération reset timer un nombre quelconque de fois.

Dans les trois cas ci-dessus, le temporisateur T_{SSF} peut avoir trois valeurs différentes définies par l'application.

L'entité SSF doit réinitialiser le temporisateur T_{SSF} lorsqu'elle reçoit ou émet une opération différente des opérations indiquées précédemment. Ce temporisateur n'est pas utilisé dans l'état «monitoring» (voir 3.1.1.5.6).

A la fin de la temporisation T_{SSF} , la machine FSM de l'entité SSF passe à l'état idle, abandonne l'interaction avec l'entité SCF et l'entité CCF active la machine BCSM si possible.

²⁾ Les autres demandes en attente devant être traitées comme la demande CallInformationRequest sont RequestNotificationChargingEvent et ApplyCharging lorsque son paramètre «SendCalculationToSCFIndication» est mis à vrai (true).

Le diagramme d'états de l'entité SSF comprend les transitions (événements) suivantes:

- e1 Passage par un point TDP
- e2 Echech de déclenchement
- e3 Réception de lancement d'appel
- e4 Détection de déclenchement
- e5 Interaction utilisateur demandée
- e6 Interaction utilisateur terminée
- e7 Création d'une connexion temporaire
- e8 Fin d'une connexion temporaire
- e9 Retour à l'état idle depuis l'état waiting for instruction
- e10 Passage par point EDP_R
- e11 Réception d'une instruction d'acheminement
- e12 Dernière notification EDP_N reçue
- e13 Transition nulle de l'état waiting for end of user interaction
- e14 Transition nulle de l'état waiting for instruction
- e15 Transition nulle de l'état waiting for end of temporary connection
- e16 Transition nulle de l'état monitoring
- e17 Abandon (à partir d'un état quelconque) (non représenté dans le diagramme d'états de l'entité SSF)
- e18 Déconnexion (à partir d'un état quelconque) (non représenté dans le diagramme d'états de l'entité SSF)
- e19 Traitement non associé à l'appel (à partir d'un état quelconque) (non représenté dans le diagramme d'états de l'entité SSF)

Le diagramme d'états de l'entité SSF comprend les états suivants:

- Etat a Idle (repos)
- Etat b Trigger processing (traitement du déclencheur)
- Etat c Waiting for instructions (attente d'instructions)
- Etat d Waiting for end of user interaction (attente de fin d'interaction utilisateur)
- Etat e Waiting for end of temporary connection (attente de fin de connexion temporaire)
- Etat f Monitoring (surveillance)

3.1.1.5.1 Etat a: «idle»

La machine FSM de l'entité SSF prend l'état idle dans un certain nombre de conditions décrites ci-après.

La machine FSM de l'entité SSF prend l'état idle lorsqu'elle envoie ou reçoit une primitive ABORT TCAP en raison d'anomalies constatées dans un état quelconque.

La machine FSM de l'entité SSF prend l'état idle en cas d'échec du traitement du point DP pendant l'état trigger processing (transition e2).

La machine FSM de l'entité SSF prend l'état idle dans chacun des cas suivants:

- lorsque l'appel est abandonné ou que toutes les parties d'un appel se déconnectent pendant n'importe quel autre état, dans les conditions stipulées en 3.1.1.5;
- lorsqu'une opération de type initiate call, connect ou proceed call processing est traitée pendant l'état waiting for instructions, qu'aucun point EDP n'est armé et que la surveillance n'est pas activée pour des événements de taxation (transition e9);
- lorsque le temporisateur d'application T_{SSF} s'arrête pendant l'état waiting for instructions (transition e9);

- lorsqu'une opération de type ReleaseCall est traitée pendant l'état waiting for instructions (transition e9) ou monitoring (transition e12);
- lorsque le dernier point EDP-N est franchi pendant l'état monitoring, qu'aucun point EDP-R n'est armé et que la surveillance n'est pas activée pour des événements de taxation (transition e12);
- lorsque le dernier événement de taxation intervient pendant l'état monitoring et qu'aucun point EDP n'est armé (transition e12).

Lorsqu'elle passe à l'état idle, s'il y a une demande call information request en attente (voir 3.1.1.5), l'entité SSF envoie une opération de type CallInformationReport à l'entité SCF avant de revenir à l'état idle. Une fois dans cet état, si la fonction de rapport d'état status reporting est encore active, l'entité SSF la désactive et toutes les réponses en attente devant être envoyées à l'entité SCF sont rejetées.

Pendant cet état, les événements associés aux appels, énumérés ci-après, peuvent intervenir:

- l'entité CCF indique la rencontre d'un point TDP armé relatif à une possibilité de tentative d'appel/service dans le réseau RI: dans ce cas, la machine FSM de l'entité SSF passe à l'état trigger processing (transition e1);
- l'entité SCF envoie un message concernant une nouvelle transaction contenant une opération de type InitiateCallAttempt: dans ce cas, la machine FSM de l'entité SSF passe à l'état waiting for instructions (transition e3).

Toute autre opération reçue de l'entité SCF pendant que l'entité SSF 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 (Recommandation Q.774).

3.1.1.5.2 Etat b: «trigger processing»

Après la détection d'un déclenchement concernant un point TDP armé de la machine BCSM, la machine FSM de l'entité SSF est activée et passe de l'état idle à l'état trigger processing (transition e1).

Pendant cet état, l'entité SSF/CCF doit:

- effectuer les actions de traitement du point DP spécifiées en 4.2.2.5/Q.1214
 - a) vérifier si les mécanismes d'espacement des appels ou de filtrage des services sont actifs;
 - b) vérifier l'accès à l'entité SCF;
 - c) déterminer si les critères de point DP sont respectés;
 - d) traiter les interactions d'éléments de service;
- réunir et contrôler les paramètres nécessaires pour envoyer une opération de type InitialDetectionPoint ou une opération spécifique de point DP³⁾ à l'entité SCF:
 - 1) en cas de réussite des actions de traitement du point DP et s'il s'agit d'un point TDP-N, envoyer une opération générique de type InitialDetectionPoint ou une opération spécifique de point DP³⁾ à l'entité SCF, selon les indications fournies par le traitement de point DP, et revenir à l'état idle (transition e2);
 - 2) en cas de réussite des actions de traitement du point DP et s'il s'agit d'un point TDP-R, envoyer une opération générique de type InitialDetectionPoint ou une opération spécifique de point DP³⁾ à l'entité SCF, selon les indications fournies par le traitement de point DP et passer à l'état waiting for instructions (transition e4);
 - 3) en cas d'échec du traitement du point DP, revenir à l'état idle (transition e2). Le traitement du point DP échoue dans les cas suivants:
 - si l'élément de service CallGapping est activé: la machine FSM de l'entité SSF ordonne à l'entité CCF de mettre fin à l'appel à l'aide du traitement approprié;
 - si l'élément de service Service Filtering est activé: l'appel est compté (à la demande) et la machine FSM de l'entité SSF ordonne à l'entité CCF de mettre fin à l'appel à l'aide du traitement approprié;

³⁾ Les opérations *spécifiques de point DP* sont les suivantes (voir l'article 2): TAnswer, TDisconnect, TermAttemptAuthorized, TMidCall, TNoAnswer, AnalysedInformation, TCalledPartyBusy, CollectedInformation, OAnswer, OCalledPartyBusy, ODisconnect, OMidCall, ONoAnswer, OriginationAttemptAuthorized, RouteSelectFailure.

- si l'un des critères de déclenchement n'est pas respecté: la machine FSM de l'entité SSF renvoie la commande d'appel à l'entité CCF;
- si l'appel est abandonné: l'entité SSF renvoie la commande d'appel à l'entité CCF et poursuit le traitement selon la procédure décrite en 3.1.1.5;
- s'il n'y a pas assez d'informations pour effectuer le traitement (notamment en ce qui concerne les plans de numérotage ouverts): la machine FSM de l'entité SSF ordonne à l'entité CCF de mettre fin à l'appel à l'aide d'un traitement d'achèvement normalisé;
- si l'entité SCF destinataire n'est pas accessible: la machine FSM de l'entité SSF ordonne à l'entité CCF d'acheminer l'appel si possible (acheminement par défaut vers une annonce terminale, par exemple);
- s'il existe une relation de commande pour l'appel: l'entité SSF renvoie la commande d'appel à l'entité CCF.

3.1.1.5.3 Etat c: «waiting for instructions»

L'accès à cet état se fait soit à partir de l'état trigger processing, comme indiqué ci-dessus (transition e4), soit directement à partir de l'état idle au moment de la réception, à l'entité SSF, d'une primitive OPEN contenant une opération de type initiate call provenant de l'entité SCF (transition e3). Dans ce dernier cas, l'entité SSF passe immédiatement à l'état waiting for instructions, puis agit selon la procédure indiquée par le contenu de l'opération initiate call et de toutes les autres opérations reçues.

Pendant cet état, la machine FSM de l'entité SSF attend une instruction de l'entité SCF; le traitement de l'appel est suspendu. Un temporisateur d'application (T_{SSF}) doit être initialisé lors du passage à cet état.

Pendant cet état, les événements suivants peuvent se produire:

- L'utilisateur compose des chiffres supplémentaires (dans le cas de plans de numérotage ouverts): l'entité SSF doit les enregistrer.
- L'utilisateur abandonne l'appel ou se déconnecte (voir 3.1.1.5).
- Le temporisateur d'application T_{SSF} arrive à expiration: 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.
- 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.

Les opérations suivantes peuvent être reçues de l'entité SCF et traitées par l'entité SSF sans entraîner de transition d'état (transition e14):

HoldCallInNetwork
 RequestReportBCSMEEvent,
 RequestNotificationChargingEvent
 ResetTimer
 FurnishChargingInformation
 ApplyCharging
 CallInformationRequest
 SendChargingInformation

L'opération suivante peut être reçue de l'entité SCF et traitée par l'entité SSF en entraînant une transition à l'état waiting for end of user interaction (transition e5):

ConnectToResource

L'opération suivante peut être reçue de l'entité SCF et traitée par l'entité SSF en entraînant une transition à l'état waiting for end of temporary connection (transition e7):

EstablishTemporaryConnection

Les opérations suivantes peuvent être reçues de l'entité SCF et traitées par l'entité SSF en entraînant une transition à l'état monitoring (en présence de points EDP armés ou de demande de rapport CallInformationReport) (transition e11) ou idle (transition e9):

Connect
CollectInformation
AnalyseInformation
SelectRoute
SelectFacility
Continue
ReleaseCall

Une opération de type InitiateCallAttempt, si elle est reçue dans l'état idle, doit être traitée dans l'état waiting for instructions (transition e3).

Pendant le traitement des opérations indiquées ci-dessus, toutes les informations nécessaires relatives au traitement de l'appel sont fournies à l'entité de commande d'appel (entité CCF).

Toute autre opération reçue dans cet état doit être traitée selon les règles générales énoncées en 3.1.1.5.

Les opérations de type call abandon et disconnect doivent être traitées selon les règles générales énoncées en 3.1.1.5.

3.1.1.5.4 Etat d: «waiting for end of user interaction»

L'entité SSF passe à cet état à partir de l'état waiting for instructions (transition e5) à la réception de 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 e13).
- Réception d'une opération valide et correcte de retransmission par entité SRF-SCF (c'est-à-dire d'une opération de type SpecializedResourceReport ou ReturnResult from prompt and collect user information); dans ce cas, l'opération est transférée à l'entité SCF. La machine FSM de l'entité SSF reste dans l'état waiting for end of user interaction (transition e13).
- 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 agit selon l'opération reçue, comme décrit ci-dessus.

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 e13):

RequestReportBCSMEEvent
RequestNotificationChargingEvent
ResetTimer
FurnishChargingInformation
ApplyCharging
CallInformationRequest
SendChargingInformation

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.

L'opération call abandon doit être traitée selon les règles générales énoncées en 3.1.1.5.

3.1.1.5.5 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.

Pendant cet état, les événements suivants peuvent se produire:

- Réception d'une opération de type DisconnectForwardConnection: l'entité SSF passe à l'état waiting for instructions (transition e8).
- 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 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.

Les opérations suivantes peuvent être reçues de l'entité SCF et traitées par l'entité SSF sans entraîner de transition d'état (transition e15):

RequestReportBCSMEvent
RequestNotificationChargingEvent
ResetTimer
FurnishChargingInformation
ApplyCharging
CallInformationRequest
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.

L'opération call abandon doit être traitée selon les règles générales énoncées en 3.1.1.5.

3.1.1.5.6 Etat f: «monitoring»

L'entité SSF passe 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 ReleaseCall ou lorsqu'elle traite une opération de type InitiateCallAttempt alors qu'un ou plusieurs point(s) EDP est (sont) armé(s) ou qu'il y a une demande call information request 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 doit être signalée à l'entité SCF en envoyant une opération de type EventReportBCSM ou une opération spécifique de 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 une demande call information request en attente, elle 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 demande call information request en attente.
- Une demande de point EDP doit être signalée à l'entité SCF en envoyant une opération de type EventReportBCSM ou une opération spécifique de point DP; la machine FSM de l'entité SSF doit passer à l'état waiting for instructions (transition e10).

- 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 peut 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 agit conformément à cette opération, comme décrit précédemment.

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):

RequestReportBCSMEvent
RequestNotificationChargingEvent
CallInformationRequest

L'opération ReleaseCall peut être reçue de l'entité SCF et traitée par l'entité SSF, entraînant un passage à l'état idle (transition e12). S'il y a une demande call information request en attente, l'entité SSF envoie un rapport CallInformationReport à l'entité SCF.

Toute autre opération reçue dans cet état doit être traitée conformément aux règles générales énoncées en 3.1.1.5.

Les opérations call abandon et disconnect doivent être traitées 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.

Le diagramme d'états de la machine de transfert de tâche à une entité SSF assistante comporte les transitions (événements) suivantes (voir la Figure 13):

- ea1 Détection de transfert de tâche/d'assistance
- ea2 Echec 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

Le diagramme d'états de la machine de transfert de tâche à une entité SSF assistante 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)

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, réception d'une indication DisconnectForwardConnection provenant 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 (Recommandation Q.774).

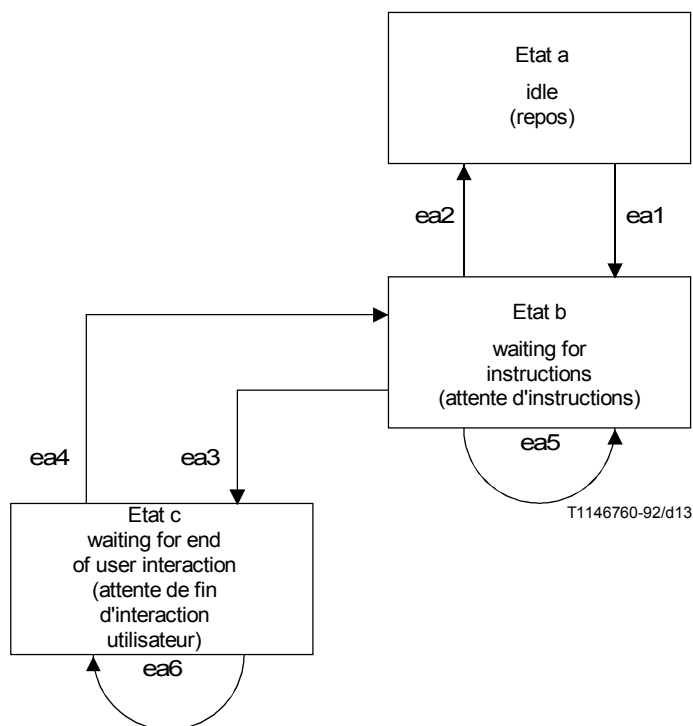


FIGURE 13/Q.1218

Machine à états finis de transfert de tâche à une entité SSF assistante

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).

