



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

Q.2630.1

Annexe B
(03/2001)

SÉRIE Q: COMMUTATION ET SIGNALISATION

RNIS à large bande – Aspects communs des protocoles
d'application du RNIS-LB pour la signalisation d'accès, la
signalisation de réseau et l'interfonctionnement

Protocole de signalisation de couche AAL de type 2 –
Ensemble de capacités 1

**Annexe B: Définition en langage SDL de
l'ensemble de capacités 1 du protocole de
signalisation AAL de type 2**

Recommandation UIT-T Q.2630.1 – Annexe B

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Q
COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4 ET N° 5	Q.120–Q.249
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 6	Q.250–Q.309
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R1	Q.310–Q.399
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R2	Q.400–Q.499
COMMULATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.799
INTERFACE Q3	Q.800–Q.849
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1699
PRÉSCRIPTIONS ET PROTOCOLES DE SIGNALISATION POUR LES IMT-2000	Q.1700–Q.1799
RNIS À LARGE BANDE	Q.2000–Q.2999
Aspects généraux	Q.2000–Q.2099
Couche d'adaptation ATM de signalisation (SAAL)	Q.2100–Q.2199
Protocoles du réseau sémaphore	Q.2200–Q.2299
Aspects communs des protocoles d'application du RNIS-LB pour la signalisation d'accès, la signalisation de réseau et l'interfonctionnement	Q.2600–Q.2699
Protocoles d'application du RNIS-LB pour la signalisation de réseau	Q.2700–Q.2899
Protocoles d'application du RNIS-LB pour la signalisation d'accès	Q.2900–Q.2999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

Recommandation UIT-T Q.2630.1

Protocole de signalisation de couche AAL de type 2 – Ensemble de capacités 1

ANNEXE B

Définition en langage SDL de l'ensemble de capacités 1 du protocole de signalisation AAL de type 2

Résumé

La présente annexe contient la définition en langage SDL de l'ensemble de capacités 1 du protocole de signalisation de couche AAL de type 2 qui complète l'UIT-T Q.2630.1. Les diagrammes SDL ne sont disponibles qu'en format électronique.

Source

L'Annexe B de la Recommandation Q.2630.1 de l'UIT-T, élaborée par la Commission d'études 11 (2001-2004) de l'UIT-T, a été approuvée le 1^{er} mars 2001 selon la procédure définie dans la Résolution 1 de l'AMNT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'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.

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

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de l'AMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT n'avait pas été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

© UIT 2001

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

	Page
B.1 Introduction.....	1
B.2 Diagramme du système SDL	1
B.3 Diagramme de structure de bloc SDL	1
B.4 Spécifications SDL pour la fonction nodale	7
B.4.1 Introduction	7
B.4.2 Diagrammes SDL de la fonction nodale 1	7
B.4.3 Procédures de la fonction nodale.....	14
B.4.4 Structure des données des messages et paramètres de signalisation AAL de type 2	16
B.5 Diagrammes SDL des entités de protocole.....	45
B.5.1 Introduction	45
B.5.2 Diagrammes SDL des procédures protocolaires sortantes, entrantes et de maintenance.....	45

Recommandation UIT-T Q.2630.1

Protocole de signalisation de couche AAL de type 2 – Ensemble de capacités 1

ANNEXE B

Définition en langage SDL de l'ensemble de capacités 1 du protocole de signalisation AAL de type 2

Les définitions SDL peuvent être plus détaillées que la définition générale figurant au paragraphe 8. Cependant, si une différence technique apparaissait entre la définition de la présente annexe et celle donnée au paragraphe 8, ce serait cette dernière qui ferait foi.

B.1 Introduction

Les définitions SDL du protocole de signalisation AAL de type 2 décrit dans la présente Recommandation dépendent des diagrammes de système et de structure de bloc SDL définis dans la présente annexe.

La définition SDL donnée dans la présente annexe suppose qu'un seul événement se produise à un moment donné. La vitesse de transfert respective des différents événements à l'intérieur de l'entité de signalisation AAL de type 2 n'intervient donc pas. Il n'en demeure pas moins que la résolution des problèmes dus aux collisions entre événements et les vitesses de transfert sont fonction de l'implémentation.

B.2 Diagramme du système SDL

Le diagramme du système SDL est représenté sur la Figure B.1.

B.3 Diagramme de structure de bloc SDL

Les diagrammes de structure de bloc SDL sont représentés sur la Figure B.2 (parties 1 à 4 de 4).