Dans cet état, l'entité SSF envoie une opération de type AssistRequestInstructions à l'entité SCF et 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.
- Réception d'une déconnexion de canal support: la machine FSM passe à l'état idle (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

L'opération suivante peut être reçue de l'entité SCF et traitée par l'entité SSF assistante en entraînant une transition à l'état waiting for end of user interaction (transition ea3):

ConnectToResource

L'opération suivante peut être reçue de l'entité SCF et traitée par l'entité SSF assistante en entraînant une transition à l'état idle (transition ea2):

ReleaseCall

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).
- L'entité SRF indique à l'entité SSF la fin de l'interaction utilisateur en lançant une déconnexion: la machine FSM de l'entité SSF revient à l'état waiting for instructions (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 agit selon l'opération reçue, comme décrit ci-dessus.

L'opération suivante peut être reçue en provenance de l'entité SCF et traitée 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 dans cet état, entraînant une transition à l'état waiting for 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.

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 Considérations générales

Ce paragraphe contient la définition des procédures relatives à l'entité application (AE) (*application entity*) de l'entité SCF (AE-SCF) applicables à l'interface SCF-SSF/SRF/SDF. Ces procédures font appel 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 l'un ou plusieurs éléments du service application (ASE) appelés utilisateurs-TC. Les paragraphes ci-après définissent les règles applicables l'élément ASE 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 des limites aux programmes de logique de services.

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 champ d'application de la présente Recommandation est limité à la zone ombrée de la Figure 14.

Les interfaces représentées à la Figure 14 font appel aux primitives de l'entité ASE de l'utilisateur du gestionnaire de transaction spécifiées dans la Recommandation Q.771 [interface (1)] et aux primitives de couche réseau (N-) spécifiés dans la Recommandation Q.711 [interface (2)]. Le fonctionnement et les paramètres associés du protocole d'application du réseau intelligent (protocole INAP) sont définis en 2.

3.1.2.3 Relation entre la machine SCF FSM et les fonctions de maintien avec les programmes de logique de service

La première interface entre la machine SCF FSM et les programmes de logique de service/fonctions de maintenance 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 où d'un appel déclenché par un utilisateur final ou d'un appel déclenché par la logique de service du RI):

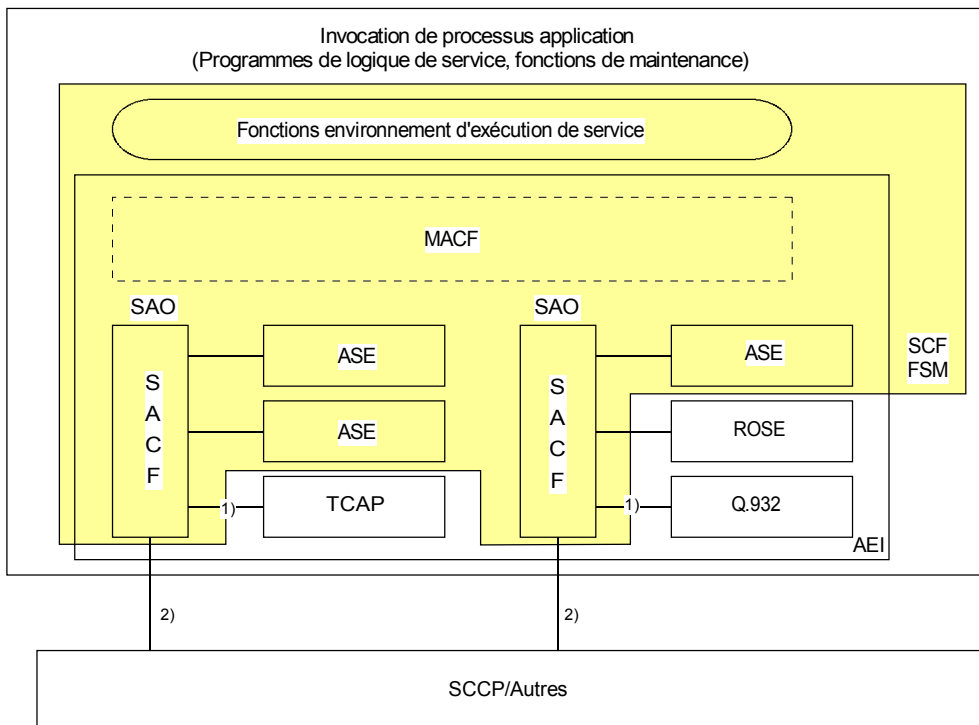
- Lorsqu'une demande de traitement d'appel RI est reçue en provenance de l'entité SSF, une instance de modèle d'état SCF (SCSM) est créée et le programme de logique de service associé est appelé.
- Lorsque le déclenchement d'un appel est demandé par la logique de service, une instance de modèle SCSM est créée.

Dans chacun des cas, la machine FSM SCF gère l'interaction avec la machine SSF FSM (et la machine SRF FSM et la machine 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 exécutées par l'entité de gestion SCF (entité SCME). L'entité SCME assure l'interface avec différents modèles d'état d'appel SCF (SCSM) et le gestionnaire d'accès d'entité fonctionnelle (gestionnaire FEAM). La structure de la machine SCF FSM est représentée à la Figure 15.

Dans ce qui suit on donne une description systématique des aspects de procédure de l'interface entre l'entité SCF et les autres entités fonctionnelles, avec pour principal objet de spécifier le bon ordre des opérations plutôt que les capacités fonctionnelles des entités. Par conséquent, on ne décrit qu'un sous-ensemble des capacités fonctionnelles de l'entité SCF.

Le modèle de procédure associe un modèle SCSM à chaque demande émanant de l'entité SSF. Le modèle SCSM maintient les dialogues avec les entités SSF, SRF et SDF pour la logique de service.



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*)

¹⁾ Primitives TC ou primitives Q.932.

²⁾ Primitives N.

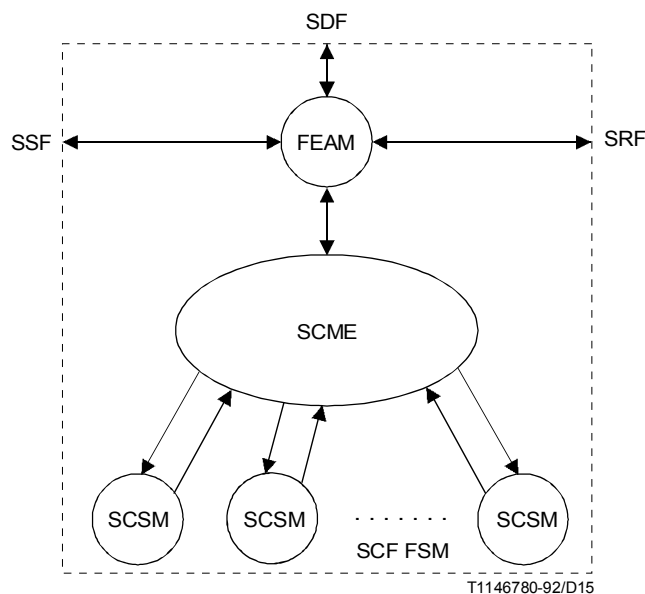
NOTE – La machine SCF FSM comporte plusieurs machines FSM.

FIGURE 14/Q.1218

Modèle fonctionnel d'entité application de la fonction de commande de service (SCF AE)

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 de la machine SCF FSM. Cette entité est appelée entité de gestion SCF (SCME). Outre les tâches ci-dessus, l'entité SCME maintient les dialogues avec les entités SSF, SDF et SRF pour toutes les instances du modèle SCSM. En particulier, l'entité 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 toutes les tâches asynchrones (avec le traitement d'appel) (par exemple la commande de flux). Il appartient à l'entité SCME de détecter les surcharges nodales et d'envoyer l'indication de surcharge (par exemple, espacement automatique des appels) à l'entité SSF pour exercer la commande de flux sur les demandes. Les autres activités sont le traitement associé de type non-appel dû aux modifications du filtrage du service, de l'espacement des appels et de la surveillance des ressources et également la fourniture de messages relatifs à l'état des ressources à l'entité SSF; et
- 4) prend en charge les interactions durables entre les SCF et les autres entités fonctionnelles.



| | |
|------|--|
| FEAM | Gestionnaire d'accès aux entités fonctionnelles (<i>functional entity access manager</i>) |
| SCME | Entité de gestion, entité SCF (<i>SCF management entity</i>) |
| SCSM | Modèle d'état d'appel de l'entité SCF (<i>SCF call state model</i>) |

FIGURE 15/Q.1218
Structure de la machine FSM SCF

En définitive, le gestionnaire d'accès aux entités fonctionnelles (gestionnaire FEAM) soulage l'entité SCF des fonctions d'interface de niveau inférieur. Les fonctions du gestionnaire FEAM sont:

- 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 à l'entité SCME; et
- 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, SRD 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 file d'attente est assurée par l'entité SCF.

3.1.2.4 Diagramme de transition d'état partiel de l'entité de gestion SCF (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.

L'entité SCME exécute les opérations suivantes:

- Request current status report (demande de rapport d'état actuel);
- Request every status change report (demande de rapport de toute modification d'état);
- Request first status match report (demande de rapport de la première coïncidence d'état);

- Cancel status report (y compris l'identificateur Invoke ID précédemment utilisé pour les opérations request first status match report ou request every state change report);
- Status report (rapport d'état);
- Activate service filtering (activer le filtrage de service);
- Service filtering report (rapport de filtrage de service);
- Call gap (espacement d'appel); et
- Activity test (test d'activité).

L'émission des opérations call gap et activity test 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 indiquées ci-dessus ne modifient pas l'état de l'entité SCME, ces opérations sont transmises au modèle SCSM concerné.

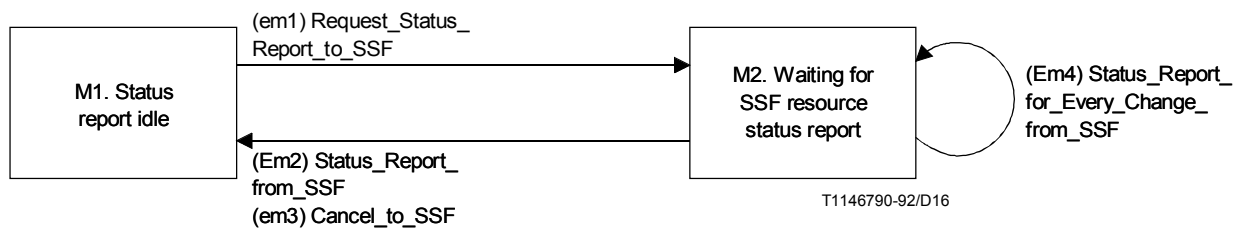


FIGURE 16/Q.1218

Machine FSM de rapport d'état (*status report*) de l'entité SCME

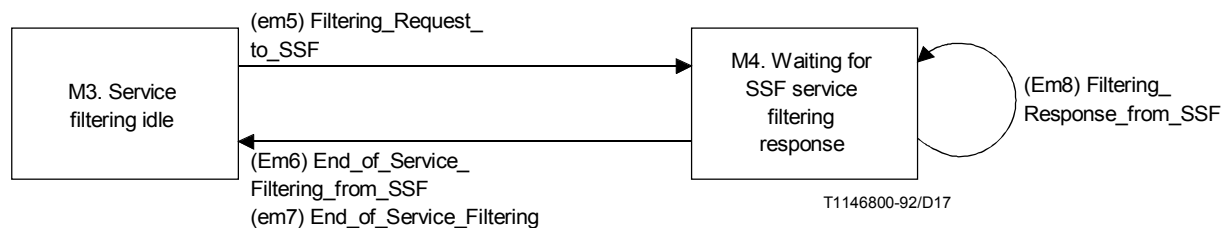


FIGURE 17/Q.1218

Machine FSM de filtrage du service (*service filtering*) de l'entité SCME

3.1.2.4.1 Etat M1: «status report idle»

On considère que l'événement⁴⁾ ci-après se trouve 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 request current status report ou request first status match report précédemment émise vers l'entité SSF. Cet événement provoque le passage à l'état M1 status report idle;
- (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 par la transmission d'une opération cancel status report à l'intention de l'entité SSF. Cet événement a lieu seulement pour les opérations suivantes précédemment émises: request first status match report ou request every status change report. Cet événement provoque le passage à l'état M1 status report idle; et
- (Em4) Status_Report_for_Every_Change_from_SSF: événement externe produit par la réception de la réponse à l'opération 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, 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 activate service filtering. Cet événement provoque le passage à l'état M4, 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 service 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 service. 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 service 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, waiting for SSF service filtering response.

Quand l'opération service filtering est active, une autre opération service filtering peut être adressée à l'entité SSF utilisant les mêmes critères de filtrage; ce second filtre se substitue au précédent.

3.1.2.4.5 L'objet commande de ressource (resource control object ou objet RCO)

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.

⁴⁾ Tous les événements sont numérotés, et le numéro d'un événement est préfixé avec soit la lettre «E» (pour les événements externes) ou par la lettre «e» (pour les événements internes), et ce préfixe est inclus entre parenthèses au début de chaque nom d'événement.

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; et
- 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); et
- 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; et
- 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 d'état du modèle SCSM est représenté à la Figure 18 pour ce qui est des procédures relative à la partie machine SCF FSM de l'entité physique SCP/AD/SN au cours du traitement d'un appel RI. Chaque état est examiné dans les sous-paragraphes ci-après. Chaque état sauf idle, SDF request idle et waiting for SDF response, dispose de sous-machines FSM composées de sous-états.

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 et les fonctions de maintenance en sont informés, et le modèle SCSM reste dans le même état. Selon la classe de l'opération, l'erreur peut être signalée aux entités SSF, SRF ou SDF (voir la 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 et il en informe le programme de logique de service et revient à l'état idle (Repos). Dans ce cas, toutes les ressources attribuées à cet appel, y compris celles qui sont nécessaires au dialogue associées avec d'autres fonctions, doivent être retirées. Pour simplifier le diagramme, ces transitions ne sont pas représentées dans les figures.

Lorsque le programme de logique de service demande des informations relatives à l'appel, le modèle SCSM transmet l'opération call information request à l'entité SSF et l'opération call information report est mise en attente.

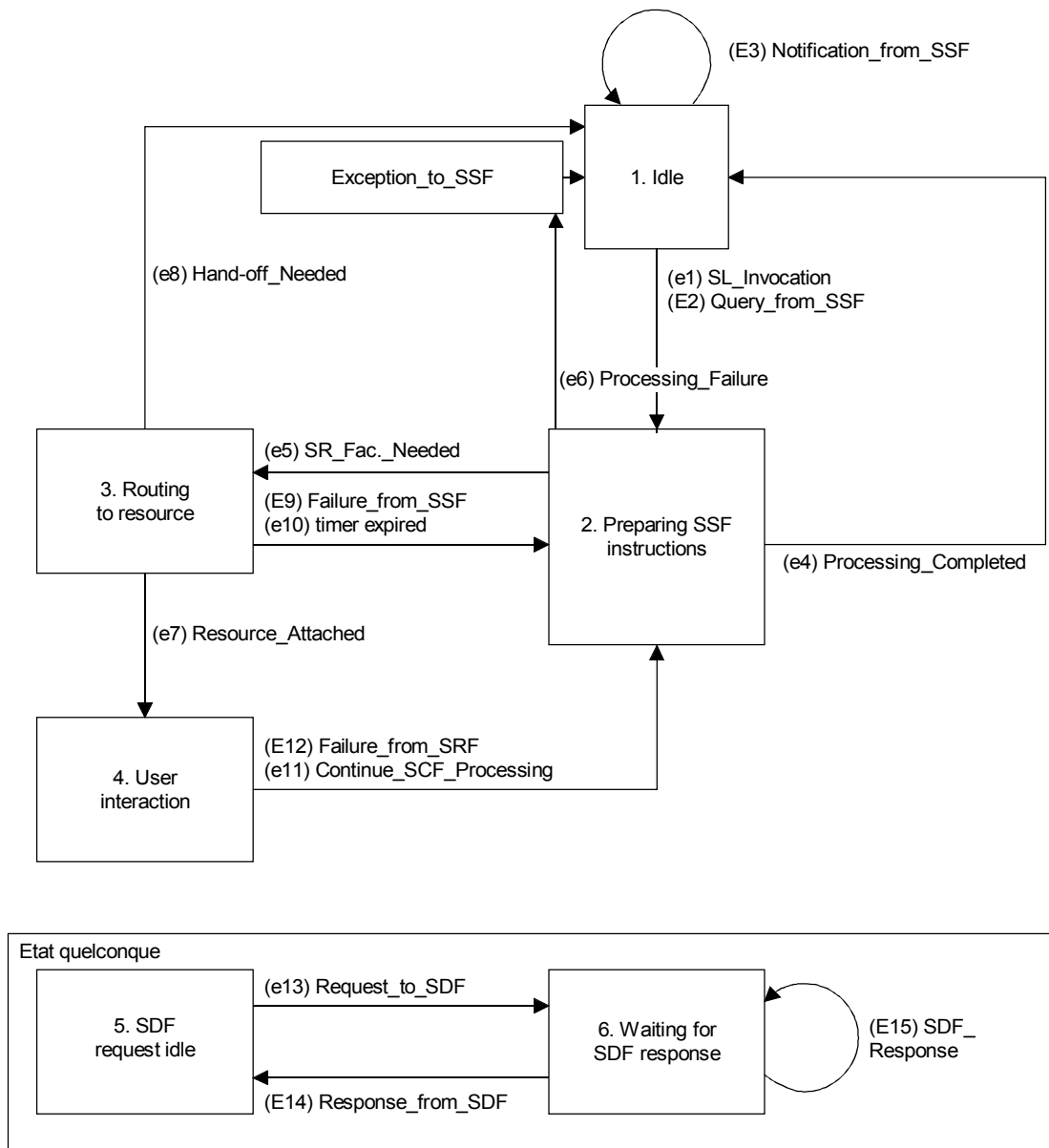
L'opération cancel (pour l'opération call information request) peut être seulement envoyée à l'entité SSF lorsque l'opération call information report est en attente.

Dans tout état (sauf l'état idle), le modèle SCSM peut recevoir l'opération call information report⁵⁾ provenant de l'entité SSF, lorsque l'opération call information report est mise en attente.

A partir d'un état (sauf l'état idle), lorsque l'opération call information report 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 réception de l'opération call information report.

Les règles générales pour une composante ou une séquence de composantes envoyée dans un ou plusieurs messages du sous-système TCAP qui peut inclure une seule ou plusieurs opérations, sont spécifiées en 3.1.1.5 diagramme de transition d'état de l'entité SSF (elles ne sont pas décrites ici).

⁵⁾ Toutes les autres demandes en attente qui doivent être traitées de la même façon que la demande call information request sont l'événement request notification charging event et/ou la demande apply charging lorsque le paramètre «send calculation to SCF indication» est mis à Vrai (True).



T1146810-92/D18

FIGURE 18/Q.1218

Machine à états finis du modèle SCSM

Le modèle SCSM dispose d'un temporisateur d'application, le temporisateur $T_{SCF-SSF}$, qui a pour objet de réinitialiser le temporisateur T_{SSF} utilisé pour éviter d'avoir des durées de suspension d'appel trop importants et pour conserver l'association entre l'entité SSF et l'entité SCF.

Le temporisateur $T_{SCF-SSF}$ est initialisé dans les cas suivants:

- sur réception par l'entité SCF d'une opération initial DP, DP specific request instructions ou assist request instructions (voir 3.1.2.5.2.1 état 2.1: «Preparing SSF instructions», et 3.1.2.5.2.2.1 état 2.2.1: «Preparing SSF instructions»). Dans ce cas ce temporisateur est réinitialisé lorsque la première demande, autre que l'opération reset timer est envoyée à l'entité SSF. A la fin de la temporisation $T_{SCF-SSF}$, l'entité SCF peut réinitialiser une fois le temporisateur T_{SSF} , en utilisant l'opération reset timer, et également réinitialiser le temporisateur $T_{SCF-SSF}$. A la fin de la deuxième temporisation $T_{SCF-SSF}$, le modèle SCSM informe le programme de logique de service et les fonctions de maintenance, et le modèle SCSM passe à l'état idle;
- sur envoi par l'entité SCF d'une opération hold call in network (voir 3.1.2.5.2.2.2 état 2.2.2: «queueing»). Dans ce cas, à la fin de la temporisation $T_{SCF-SSF}$, l'entité SCF peut réinitialiser le temporisateur T_{SSF} en utilisant l'opération reset timer sur un nombre quelconque de temporisateurs; et

- passage de l'entité SCF passe à l'état «Waiting for Assist Request Instructions» ou l'état «user interaction» (voir 3.1.2.5.3.2 et 3.1.2.5.4). Dans ces cas, à la fin de la temporisation $T_{SCF-SSF}$, l'entité SCF peut réinitialiser le temporisateur T_{SSF} en utilisant l'opération reset timer un nombre quelconque de fois.

Dans trois cas, la temporisation $T_{SCF-SSF}$ peut avoir trois 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} .

Lorsqu'elle reçoit ou envoie une autre opération, l'entité SCF doit réinitialiser le temporisateur $T_{SCF-SSF}$. Ce temporisateur n'est pas utilisé dans l'état «Waiting for Notification or Request» (voir 3.1.2.5.2.3).

Le modèle SCSM dispose également d'un temporisateur d'application, le temporisateur $T_{ASSIST/HAND-OFF}$, dont l'objet est d'empêcher un temps de suspension assistance/sans assistance excessif. Le modèle SCSM arme le temporisateur $T_{ASSIST/HAND-OFF}$ lorsque le modèle SCSM envoie une opération establish temporary connection ou select route/connect avec une opération ID de corrélation. Ce temporisateur est arrêté lorsque le modèle SCSM reçoit une opération assist request instruction provenant de l'entité SSF assistante/automatisée. A la fin de la temporisation $T_{ASSIST/HAND-OFF}$, le modèle SCSM informe la logique de service et les fonctions de maintenance et le modèle SCSM passe à l'état idle.

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 deux catégories suivantes:

- 1) opérations relatives au traitement d'appel; et
- 2) opérations relatives au traitement des non-appels.

Ces dernières opérations peuvent être adressées à l'entité SSF dans une série de messages du sous-système TCAP ou dans une séquence composante, alors que la première ne peut être envoyée à raison d'une seule à la fois, séparée par des messages EDP-R reçus par le modèle SCSM. Les opérations de traitement d'appel sont les suivantes:

- Analyse information (analyser l'information);
- Collect information (collecter les informations);
- Connect (connecter);
- Connect to resource (connecter à la ressource);
- Continue (suite);
- Disconnect forward connection (déconnexion de la connexion vers l'avant);
- Establish temporary connection (établir une connexion temporaire);
- Initiate call attempt (déclencher une tentative d'appel);
- Release call (libérer l'appel);
- Select facility (sélectionner la facilité); et
- Select route (sélectionner le trajet).

Les opérations de traitement des non-appels comprennent le reste des opérations à l'interface SCF-SSF et 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 sous-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» (repos)

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 initiate call attempt à l'entité SSF;
- (E2) Query_from_SSF: événement externe, déclenché par la réception d'une des opérations suivantes:
 - Assist Request Instructions (pour le cas du service sans assistance); et
 - DP-Specific Request Instructions⁶⁾.

⁶⁾ Les opérations DP-specific request instructions sont les suivantes (voir l'article 2): TAnswer, TDisconnect, TemAttemptAuthorized, TMidCall, TNoAnswer, AnalysedInformation, CalledParty Busy, CollectedInformation OAnswer, OCalledPartyBusy, ODisconnect, OMidCall, ONoAnswer, OriginationAttemptAuthorized, RouteSelectFailure.

Ces deux événements provoquent le passage à l'état 2, preparing SSF instructions.

- (E3) Notification_from_SSF: Événement externe déclenché par la réception d'une opération initial DP ou DP specific request instructions 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: «preparing SSF instructions» (préparation des instructions pour l'entité 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 terminé le traitement des instructions à l'intention de 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 ou 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 traitement⁷⁾ d'exception approprié et un retour à l'état 1 idle.

Pour décrire plus avant les procédures associées à cet état, celui-ci est divisé en trois sous-états, qui sont décrits dans les sous-paragraphes qui suivent. Cette subdivision est illustrée à la Figure 19.

3.1.2.5.2.1 Etat 2.1: «Preparing SSF instructions» (préparation des instructions pour l'entité SSF)

Dans l'état 2.1 preparing SSF instructions, 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 que les événements ci-après sont dans cet état:

- (e2.1) Non-Call_Processing_Instructions: événement interne produit par la logique de service lorsqu'il est nécessaire d'envoyer une telle opération à l'entité SSF. Cet événement provoque l'émission d'une ou plusieurs opérations suivantes à l'intention de l'entité SSF:
 - Apply charging (appliquer la taxation);
 - Call information request (demande d'information relative à l'appel);
 - Cancel (annulation de l'opération Call Information Request);
 - Furnish charging information (fournir les informations relatives à la taxation);
 - Request report BCSM event (demande de rapport d'événement BCSM);
 - Request notification charging event (demande de notification d'événement de taxation);
 - Reset timer (réinitialisation du temporisateur); et
 - Send charging information (envoyer l'information de taxation).

Cet événement provoque le retour à l'état 2.1 Preparing SSF instructions.

- (e2.2) SR_Facilities_Needed: événement interne, produit par la logique, on doit utiliser l'entité SRF. Cet événement correspond à l'événement SCSM (e5).
- (e2.3) Call_Processing_Instruction_Ready (surveillance⁸⁾ non nécessaire): événement interne produit par la logique de service lorsque l'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'opération particulière call information report, event notification charging or apply charging report. Elle provoque l'émission des opérations suivantes à l'intention de l'entité SSF:
 - Analyse information (analyser l'information);
 - Collect information (collecter l'information);

⁷⁾ 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 appropriée à l'intention de l'entité SSF.

⁸⁾ Comprenant les opérations call information report, notification charging event et/ou apply charging report lorsque le paramètre indicateur «envoyer le calcul au SCP» pour l'opération apply charging est mis à «vrai».

- Connect (connecter);
- Continue (continuer);
- Release call (libérer l'appel);
- Select facility (sélectionner la facilité); et
- Select route (sélectionner le trajet).

Cet événement correspond à l'événement SCSM (e4).

- (e2.4) Call_Processing_Instruction_Ready (surveillance⁹⁾ nécessaire): 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 call information report, event notification charging ou apply charging report et qu'il est nécessaire d'émettre une telle demande). Elle provoque l'émission des opérations suivantes à l'intention de l'entité SSF:
 - Analyse information;
 - Collect information;
 - Connect;
 - Continue;
 - Release call;
 - Select facility; et
 - Select route.

En outre, une ou plusieurs des opérations ci-après peuvent être émises à l'entité SSF avant les opérations précitées:

- Apply charging;
- Call information request;
- Cancel (pour call information request);
- Furnish charging information;
- Request report BCSM event;
- Request notification charging event; et
- Send charging information.

Cet événement provoque le passage à l'état 2.3, waiting for notification or request.

- (e2.5) Ready_for_Queueing_Processing: 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: événement interne correspondant à l'événement SCSM (e6) Processing_Failure.

3.1.2.5.2.2 Etat 2.2 «Queueing FSM» (Mise en file d'attente FSM)

Lorsque l'entité SCF traite une 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 occupée¹⁰⁾. Cette 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 au 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:

- Hold call in network (mettre en garde l'appel dans le réseau);
- Apply charging;
- Call information request;
- Cancel (pour call information request);

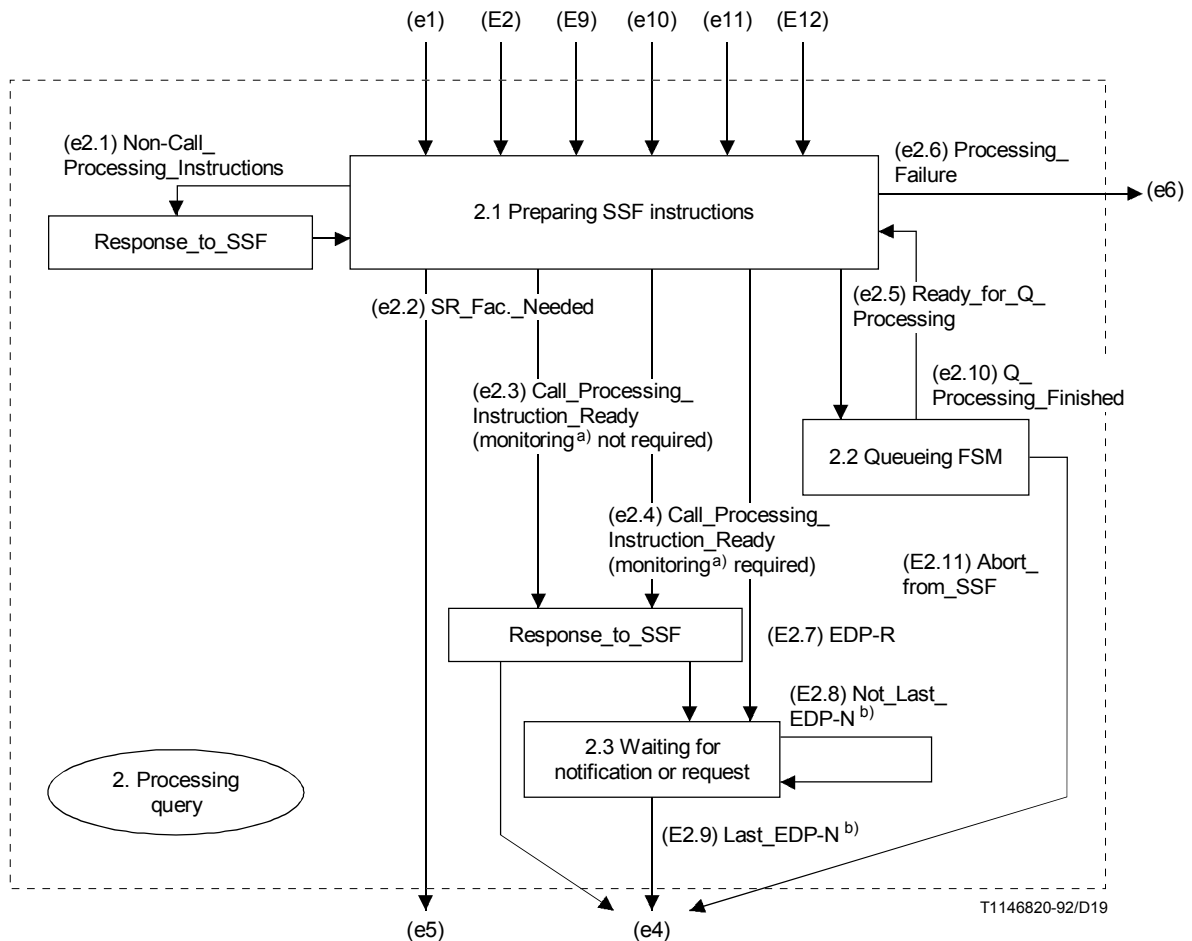
⁹⁾ Comprenant les opérations call information report, notification charging event et/ou apply charging report lorsque le paramètre indicateur «envoyer le calcul au SCP» pour l'opération apply charging est mis à «vrai».

¹⁰⁾ La façon dont l'état des ressources est maintenu est décrite en 3.1.2.4.5.

- Furnish charging information;
- Request report BCSM event;
- Request notification charging event;
- Reset timer; et
- Send charging information.

On considère que les événements sont dans cet état:

- (e2.10) Queueing_Processing_Finished: 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, preparing SSF instructions.
- (E2.11) Abort_from_SSF: événement externe déclenché par la réception du message Abort provenant de l'entité SSF (sur abandon d'appel), et qui provoque un message correspondant à l'événement SCSM (e4).



^{a)} Comprenant les opérations call information request, apply charging avec les opérations report request et request notification charging event.

^{b)} Comprenant les opérations call information report, apply charging report et event notification charging.

FIGURE 19/Q.1218

Développement partiel de l'état 2 de la machine FSM

Cet état se développe ensuite dans une machine FSM, qui est décrite à la Figure 20.