NOTE 1 – Le bloc USER et le processus qui lui est associé (non représenté sur la figure) ne font pas partie de l'entité de signalisation AAL de type 2; ils sont toutefois utilisés pour indiquer les différentes entités utilisatrices desservies.

NOTE 2 – Les procédures situées dans le processus NodalF2 (*fonction nodale 2*) et auxquelles fait appel le processus NodalF1 (*fonction nodale 1*) ne sont pas développées plus avant dans la présente annexe.

NOTE 3 – Les procédures mises en œuvre dans le processus NodalF2 auxquelles le processus NodalF1 fait appel évoquent un échange de signaux implicites entre les processus NodalF1 et NodalF2.

NOTE 4 – Chacun des convertisseurs de transport de signalisation comprend une entité STI (interface de transport de signalisation). Ces convertisseurs, ainsi que leur identificateur de processus (ProcessID) (SDL) sont connus par la fonction nodale 2. L'adjonction ou la suppression de relations de signalisation, avec création ou destruction de processus STI et STC, n'est pas représentée dans les diagrammes SDL de la présente annexe.

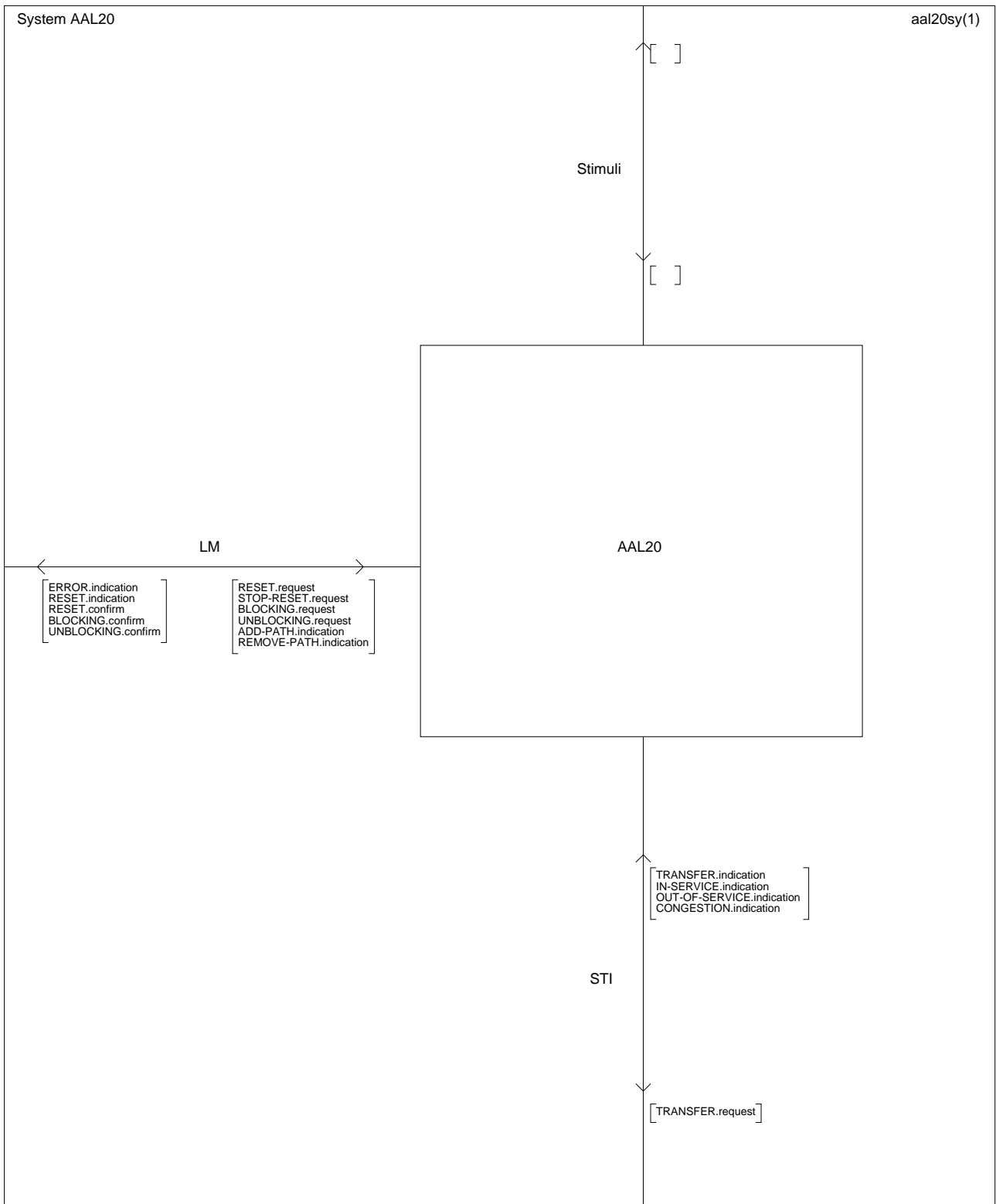


Figure B.1/Q.2630.1 – Diagramme système SDL de l'entité de signalisation AAL de type 2

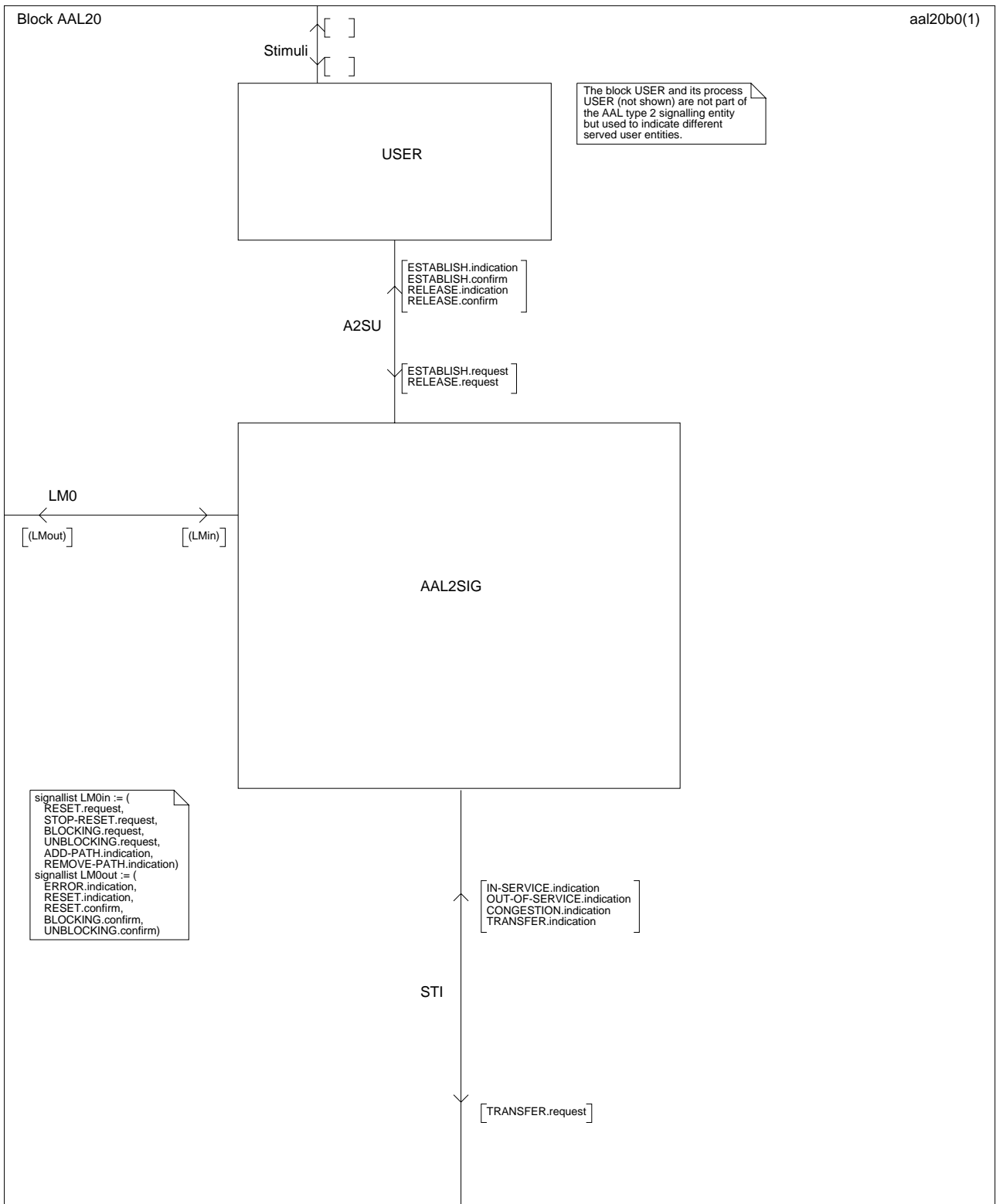


Figure B.2/Q.2630.1 – Structure bloc SDL de l'entité de signalisation AAL de type 2 (feuille 1 de 4)