Cette machine FSM ne décrit pas explicitement toutes les combinaisons possibles des fonctions de surveillance des ressources utilisées pour la mise en file d'attente. Les possibilités ci-après peuvent être utilisées dans les réalisations:

- Request first status match report au moyen de l'entité SCME;
- Request current status change report au moyen de l'entité SCME;
- Request every status change report au moyen de l'entité SCME; et
- Surveillance fondée sur l'émission par le modèle SCSM de l'opération request event report BCSM et de la réception consécutive de l'opération event report BCSM pour signaler la disponibilité de la ressource. La demande et le rapport apparaissent dans un seul contexte d'appel différent. Dans ces cas, les opérations Query et Update 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.

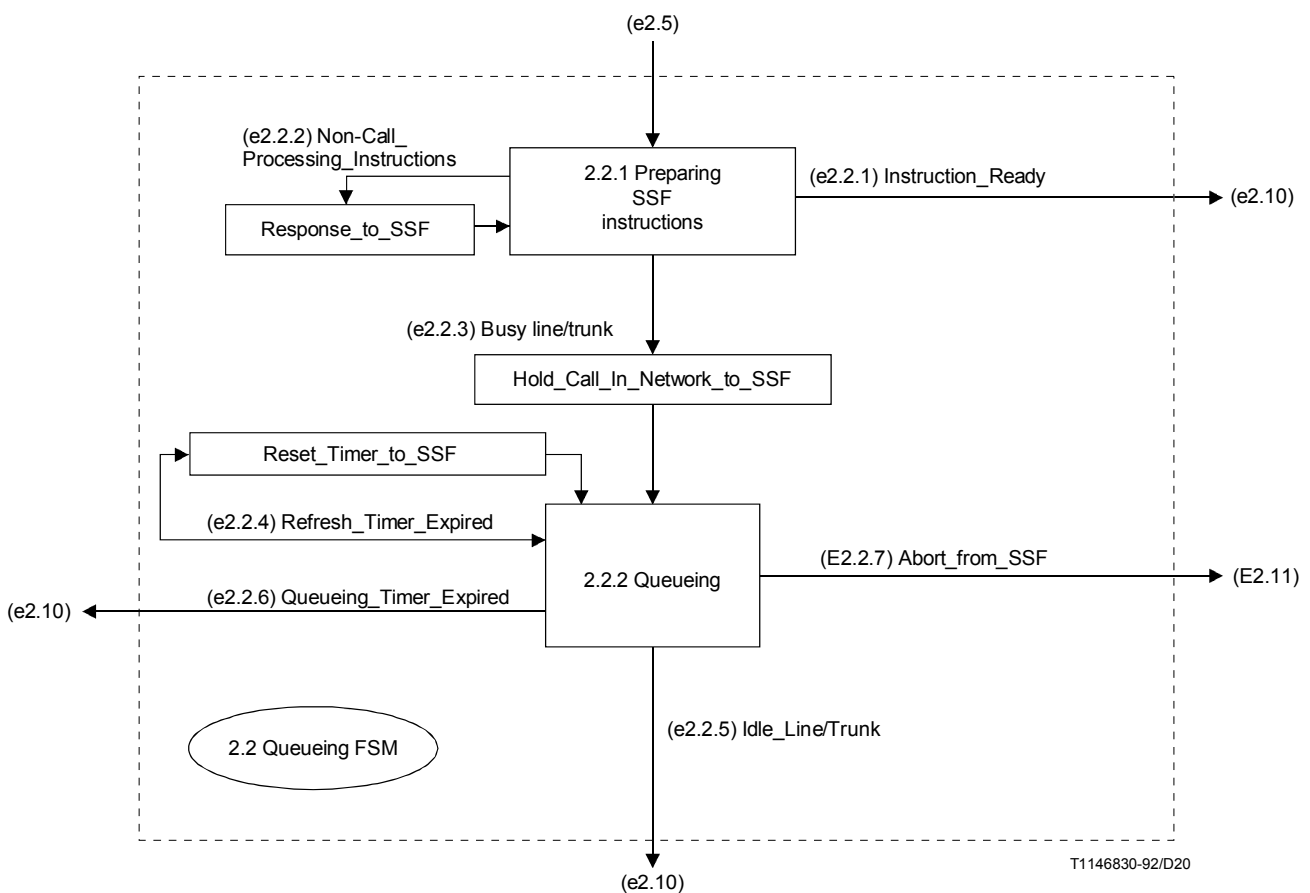


FIGURE 20/Q.1218

Développement partiel de l'état 2 de la machine FSM concernant la mise en file d'attente

3.1.2.5.2.2.1 Etat 2.2.1 Preparing SSF instructions (Préparation des instructions pour l'entité SSF)

Dans cet état, le modèle SCSM prépare les instructions à l'intention de l'entité SSF pour l'exécution de l'appel. Les événements ci-après sont considérés comme étant 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 Preparing SSF instructions [transition (e2.10)].

- (e2.2.2) Non-Call_Processing_Instructions: événement interne produit par la logique de service lorsqu'il est nécessaire d'envoyer cette opération vers l'entité SSF. Elle produit une ou plusieurs des opérations ci-après envoyées à l'entité SSF:
 - a) Apply charging;
 - b) Call information request;
 - c) Cancel (pour call information request);
 - d) Furnish charging information;
 - e) Request report BCSM event;
 - f) Request notification charging event;
 - g) Reset timer; et
 - h) Send charging information.

Cet événement provoque le retour à l'état 2.2.1, Preparing SSF instructions.

- (e2.2.3) Busy_Line/Trunk: événement interne produit par l'objet resource control object lorsqu'aucune ligne/jonction terminale n'est disponible. Cet événement produit l'envoi de l'opération hold call in network à l'entité SSF, 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. Lorsque le modèle SCSM passe à cet état, le temporisateur de mise en file d'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 reset timer doit être envoyée à l'entité SSF/CCF pour que celui-ci n'abandonne pas l'appel. La valeur de cette temporisation est coordonnée avec celle du temporisateur T_{SSF} concerné dans l'entité SSF/CCF.

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 reset timer 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 preparing SSF instructions [Transition (e2.10)] (les procédures qui suivent dépendent de la décision de la logique de service qui diffuse ou non l'annonce terminale).
- (E.2.2.7) 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 (entité SCME) veille à la mise à jour des données de mise en file d'attente au moyen de l'opération cancel de l'objet resource control object.

3.1.2.5.2.3 Etat 2.3: «Waiting for notification or request» (Attente ou de demande de notification)

Dans cet état, le modèle SCSM attend une notification ou une demande de l'entité SSF.

Après passage à cet état, le modèle SCSM arrête le temporisateur $T_{SCF-SSF}$.

On considère que les événements ci-après sont dans cet état:

- (E2.7) EDP-R: événement externe, déclenché par la réception d'une des opérations suivantes:
 - a) Event report BCSM (pour un point EDP-R); et
 - b) famille d'opérations DP-specific request instructions.

Cet événement provoque le passage à l'état 2.1 Preparing SSF instructions.

- (E2.8) Not_Last_EDP-N: événement externe, déclenché par la réception d'une des opérations ci-après:
 - a) Apply charging report;
 - b) Call information report;
 - c) Event report BCSM (pour un point EDP_N);
 - d) Event notification charging; et
 - e) famille d'opérations DP-specific request instructions.

Dans ce cas, il y a toujours un point EDP particulier armé¹¹⁾. Cet événement provoque le retour à l'état 2.3 Waiting for notification or request.

- (E2.9) Last_EDP-N: événement externe, déclenché par la réception de l'une des opérations suivantes:
 - 1) Apply charging report;
 - 2) Call information report;
 - 3) Event report BCSM (pour un point EDP_N);
 - 4) Event notification charging; et
 - 5) famille d'opérations DP-specific request instructions.

Dans ce cas, il n'y a pas de point EDP particulier armé¹¹⁾. Cet événement correspond à l'événement SCSM (e4).

Fin de la description de l'état 2 Preparing SSF instructions.

3.1.2.5.3 Etat 3: «Routing to resource» (acheminement vers la ressource)

Le terme ressource désigne l'un des moyens relevant de l'entité SRF (par exemple, périphérique intelligent).

Dans cet état, il est nécessaire d'avoir des interactions avec l'entité SSF. 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) Hand-off_Needed: lorsque la procédure de «passation» est déclenchée, le modèle SCSM met un terme à l'interaction avec l'entité SSF déclencheuse. Cet événement provoque le passage à l'état 1, idle. Lorsque l'entité SCME reçoit l'opération Assist request instructions en provenance de l'entité SSF vers laquelle la passation a été effectuée, l'entité SCME crée un nouveau modèle SCSM.
- (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, Preparing SSF instructions; et
- (e10) Timer_Expired: cet événement correspond à la fin de la temporisation T_{ASSIST/HAND-OFF}. Cet événement provoque le passage à l'état 2, Preparing SSF instructions.

Pour décrire plus avant les procédures relatives à cet état, l'état est divisé en trois sous-états, décrits dans les trois paragraphes suivants. Cette subdivision est représentée à la Figure 21.

3.1.2.5.3.1 Etat 3.1 «Determine mode» (déterminer le mode)

Dans cet état, le modèle SCSM détermine le mode d'interaction d'utilisateur pour connecter l'appel à l'entité SRF. On considère que les événements suivants appartiennent à cet état:

- (e3.1) Instruction_Ready: événement interne qui a lieu seulement dans le cas entité SSF déclencheuse relais. Dans ce cas, le modèle SCSM envoie à l'entité SSF déclencheuse l'opération connect to resource accompagnée de l'opération play announcement ou prompt & collect user information et passe à l'état 4 user interaction. Ce passage correspond à l'événement (e7);
- (e3.2) Assist_Needed: événement interne qui se produit lorsqu'il faut l'entité SSF assistante ou lorsqu'il faut une relation directe SCF-SRF. Dans ce cas, le modèle SCSM envoie l'opération establish temporary connection 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 Waiting for assist request instructions; et

¹¹⁾ Comprenant les opérations suivantes call information report, notification charging event et/ou apply charging report lorsque le paramètre «send calculation to SCF indication» pour l'opération apply charging est mis à «vrai».

- (e3.3) Hand-off_Needed: événement interne qui n'a lieu que dans le cas de passation. Dans ce cas, le modèle SCSSM envoie l'opération connect ou select route avec l'adresse de l'entité SSF vers laquelle la passation a lieu en direction de l'entité SSF déclencheuse, et passe à l'état 1 idle.

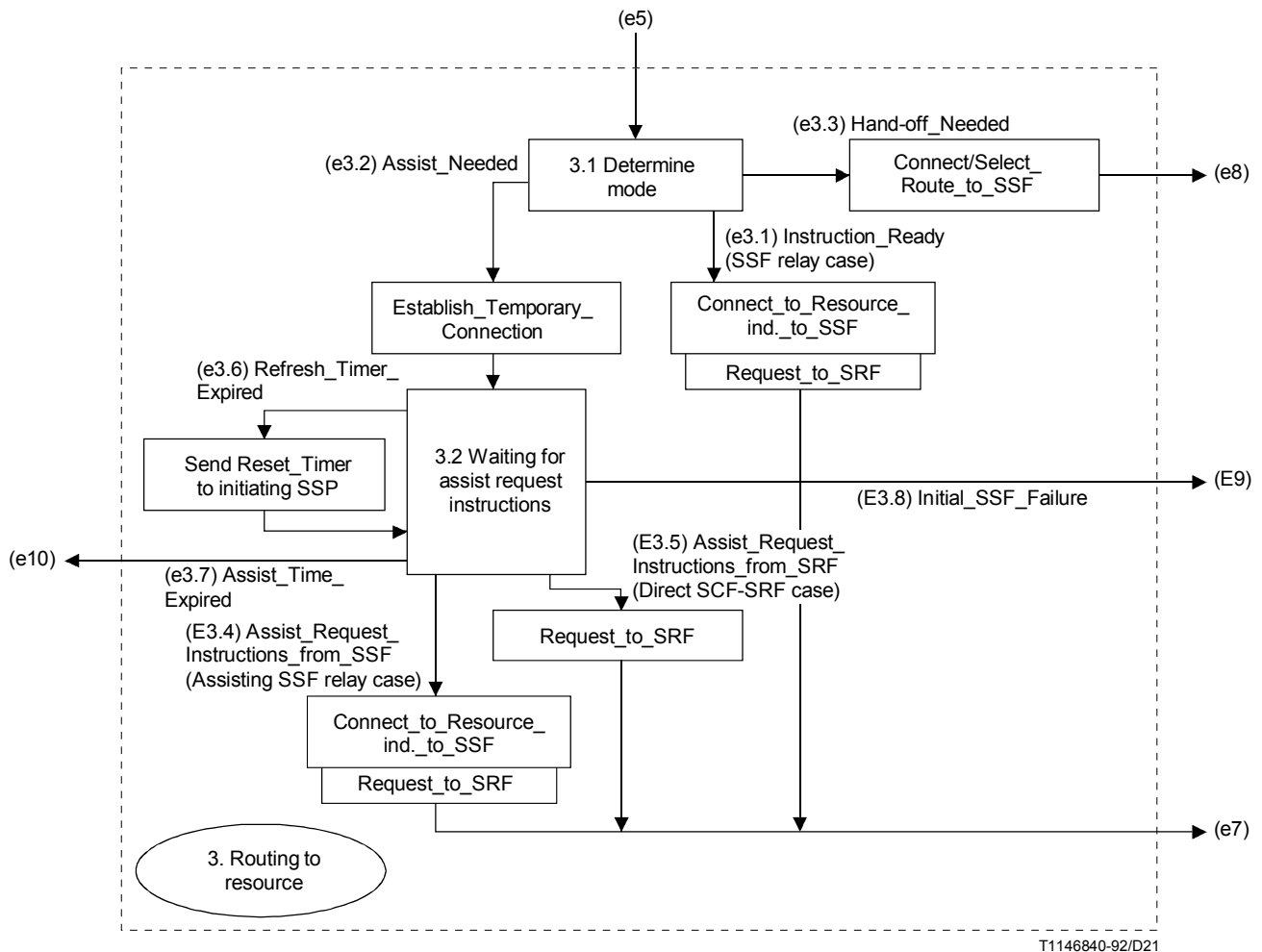


FIGURE 21/Q.1218

Machine FSM associée à l'état 3

3.1.2.5.3.2 Etat 3.2 «Waiting for assist request instructions» (attente d'instruction de demande d'assistance)

Dans cet état, le modèle SCSSM attend l'opération assist request instructions provenant de l'entité SSF assistante (cas entité SSF relais) ou de l'entité SRF (cas direct SCF-SRF). Après passage à cet état, le modèle SCSSM déclenche le temporisateur $T_{ASSIST/HAND-OFF}$ et réinitialise le temporisateur $T_{SCF-SSF}$. On considère que les événements ci-après font partie de cet état:

- (E3.4) Assist_Request_Instructions_from_SSF (cas relais par l'entité SSF assistante): événement externe causé par la réception de l'opération assist request instructions provenant de l'entité SSF assistante. Dans ce cas, le modèle SCSSM transmet l'opération connect to resource accompagnée par l'opération play announcement ou prompt & collect user information vers 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 direct): événement externe déclenché par la réception de l'opération assist request instructions provenant de l'entité SRF. Dans ce cas, le modèle SCSSM transmet l'opération play announcement ou prompt & collect user information vers l'entité SRF, et passe à l'état 4, user interaction. Cette transition correspond à l'événement (e7);
- (e3.6) Refresh_Timer_Expired: événement interne qui se produit à la fin de la temporisation $T_{SCF-SSF}$. Dans ce cas, le modèle SCSSM transmet l'opération reset timer à l'entité SSF déclencheuse, et revient au même état;

- (e3.7) Assist_Timer_Expired: événement interne qui se produit à la fin de la temporisation $T_{ASSIST/HAND-OFF}$. Dans ce cas, le modèle SCSM informe l'entité SCME et le programme de logique de service passe à l'état 2 Preparing SSF instructions. Cet événement correspond à l'événement (e10); et
- (E3.8) Initial_SSF_Failure: événement externe provoqué par la réception d'une anomalie SSF. Cet événement provoque une transition qui correspond à l'événement (E9) du modèle SCSM.

3.1.2.5.4 Etat 4: «User interaction» (Interaction d'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 annonces). Lorsque l'interaction a eu lieu, l'entité SCF peut demander à l'entité SSF de déconnecter le support entre entités SSF et SRF. Elle peut aussi envoyer une opération user interaction (interaction d'utilisateur) indiquant qu'elle autorise la déconnexion déclenchée par l'entité SRF.