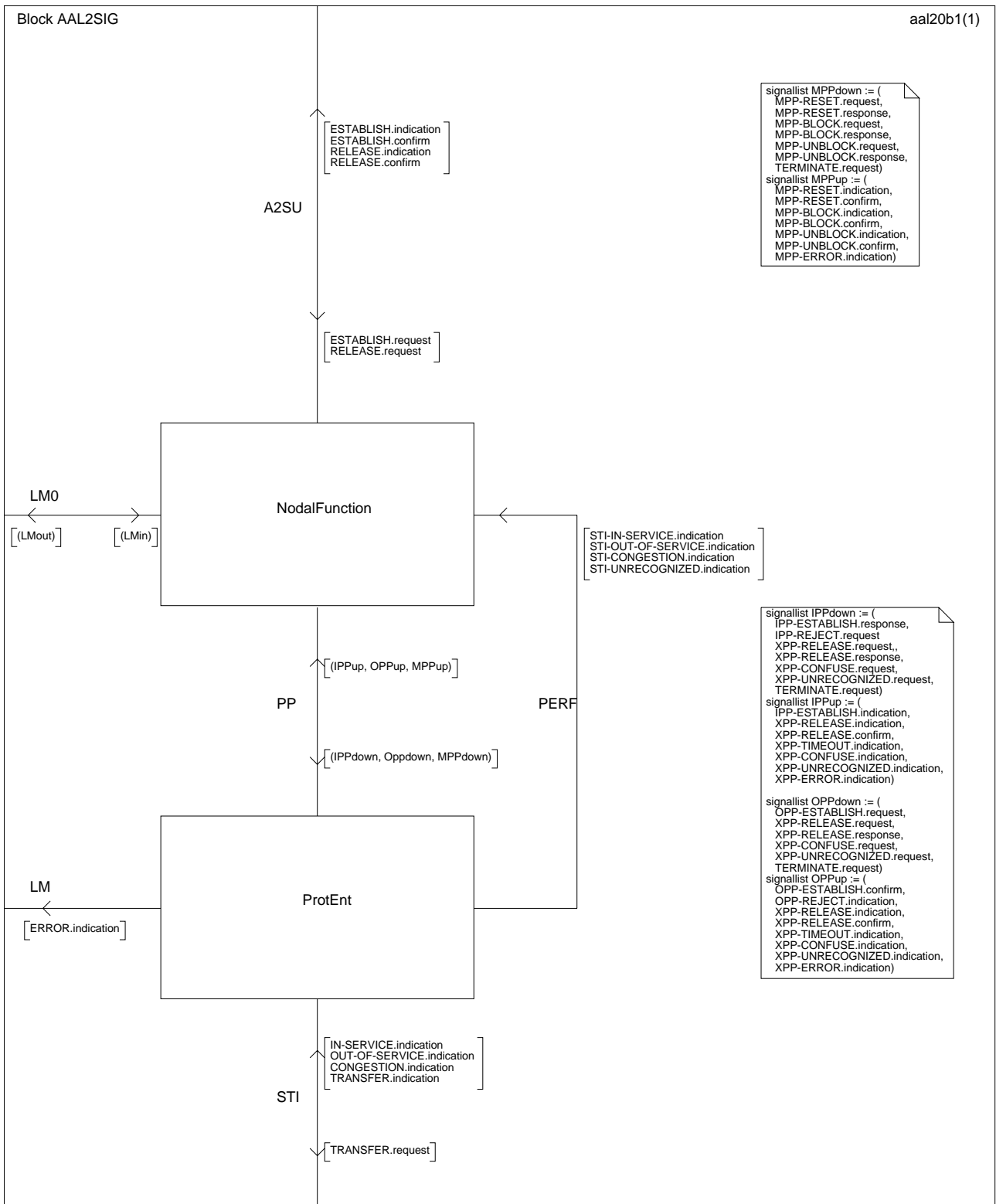


Figure B.2/Q.2630.1 – Structure bloc SDL de l'entité de signalisation AAL de type 2 (feuille 2 de 4)