En passant à cet état, le modèle SCSM réinitialise le temporisateur $T_{SCF-SSF}$.

Les événements ci-après provoquent la sortie de cet état:

- (e11) Continue_SCF_Processing: dans ce cas, l'entité SCF a obtenu toutes les informations nécessaires de la part de l'entité SRF, qui sont nécessaires pour charger l'entité SSF d'exécuter l'appel. Cet événement provoque le passage à l'état 2, Preparing SSF instructions.
- (E12) Failure_from_SRF: dans ce cas, l'entité SCF constate que:
 - a) la ressource choisie ne peut assurer sa fonction; et
 - b) la ressource choisie ne peut être remplacée.

De la même manière, cet événement provoque le passage à l'état 2, Preparing SSF instructions.

Afin de pouvoir examiner le traitement de cet état avec plus de détail, on a utilisé une machine FSM distincte décrite à la Figure 22.

3.1.2.5.4.1 Etat 4.1 «Waiting for response from the SRF» (Attente d'une réponse de l'entité SRF)

Dans cet état, l'entité SCF attend la réponse à l'opération 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 à l'intention de 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 specialized resource report ou return result from prompt and collect user information. Sur réception de cette opération, le modèle SCSM revient au même état;
- (E4.2') Final_Response_from_SRF: événement externe déclenché par la réception de l'opération specialized resource report en réponse à l'opération antérieure play announcement ou return result for prompt & collect user information avec l'autorisation d'une déconnexion déclenchée par l'entité SRF. Dans le cas de prise de relais par l'entité SSF déclencheuse-relais et le cas SCF-SRF direct, sur réception de cet événement, le modèle SCSM passe à l'état 2, Preparing SSF instructions. Cet événement correspond à l'événement (e11);
- (e4.3) Continue_SCF_Processing: événement interne qui se produit lorsque le modèle SCSM met fin à l'état user interaction et demande la déconnexion du support entre l'entité SSF déclencheuse et l'entité SRF au moyen d'une déconnexion déclenchée par l'entité SCF. Dans ce cas, le modèle SCSM envoie l'opération disconnect forward connection à l'entité SSF déclencheuse et passe à l'état 2, Preparing SSF instructions. Cet événement correspond à l'événement (e11).
- (e4.3') Continue_SCF_Processing: événement interne qui se produit lorsque le modèle SCSM met fin à l'état user interaction et demande la déconnexion du support entre l'entité SSF déclencheuse et l'entité SRF au moyen d'une déconnexion déclenchée par l'entité SRF. Dans ce cas, le modèle SCSM envoie l'opération play announcement ou prompt and collect avec la permission de déconnexion déclenchée par l'entité SRF à l'intention de l'entité SRF. Dans le cas d'une entité SSF assistante, la déconnexion déclenchée par l'entité SRF ne peut être utilisée. Dans ce cas, le modèle SCSM revient au même état;
- Les événements (E4.4) Call_Abandoned_from_SRF et (E4.5) Failure_from_SRF, sont mis en correspondance avec l'événement (E.12) Failure_from_SRF et le modèle SCSM passe à l'état 2, Preparing SSF instructions;
- (e4.6) Refresh_Timer_Expired: événement interne qui se produit à la fin de la temporisation $T_{SCF-SSF}$. Dans ce cas, le modèle SCSM transmet l'opération reset timer à l'entité SSF déclencheuse/assistante, et revient au même état; et

- (e4.7) Cancellation_Required: événement interne qui se produit lorsque le modèle SCSM annule l'opération précédente play announcement ou prompt & collect user information. Dans ce cas, le modèle SCSM envoie l'opération cancel à l'entité SSF assistante (cas entité SSF relais) ou l'entité SRF (cas SCF-SRF direct), et revient au même état.

Il convient de noter qu'il y a déconnexion du support entre les entités SSF et SRF lorsque le modèle SCSM quitte cet état.

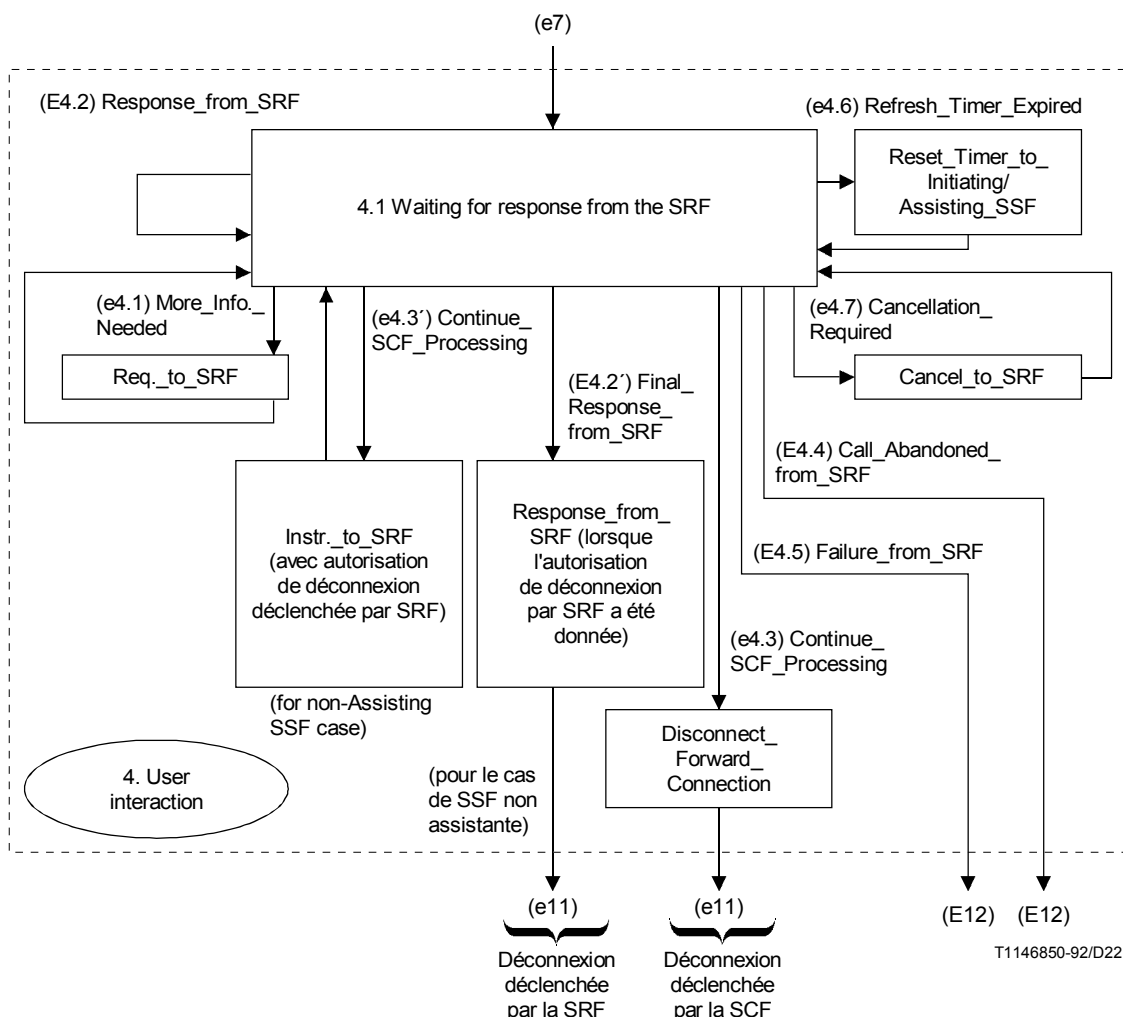


FIGURE 22/Q.1218

Machine FSM représentative de l'état 4

3.1.2.5.5 Etats connexes SDF

L'interaction avec l'entité SDF est possible à partir de n'importe quel état de l'entité SCF. Dans le paragraphe ci-après, on spécifie les états connexes SDF.

3.1.2.5.5.1 Etat 5: «SDF request idle» (Demande SDF repos)

Sous-état d'un état quelconque représentant le cas avant émission de la demande formulée à l'entité SDF. On considère que l'événement ci-après appartient à cet état:

- (e13) Request_to_SDF: événement interne dû à la nécessité pour la logique de service de collecter des informations additionnelles en provenance de l'entité SDF ou de mettre à jour l'entité SDF. Cet événement provoque le passage à l'état 6, Waiting for SDF response.

3.1.2.5.5.2 Etat 6: «Waiting for SDF response» (Attente d'une réponse du SDF)

Dans cet état, l'entité SCF attend une réponse en provenance de l'entité SDF. On considère que les événements ci-après appartiennent à cet état:

- (E14) *Response_from_SDF*: événement externe, déclenché par la réception de la réponse à l'opération Query ou update précédemment émise à l'intention de l'entité SDF. Cet événement provoque le passage à l'état 5 SDF request idle; et
- (E15) *SDF_Response*: événement externe, déclenché par la réception de la SDF Response à une opération Query ou Update précédemment émise à l'intention de l'entité SDF. Cet événement informe l'entité SCF que l'opération demandée peut être différée, et ne provoque pas de sortie de cet état, et le modèle SCSM reste toujours dans l'état 6 Waiting for SDF response.

Il convient de noter que l'entité SCF peut changer la valeur du temporisateur du sous-système TCAP pour l'opération de traitement à une valeur plus élevée dépendante de la réalisation sur réception de l'état (E15).

3.1.3 Procédures relatives à l'entité application de l'entité 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 & 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 de type message prenant en charge les structures de la couche application définies.

3.1.3.2 Modèle et interfaces

Le modèle fonctionnel de la fonction AE-SRF est représenté à la Figure 23/Q.1218; 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 champ 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 coudre réseau (N-) spécifiées et dans la Recommandation Q.711 [interface (2)]. Les opérations et les paramètres associés au protocole d'application du réseau intelligent (protocole INAP) sont définis en 2.

3.1.3.3 Relation entre la machine SRF FSM et les fonctions de maintenance/gestion de connexion support

L'interface élémentaire entre la machine SRF 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.

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 déclenché 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.

Le modèle associe une machine à états finis (FSM) à chaque demande d'interaction initiale provenant de l'entité SCF¹²⁾. 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

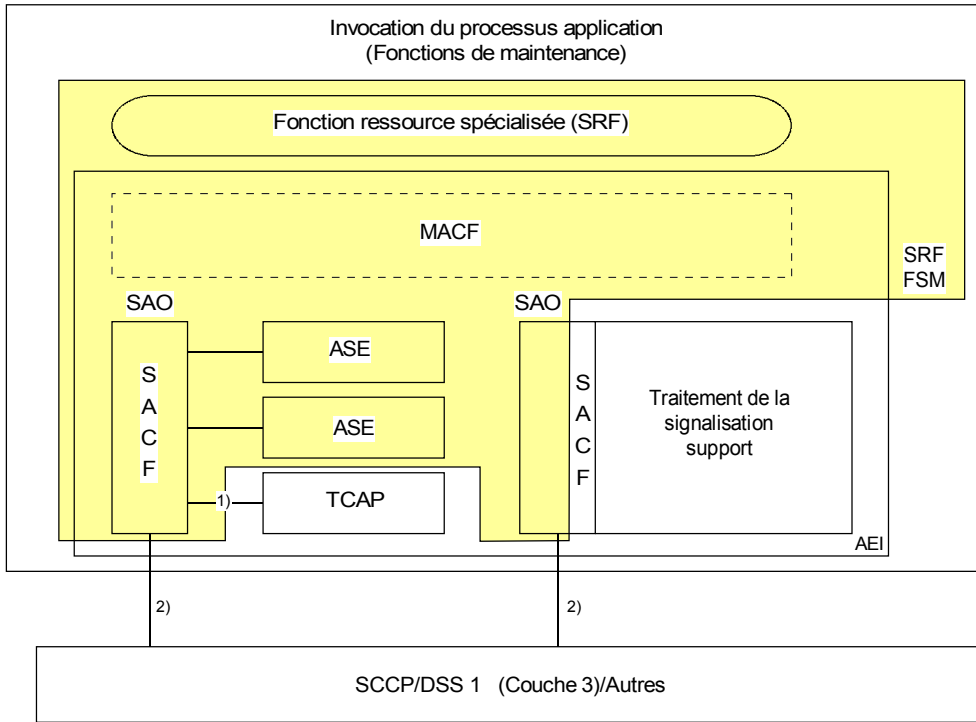
¹²⁾ Une telle demande est exécutée par le modèle SCSM lorsqu'il se trouve dans son état 4.

objets du modèle SRSM. Cette entité est appelée entité de gestion SRF (SRME) (*SRF managing entity*). Outre les tâches ci-dessus, l'entité SRME maintient le dialogue avec les entités SCF et SSF pour toutes les instances du modèle SCSM. En particulier, l'entité SRME:

- 1) interprète les messages de sortie provenant des autres entités fonctionnelles et les traduit en événements SRSM correspondants;
- 2) traduit les sorties SRSM en messages correspondants à l'intention des autres entités fonctionnelles; et
- 3) gère la fonctionnalité de test d'activité pour la relation SCF-SRF.

Enfin, le gestionnaire d'accès aux entités fonctionnelles (FEAM) soulage l'entité SRME des fonctions d'interface de niveau inférieur. Les fonctions assurées par le gestionnaire FEAM sont:

- 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; et
- 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.



T1146860-92/D23

- 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*)

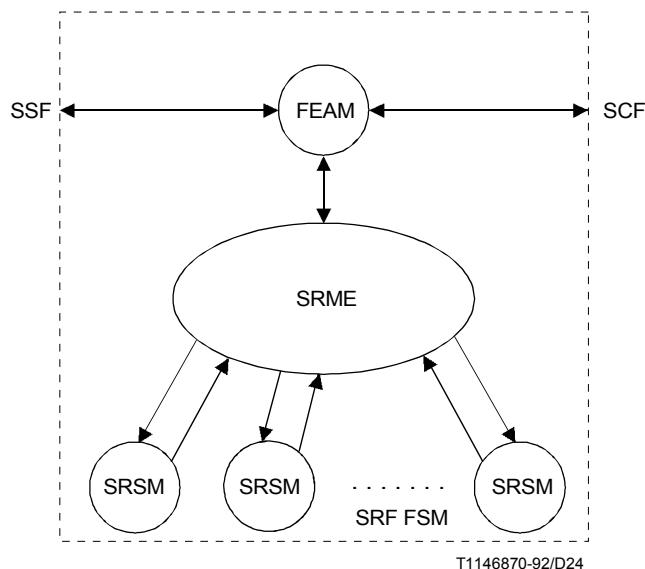
1) Primitives TC ou primitives Q.932.

2) 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



| | |
|------|--|
| 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

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.

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:

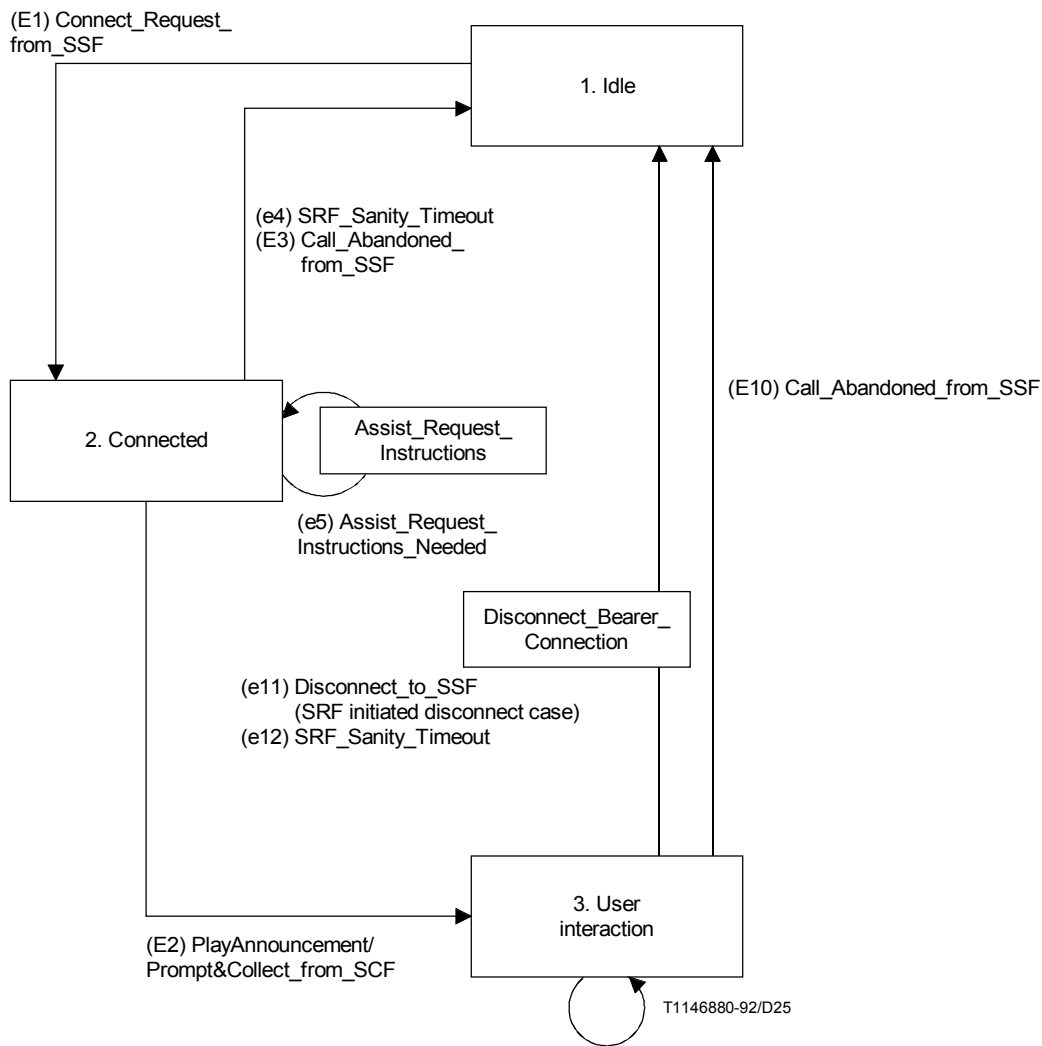
- Le modèle SRSM traite les opérations dans l'ordre de réception.
- Le modèle SRSM examine les opérations successives de la séquence. Lorsqu'il trouve opération annulation (pour une opération play announcement ou prompt & collect) dans la séquence dans l'état user interaction, le modèle SRSM exécute immédiatement. Dans tous les autres cas, le modèle SRSM met en file d'attente les opérations et attend un événement (un tel événement peut être la fin d'une opération en cours d'exécution, ou la réception d'un événement externe).
- S'il y a une erreur de traitement de l'une des opérations de 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, l'interaction est ABORTed (abandonnée).

Dans un état quelconque, lorsqu'il y a une erreur dans une opération reçue, les fonctions de maintenance en sont informées et le modèle SRSM reste dans le même état dans lequel il se trouvait lorsqu'il a reçu les opérations erronées; selon la classe de l'opération, l'erreur peut 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 ou SSF a pris fin, le modèle SRSM en informe les entités SSF ou SCF et revient à l'état idle après s'être assuré que toutes les ressources affectées à l'appel, y compris les dialogues avec les autres fonctions co-situées, ont été retirées.

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 dispose d'un temporisateur d'application, le temporisateur 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 assist request instructions (cas SCF-SRF direct). Le temporisateur est arrêté lorsqu'une demande est reçue en provenance de l'entité SCF. L'entité SRF peut réinitialiser le temporisateur T_{SRF} sur transmission de l'opération specialized resource report ou de renvoi de résultat pour l'opération prompt and collect user information quand il n'y a pas d'opération user interaction 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 retirées.



- (E5) P&P&C_from_SCF
- (E6) Cancel_from_SCF
- (e7) SRF_Report_to_SCF
- (e8) PA/P&C: Cancelled_to_SCF
- (e9) Cancel_Error_to_SCF

FIGURE 25/Q.1218

Le modèle SRSM

3.1.3.4.1 Etat 1: «idle» (repos)

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 état de signalisation support associés à l'établissement de la connexion n'intéressent pas la machine FSM. Le modèle SRSM passe à l'état «connected»;
- (E3) Call_Abandoned_from_SSF: cet événement se produit lorsque le modèle SRSM reçoit un message Release émis par l'entité SSF dans l'état «connected», indiquant que le participant a effectué une déconnexion. 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 «connected» pendant une durée définie par l'opérateur de réseau (temporisateur T_{SRF}) sans avoir d'opération PA/P&C 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»;
- (E10) Call_Abandoned_from_SSF: cet événement se produit lorsque le modèle SRSM reçoit un message release de l'entité SSF dans l'état user interaction; indiquant que le participant s'est déconnecté. 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 PA/P&C à partir de l'état (E2) ou (E5) SCF avec le paramètre. 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 specialized resource report vers l'entité SCF (e7). Le modèle SRSM passe à l'état «idle»; et
- (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 PA/P&C 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 que l'opération initiale PA/P&C n'a pas encore été reçue (par exemple, lorsque l'on utilise les procédures establish temporary connection). 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 sémaphore support émise par l'entité SSF dans l'état idle. Les détails de la machine à état de signalisation support associés à l'établissement de la connexion n'intéressent pas la machine FSM de l'entité SRF. Le modèle SRSM passe à l'état «connected»;
- (E2) PA/P&C_from_SCF: cet événement a lieu lorsque la première ou les premières opération(s) play announcement ou prompt & collect user information émise(s) par l'entité SCF est ou sont reçue(s). Le modèle SRSM passe à l'état «user interaction»;
- (E3) Call_Abandoned_from_SSF: cet événement a lieu lorsque l'entité SRF reçoit un message de libération provenant de l'entité SSF, indiquant que le participant s'est déconnecté. 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'opération de réseau (temporisateur T_{SRF}) sans avoir d'opération PA/P&C à exécuter. 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»;
- (e5) Assist_Request_Instructions_Needed: cet événement se produit lorsque l'opération assist request instructions est envoyée du modèle SRSM vers l'entité SCF en l'absence d'événement (E2) PA/P&C déclenché par la présence d'une opération PA/P&C concaténée avec la demande d'établissement émise par l'entité SSF (E1) (cas SCF-SRF direct). Cet événement ne produit pas de modification d'état.

3.1.3.4.3 Etat 3: «User interaction» (Interaction d'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 PA/P&C qui sont mises en tampon comme indiqué dans les procédures.

- (E2) et (E5) PA/P&C_from_SCF: cet événement est déclenché par la réception d'une ou plusieurs opérations initiales ou suivantes play announcement ou prompt & collect user information 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 PA/P&C): cet événement est déclenché par la réception de l'opération correspondante play announcement ou prompt & collect user information 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 est produit par l'envoi d'une opération specialized resource report vers l'entité SCF. Le modèle SRSM reste à l'état «user interaction».
- (e8) PA/P&C_Cancelled_to_SCF: cet événement a lieu lorsque l'erreur PA/P&C causée par l'opération cancel (pour les opérations play announcement ou prompt & collect user information) est envoyée vers l'entité SRF. Cet événement représente l'annulation effective d'une opération PA/P&C 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 cancel (pour l'opération play announcement ou prompt & collect user information) est envoyée vers l'entité SRF. Cet événement représente l'échec de l'annulation d'une opération PA/P&C. Le modèle SRSM reste dans l'état «user interaction».
- (E10) Call_disconnect_from_SSF: cet événement a lieu lorsque le modèle SRSM reçoit un message de libération en provenance de l'entité SSF indiquant qu'un participant s'est déconnecté. Le modèle SRSM passe à l'état «idle».
- (e11) Disconnect_to_SSF: cet événement a lieu lorsque le modèle SCF a autorisé une connexion déclenchée par l'entité SRF avec la dernière opération PA/P&C 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 specialized resource report vers l'entité SCF. Le modèle SRSM passe à l'état «idle».
- (e12) SRF_sanity_timeout: cet événement a lieu lorsque le modèle SRSM a été connecté pendant une durée définie par l'opérateur de réseau (temporisateur T_{SRF}) sans avoir d'opération PA/P&C à exécuter. 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 avec les exemples de scénarios physiques d'architecture de protocole décrite en 0.2.

Les procédures service assist (assistance au service) et service hand-off (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 point SSP (*service switching point* ou point de commutation de service), ou directement associé à ce point, qui interagit avec le point SCP (*service control point*, point de commande de service), 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 autre nœud qui n'est pas le point SSP qui interagit avec le point SCP mais que 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; et
- 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 «Hand-off» 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 basé sur le sous-système TCAP du système de signalisation n° 7; la signalisation de commande support peut être n'importe quel système.

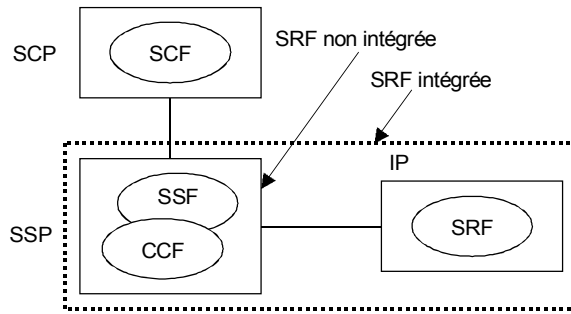
Chaque scénario va être maintenant examiné en utilisant des diagrammes à flèche.

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 connect to resource examinée dans le paragraphe traitant de l'interaction d'utilisateur ci-dessus. L'établissement de la relation SCF-SRF dans ce cas est implicite.

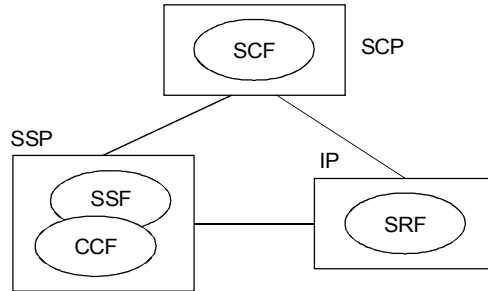
Dans le cas ii) il faut que le périphérique intelligent indique au point SCP qu'il est prêt à recevoir les opérations (voir la Figure 28). L'établissement de la relation SCF-SRF est explicite. Il convient de noter qu'il est nécessaire d'acheminer un identificateur de corrélation (Correlation ID) pour 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 que l'opération Assist Request instructions, les procédures sont les mêmes que dans le cas i). La Figure 29 décrit le préambule utilisé.

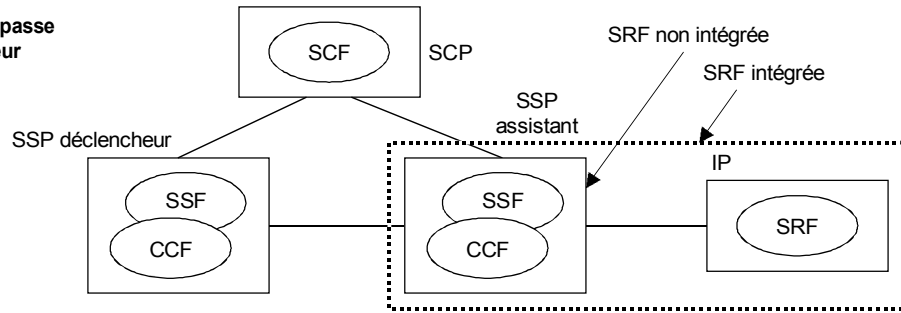
Cas i) SSF relais



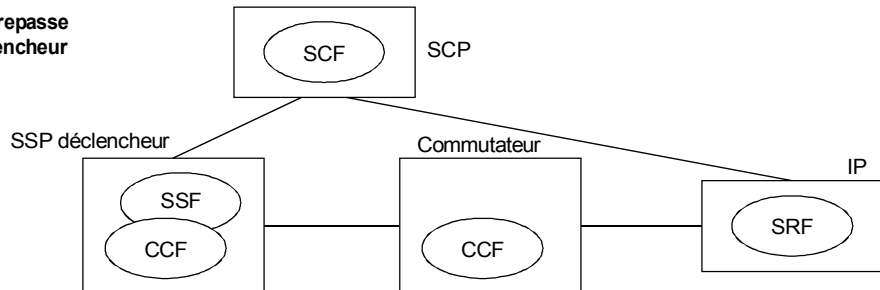
Cas ii) Trajet direct SCP-IP



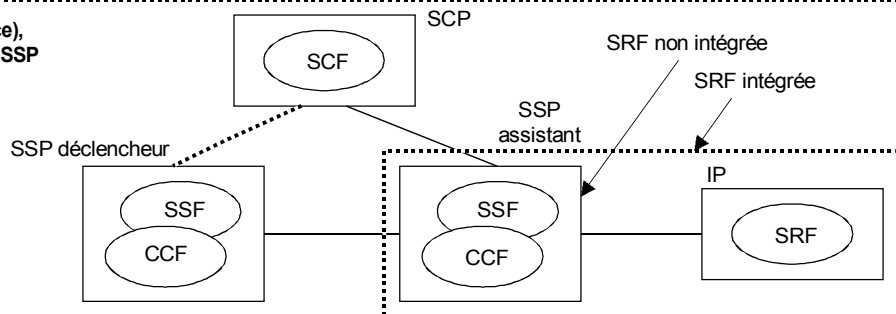
Cas iii) Assistance avec relais, passe la commande au SSP déclencheur après exécution de l'IU



Cas iv) Assistance sans relais, repasse la commande au point SSP déclencheur à la fin de l'IU



Cas v) Hand-off (sans assistance), maintient la commande du point SSP assistant après la fin de l'IU



T1146890-92/D26

FIGURE 26/Q.1218

Scénarios physiques

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.

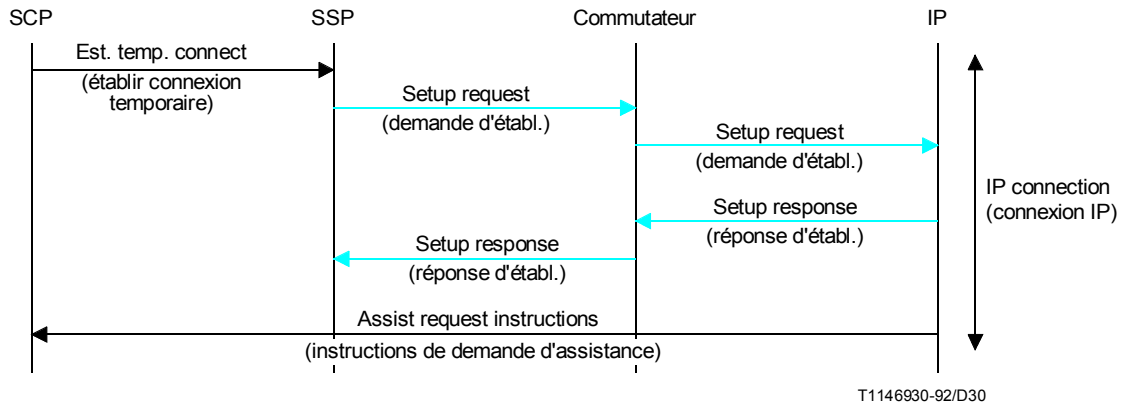


FIGURE 30/Q.1218

Préliminaires pour le cas assist avec IP externe et messagerie directe SCP-IP

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 «Assist Request Instructions» 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 tels 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.