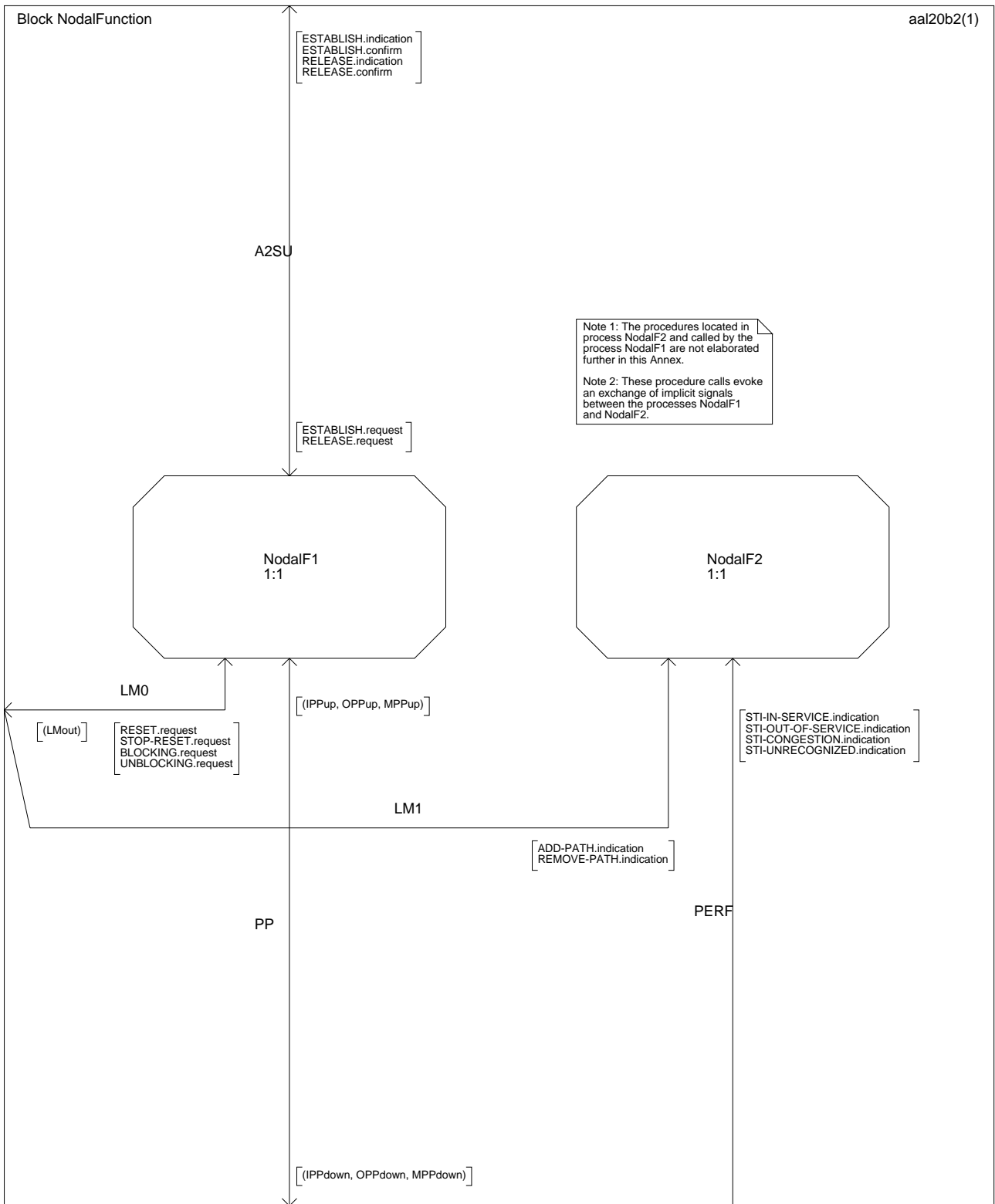


Figure B.2/Q.2630.1 – Structure bloc SDL de l'entité de signalisation AAL de type 2 (feuille 3 de 4)