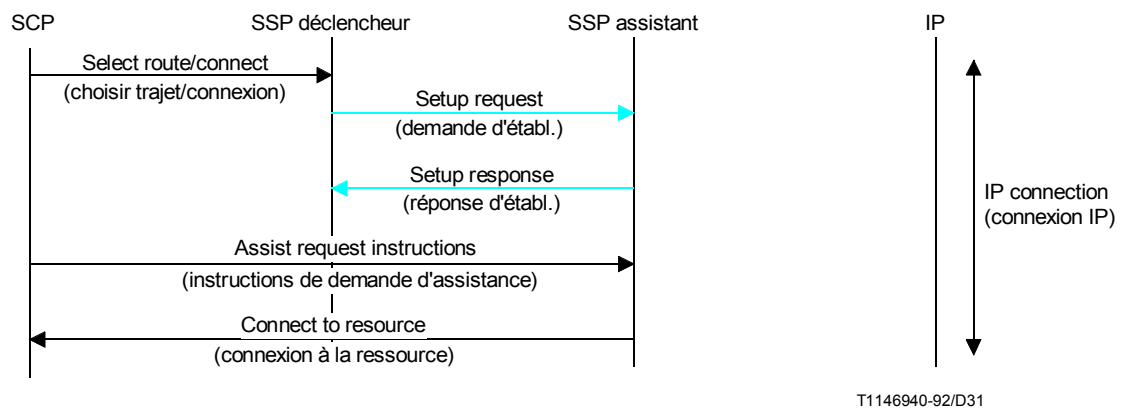


FIGURE 31/Q.1218

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 play announcement (diffusion d'annonce);
- d'établir un dialogue avec l'utilisateur d'extrémité en utilisant une ou une séquence d'opération prompt & collect user information (incitation et collecte d'information d'utilisateur);
- une combinaison des deux éléments ci-dessus; et
- l'annulation d'opérations play announcement ou prompt & collect user information en utilisant une opération générique cancel.

3.1.3.5.2.1 Play announcement/prompt & collect (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); et
- 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 de point SSP (cas direct SCF-SRF).

Le cas i) est illustré à la Figure 32.

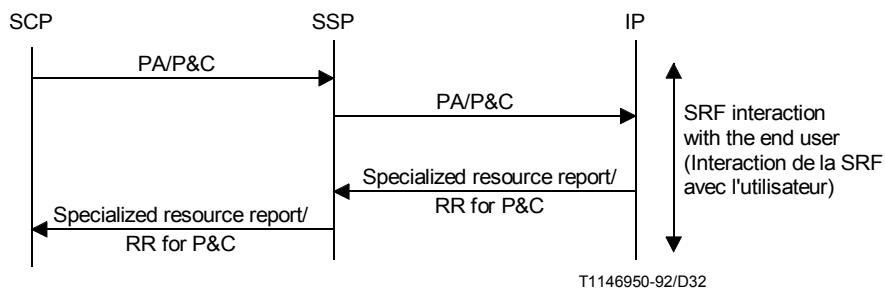


FIGURE 32/Q.1218

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 connect to resource et la première opération PA/P&C. 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 d'extrémité.

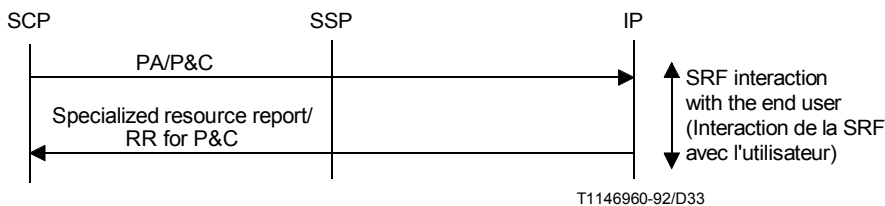


FIGURE 33/Q.1218

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.

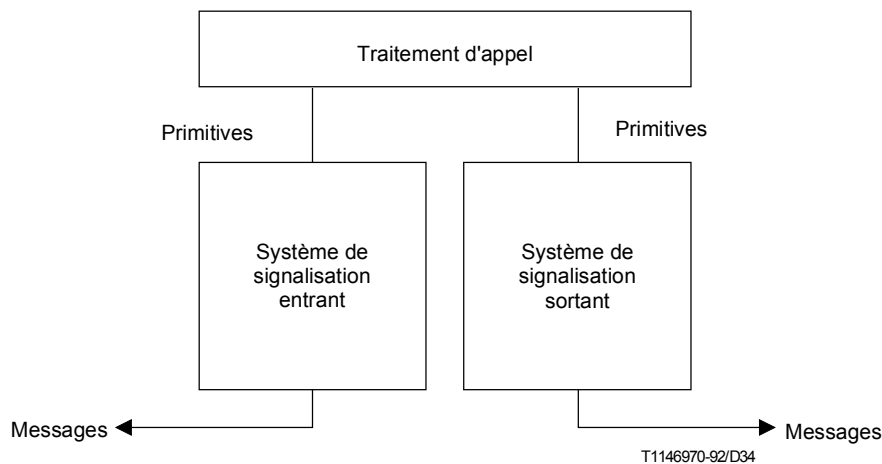


FIGURE 34/Q.1218

Relation entre le traitement d'appel et le système de signalisation entrant et sortant

De même que pour les procédures de connexion SRF, la déconnexion SRF dépend de la configuration physique du réseau.

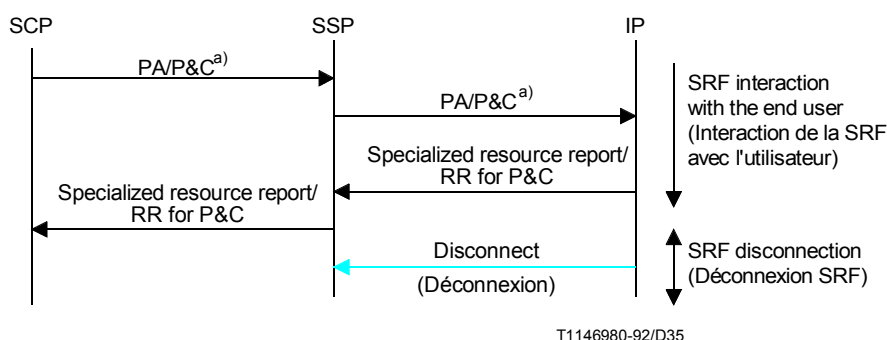
Afin de simplifier l'interface entre les entités SCF et SRF, il est nécessaire de formuler un certain nombre d'hypothèses. Ces hypothèses et les règles qui en résultent, se traduisent par des procédures ambiguës du point de vue des entités SCF et SRF. Les règles, énoncées ci-après concernent la déconnexion 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-dessus ou n'apportent pas d'élément supplémentaire du point de vue du service.

- 1) Si une série d'opérations PA/P&C 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 PA/P&C. Lorsqu'une opération PA/P&C suivante est reçue, elle est mise en tampon jusqu'à la fin de toute opération PA/P&C antérieure.
- 2) Une opération générique cancel termine l'opération PA/P&C 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 PA/P&C mise en tampon, cette opération PA/P&C est mise au rebut, mais les opérations en cours PA/P&C et toutes opérations PA/P&C sont exécutées. Une entité SRF interagit avec un utilisateur seulement et, par conséquent, l'annulation d'une opération PA/P&C n'affecte que l'utilisateur auquel cette entité SRF est connectée.

- 3) L'entité SRF doit explicitement ordonner l'opération «Disconnect» (déconnexion) ou permettre une déconnexion déclenchée par SRF à la fin de l'opération PA/P&C. Une entité SRF maintenue en connexion sans opération PA/P&C à exécuter peut de manière autonome se déconnecter si elle n'a pas reçu d'opération PA/P&C au bout d'un certain temps. Tel est le cas par exemple après qu'une opération establish temporary connection qui n'est pas suivie au bout d'un temps défini d'une opération PA/P&C. 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 PA/P&C, 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 specialized resource report pour annonce effectuée ou utiliser le return result pour l'opération prompt and collect user information.
- 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 «disconnect forward connection», 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 PA/P&C. Lorsque l'entité SRF reçoit une déconnexion par l'opération PA/P&C autorisante, elle termine le dialogue comme indiqué par l'opération PA/P&C, 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 propagent 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.

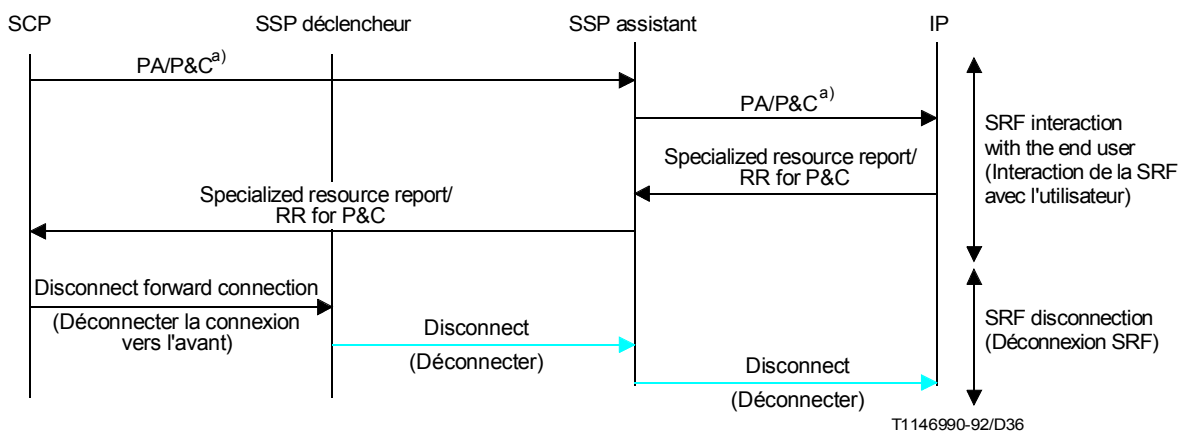
Pour le cas direct SCF-SRF, les procédures fonctionnent de la même façon. La déconnexion SRF est validée par l'entité SCF dans une opération PA/P&C. Lorsque l'entité SRF reçoit la déconnexion par l'opération PA/P&C autorisante, elle met fin au dialogue comme demandé dans l'opération PA/P&C, et déclenche 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 d'extrémité. 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 SSR 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 PA/P&C demandée. L'opération specialized resource report contient une information «annonce terminée» et le retour du résultat (return result) pour l'opération P&C contient l'élément «information collectée».

La déconnexion déclenchée par l'entité SCF fait appel à une opération appelée disconnect forward connection. Une fois que cette opération a été reçue par l'entité SSF, elle déclenche 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 disconnect forward connection 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 establish temporary connection ne se déconnecte pas vers l'utilisateur mais revient à l'état «waiting for instructions».



^{a)} Déconnexion à partir de la SRF interdite.

FIGURE 36/Q.1218

Déconnexion déclenchée par la SCF pour le scénario avec assistance

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.

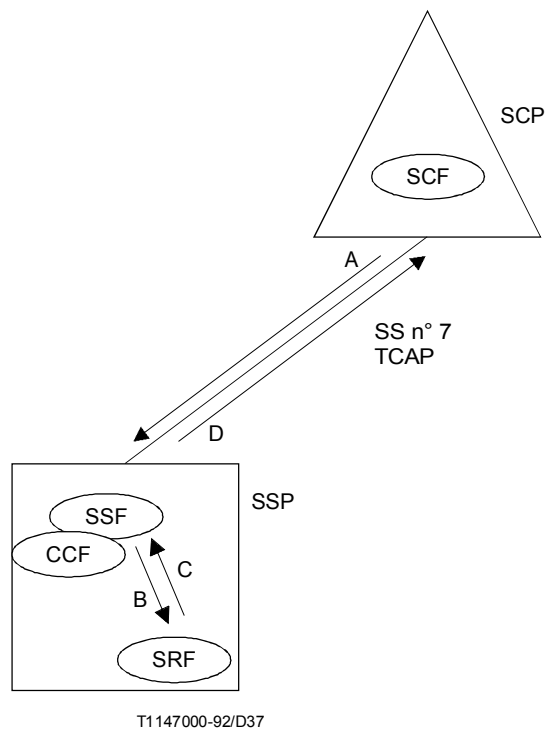


FIGURE 37/Q.1218
Point SSP avec SRF intégrée

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 | Connect to Resource; PA/P&C Setup; PA/P&C | A B |
| Interaction d'utilisateur | PA/P&C Specialized Resource Report/RR for P&C | A puis B C puis D |
| Déconnexion déclenchée par l'entité SRF | Specialized Resource Report/RR for P&C Disconnect | C puis D C (commande du support intégrée au point SSP) |
| Déconnexion déclenchée par l'entité SCF | Specialized Resource Report/RR for P&C Disconnect Forward Connection 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.

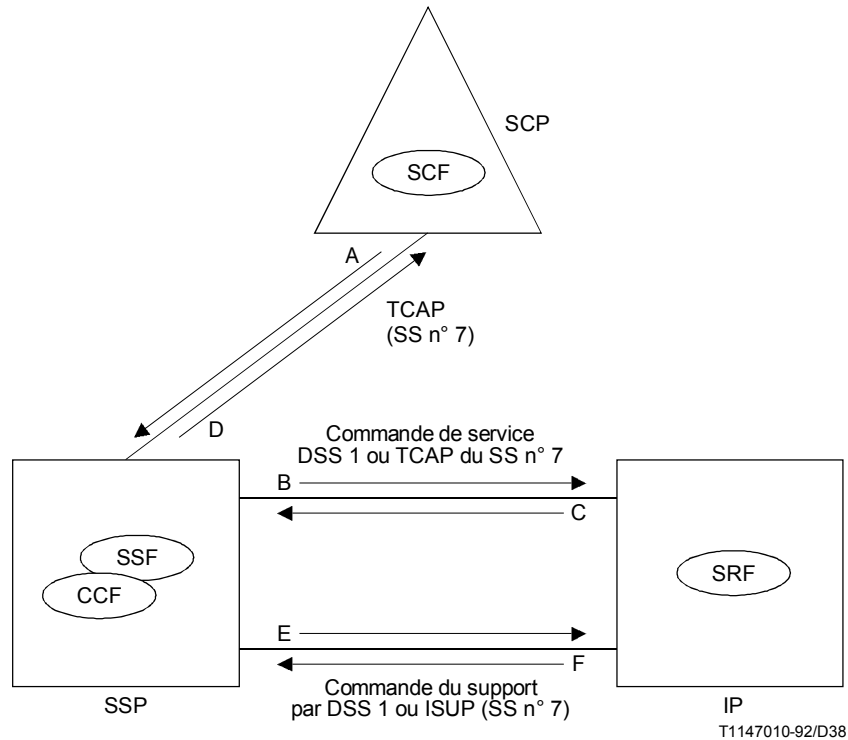


FIGURE 38/Q.1218

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 (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 | Connect to Resource; PA/P&C <i>Si on utilise la signalisation DSS 1</i> Setup; PA/P&C <i>Si on utilise le système de signalisation n° 7</i> Setup PA/P&C | A E et B (élément d'information facilité) E B |
| Interaction d'utilisateur | PA/P&C Specialized resource report/RR for P&C | A puis B C puis D |
| Déconnexion déclenchée par le SRF | Specialized Resource Report/RR for P&C Disconnect | C puis D F |
| Déconnexion déclenchée par le SCF | Specialized resource report/RR for P&C Disconnect forward connection Disconnect | C puis D A 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.

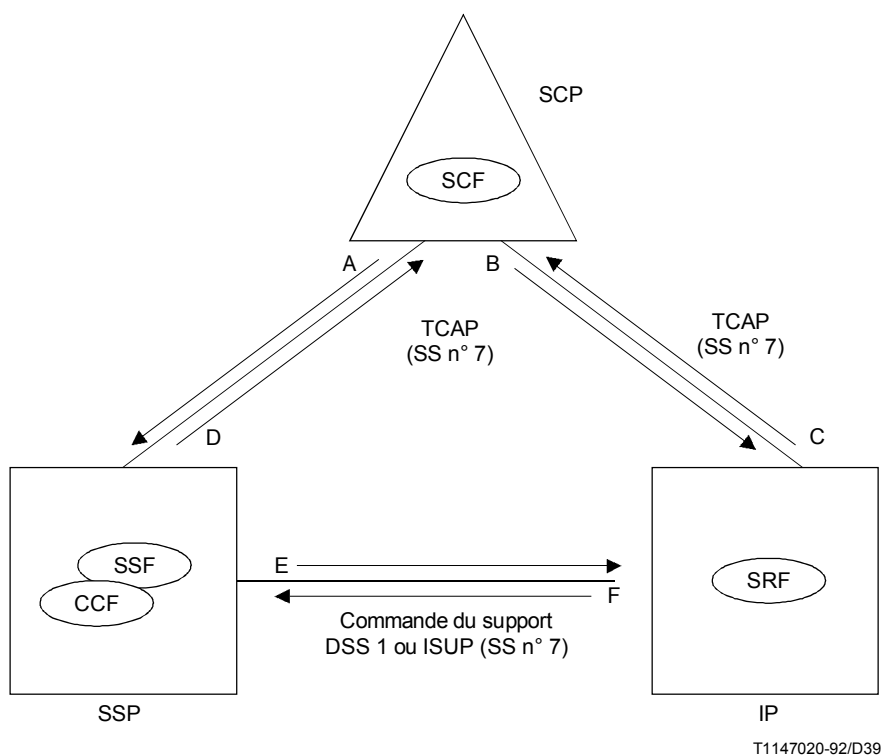


FIGURE 39/Q.1218
Transfert d'information direct SCP-IP

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 | Est. temporary connection | A |
| | Setup | E |
| | Assist request instructions | C |
| Interaction d'utilisateur | PA/P&C | B |
| | Specialized resource report/RR for P&C | C |
| Déconnexion déclenchée par SRF | Specialized Resource Report/RR for P&C | C |
| | Disconnect | F |
| | | |
| Déconnexion déclenchée par SCF | Specialized resource report/RR for P&C | C |
| | Disconnect forward connection | A |
| | Disconnect | 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.

Il convient de noter que le cas SRF intégré et SSF relais requiert une transaction entre le point SCP et le point SSP assistant (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.

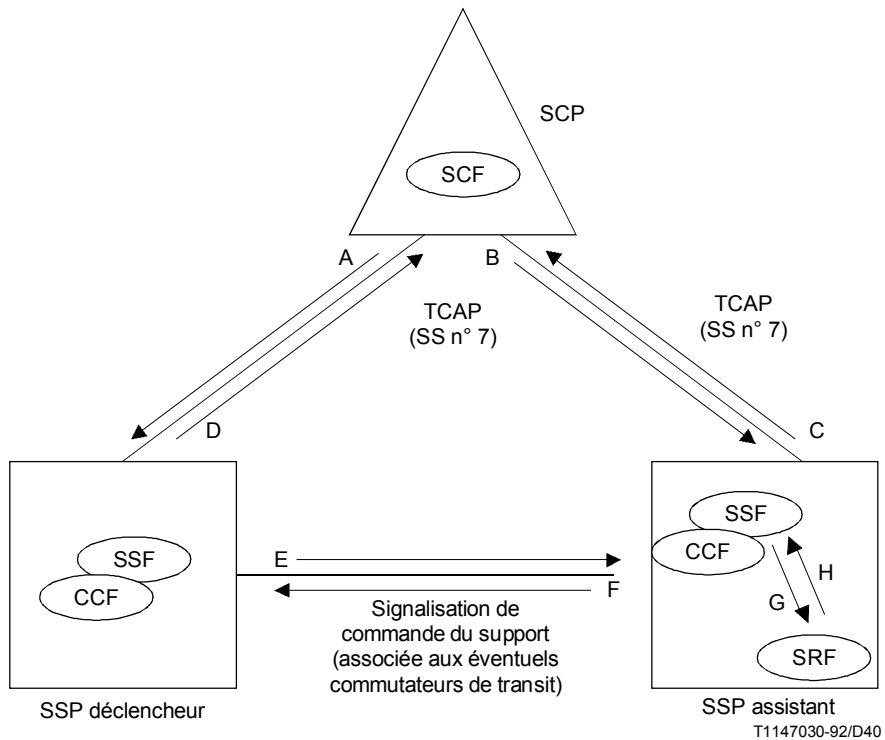


FIGURE 40/Q.1218
Point SSP assistant (SSP relais)

A la Figure 40, les scénarios de procédure peuvent être mis en correspondance comme suit:

| Nom de procédure | Opérations | Flux de protocole |
|--------------------------------|--|---|
| Préambule Assist | Establish temp. connection Setup Assist request instructions Connect to SR; PA/P&C Setup PA/P&C | A E C B G G |
| Interaction d'utilisateur | PA/P&C Specialized resource report/RR for P&C | B puis G H puis C |
| Déconnexion déclenchée par SCF | Specialized resource report/RR for P&C Disconnect forward connection Disconnect | H puis C A F puis 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 être également considéré comme composé d'une séquence faisant intervenir un service RI acheminant un appel d'un point SSP à un autre, suivie par l'un quelconque des scénarios d'interaction d'utilisateur physiques précédemment décrits. En conséquence, aucun diagramme ou tableau n'est fourni pour cette simple extension.

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.771 et Q.1400, comprend le sous-système TCAP (*transaction capabilities application part*) et un ou plusieurs éléments du service d'application appelé utilisateur-TC. Dans ce qui suit les éléments du service application des utilisateurs-TC et les règles SACF et MACF, qui assurent l'interface avec le sous-système TCAP au moyen de primitives spécifiées dans la Recommandation Q.771 sont définis.

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.

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 champ 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 en 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

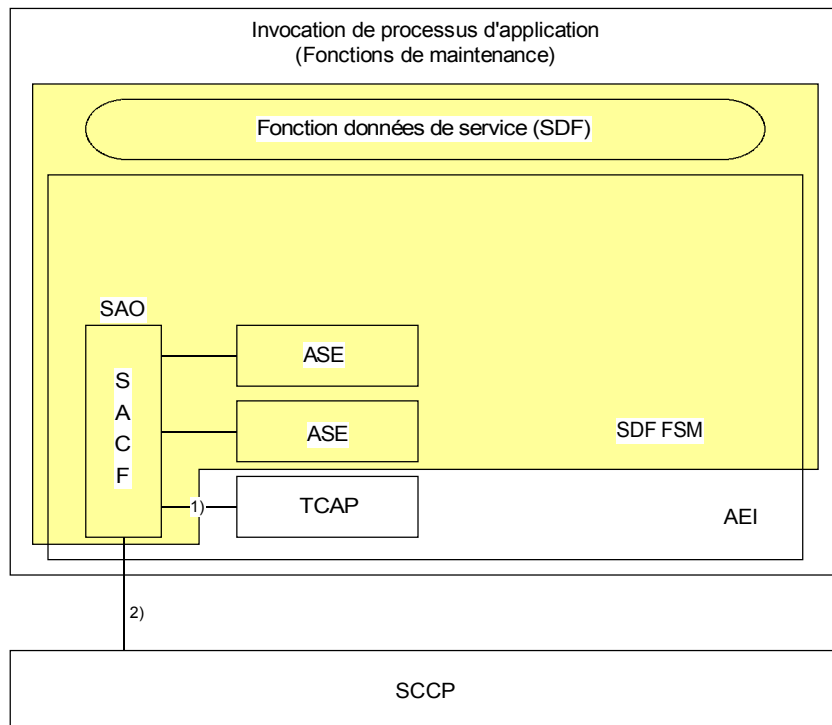
S'agissant de l'ensemble de capacités CS-1, la tâche de l'entité SDF est de répondre (de manière synchrone) à chaque demande émanant de l'entité SCF. Par conséquent la machine à états finis applicable décrite à la Figure 42 est évidente.

Chaque état est étudié dans l'un des paragraphes suivants.

Les règles générales applicables à plusieurs états sont les suivantes.

Dans un état quelconque, lorsqu'il y a une erreur dans une opération reçue, les fonctions de maintenance en sont informées et la machine SDF FSM reste dans le même état (c'est-à-dire l'état «Idle» (Repos)); selon la classe de l'opération, l'erreur peut être signalée par l'entité SDF à l'entité SCF en utilisant la composante appropriée (Recommandation Q.774).

Dans un état quelconque, si le dialogue avec l'entité SCF a pris fin, la machine SDF FSM revient à l'état repos après s'être assurée que toutes les ressources attribuées à l'appel ont été retirées.



T1147040-92/D41

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*)

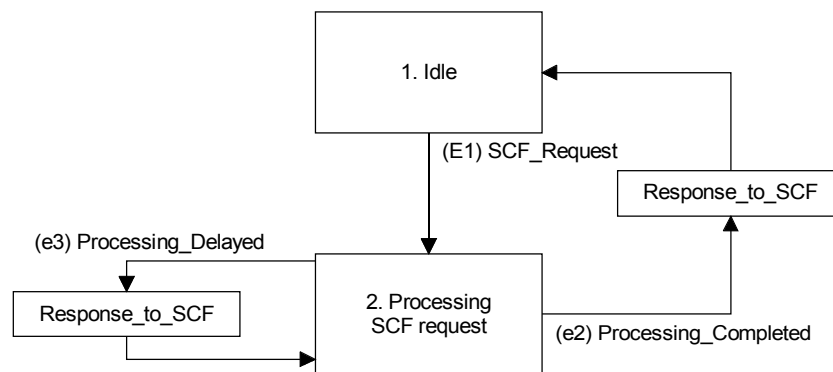
1) Primitives TC ou primitives Q.932.

2) 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



T1147050-92/D42

FIGURE 42/Q.1218

Machine SDF FSM

3.1.4.4.1 Etat 1 «Idle» (repos)

Le seul événement accepté dans cet état est (E1) SCF_Request. Il s'agit d'un événement externe produit par la réception d'une opération Query ou Update data provenant de l'entité SCF. Cet événement provoque le passage à l'état 2, Processing SCF request.

3.1.4.4.2 Etat 2: «Processing SCF request»

Les événements acceptés dans cet état sont les suivants:

- (e2) Processing_Completed; et
- (e3) Processing_Delayed.

L'événement (e2) Processing_Completed est un événement interne déclenché par la fin de la demande SCF request. Cet événement provoque l'envoi de la réponse à l'opération query ou update à l'entité SCF et le passage à l'état 1, idle.

L'événement (e3) Processing_Delayed est un événement interne produit par la détection d'un retard à la demande SCF request. Afin d'informer l'entité SCF de la situation, la machine SDF FSM envoie l'opération SDF response à l'entité SCF et reste dans l'état processing SCF request.

Il est important de ne pas oublier que pour l'interfonctionnement des réseaux, l'entité SDF peut renvoyer à l'entité SCF une référence à une autre entité physique contenant l'information demandée, au lieu de l'information demandée elle-même.

Cependant, l'obtention de cette information peut nécessiter l'envoi de plusieurs demandes de la part de l'entité SCF à différentes entités physiques SDF connexes.

Appendice I

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)

I.1 Généralités

I.1.1 Considérations générales

Le présent appendice porte sur la gestion de participant et d'autres points qui se sont révélés, au cours de l'élaboration de la présente Recommandation, être incomplets. Bien que les éléments contenus dans le présent appendice ont été élaborés autour de l'ensemble CS-1, les procédures applicables à ces capacités peuvent s'avérer insuffisamment définies et appeler un complément d'étude. Les éléments inclus dans le présent appendice contiennent une certaine base technique destinée aux travaux futurs.

I.1.2 Relation avec les autres appendices des Recommandations de la série Q.1200

Le présent appendice ne concerne que la présente Recommandation. Chaque Recommandation de la série Q.1200 comporte le cas échéant un appendice spécifique.

I.1.3 Format des documents

La présente introduction donne une explication de l'objet et du champ d'application de l'appendice.

Le paragraphe I.2 contient la liste des opérations.

Le paragraphe I.3 contient la liste des paramètres.

Le paragraphe I.4 est le module ASN.1 des opérations de paramètre.

Le paragraphe I.5 contient les procédures applicables aux opérations.

I.2 Opérations

Les opérations citées dans le présent paragraphe de l'appendice viennent s'ajouter aux opérations de l'article 2.

I.2.1 Considération applicable à toutes les opérations du présent appendice

Les opérations ou aspects d'opérations ci-après appellent un complément d'étude en ce qui concerne l'ensemble 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 par défauts définies peuvent être incomplets (par exemple identificateur de branche) et peuvent dépendre du point PIC (*point in call*) utilisé.

I.2.2 Opération Add Party (Ajouter participant)

I.2.2.1 Considérations générales

La différence entre cette opération et l'opération attach (associer) doit être précisée.

I.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 un appel conférence). Du point de vue de l'entité de commande, cette opération effectivement réunit deux appels.

I.2.3 Opération attach (associer)

I.2.3.1 Considérations générales

La différence entre cette opération et l'opération add party (ajouter participant) doit être précisée.

I.2.3.2 Description

SCF → SSF

Cette opération permet à la fonction SCF de demander à la fonction SSF d'inclure une branche dans l'instance de relation en cours. Cette branche est transférée à partir d'une autre instance de relation, instance à partir de laquelle elle a été supprimée en utilisant l'opération detach. Il faut noter que l'opération detach peut être également exécutée après l'opération attach en utilisant le même identificateur absolu.

I.2.4 Opération «change parties» (opération Changement de participants)

I.2.4.1 Description

SCF → SSF

Cette opération est utilisée pour exécuter les actions de traitement d'appel visant à transférer une connexion de correspondant particulière d'un appel à un autre appel. Du point de vue du participant à l'appel considéré, cette opération effectivement met en garde le premier appel et récupère l'appel associé mis en garde.

I.2.5 Opération de détachement

I.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.

I.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 à une autre instance relationnelle, à laquelle il sera attaché au moyen de l'opération de détachement et du même identificateur absolu.

I.2.6 Opération de mise en garde d'une connexion de correspondant

I.2.6.1 Description

SCF → SSF

Cette opération est effectuée pendant la phase active d'une communication entre deux ou plus de deux correspondant afin de mettre en garde une seule connexion de correspondant.

I.2.7 Opération de tentative d'appel au départ (dans le cas de plus d'un correspondant)

I.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 seul correspondant dans la même communication, ce qui, pour l'ensemble CS-1, doit faire l'objet d'un complément d'étude.

I.2.7.2 Description

SCF → SSF

Cette opération sert à demander à l'entité SSF de créer un nouvel appel à destination d'un ou 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 éventuelles erreurs associées à la exécution de cette opération feront l'objet d'un retour de résultat.

I.2.8 Opération de reconnexion

I.2.8.1 Description

SCF → SSF

Cette opération sert à réintégrer dans une communication un correspondant mis en garde (c'est l'inverse de l'opération de mise en garde d'une connexion de correspondant).

I.2.9 Opération de libération de la connexion d'un correspondant

I.2.9.1 Considérations générales

La différence entre cette opération et l'opération de détachement doit faire l'objet d'un complément d'étude.

I.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.

I.3 Paramètres

Les paramètres énumérés dans ce paragraphe sont des paramètres additionnels pour les opérations répertoriées à l'article 2.

I.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. Il ne sont donc inclus dans cet appendice que pour mémoire.

I.3.2 Paramètre Identification de demi-appel crée (issu de l'opération d'analyse des informations)

I.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.

I.3.3 Paramètre Identification de demi-appel crée (issu de l'opération de connexion)

I.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.

I.3.4 Paramètre Identification de demi-appel crée (issu de l'opération de tentative d'appel au départ)

I.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.

I.3.5 Paramètre Identification de demi-appel crée (issu de l'opération de sélection d'équipement)

I.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.

I.3.6 Paramètre Identification de demi-appel crée (issu de l'opération de de sélection de route)

I.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.

I.3.7 Paramètre demi-appel 1 (issu de l'opération du point de détection initial)

I.3.7.1 Description

DEFAULT {aPARTY, pending} (correspondant A par défaut, en attente)

Ce paramètre indique une information relative à un correspondant, tel 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.

I.3.8 Paramètre demi-appel 2 (issu de l'opération du point de détection initial)

I.3.8.1 Description

OPTIONAL

Ce paramètre indique une information relative à un correspondant, tel 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.

I.3.9 Paramètre identification d'appel

I.3.9.1 Description

Indique un identificateur pour désigner une instance d'appel accessible à l'entité SCF. Voir au 4.2.2.1/Q.1214 la description du segment d'appel.

I.4 Modules en notation ASN.1 des opérations et des paramètres

Les modules ci-après 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 en I.3.

I.4.1 Syntaxe abstraite du protocole d'application de l'ensemble CS-1 de réseau intelligent – appendice

Le paragraphe ci-après spécifie la syntaxe abstraite additionnelle pour le protocole d'application de l'ensemble CS-1 de 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 an 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

I.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 bras» 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.

Imprimé en Suisse

Genève, 1993