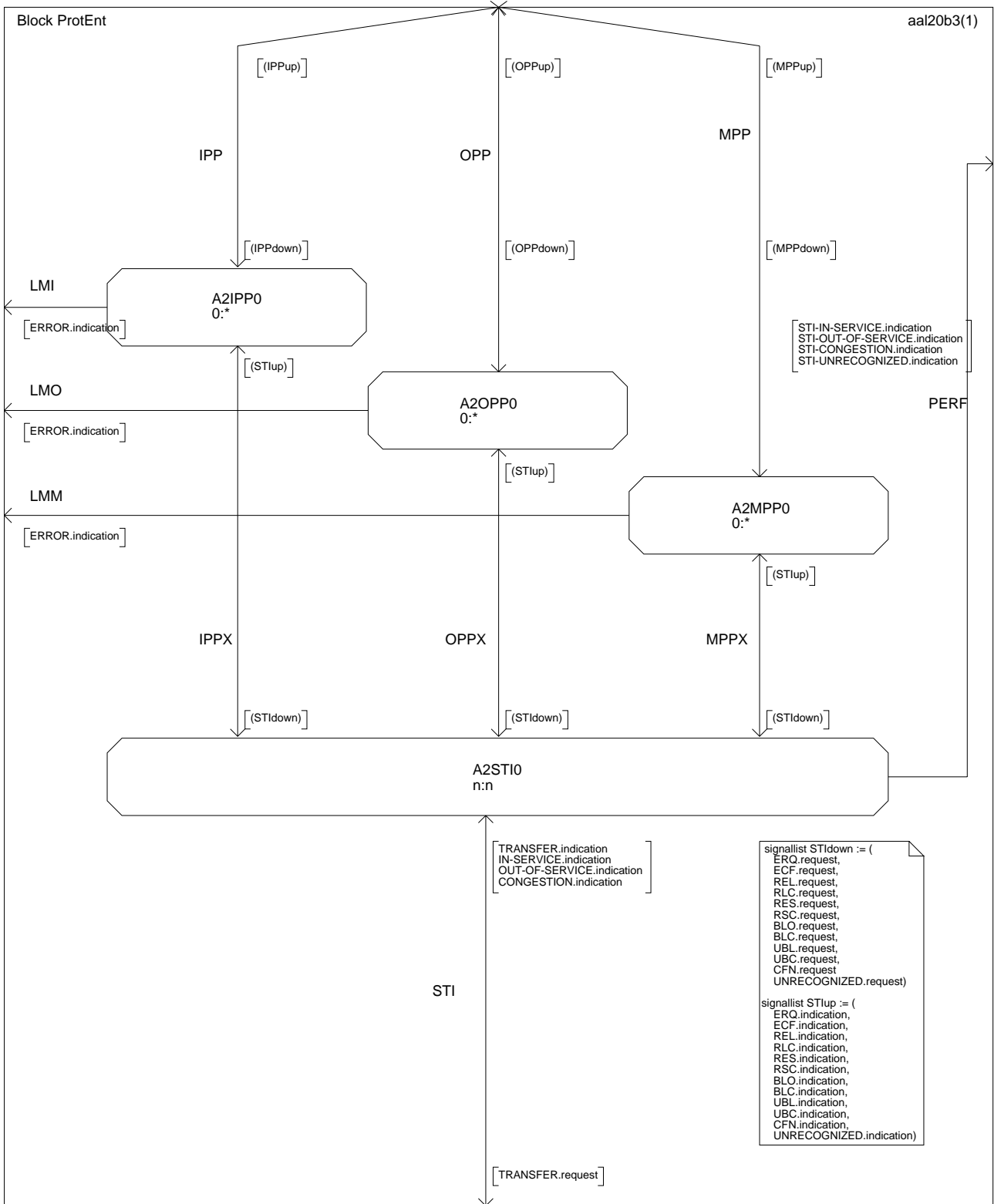


Figure B.2/Q.2630.1 – Structure bloc SDL de l'entité de signalisation AAL de type 2 (feuille 4 de 4)

B.4 Spécifications SDL pour la fonction nodale

B.4.1 Introduction

La Figure B.2 (feuillet 3 de 4) représente deux fonctions nodales distinctes, à savoir:

a) *la fonction nodale 1*

Cette entité contient toutes les fonctionnalités définies de manière détaillée au 8.2 du texte principal de la présente Recommandation et peut donc être expressément spécifiée dans les diagrammes SDL;

b) *la fonction nodale 2*

Cette entité contient toutes les fonctionnalités incorporant les parties propres à l'implémentation. Seuls les résultats sont définis de manière détaillée au 8.2 (mais non leurs mécanismes internes). Ces parties ne sont pas non plus définies de manière détaillée dans la présente annexe; seul le résultat escompté y est exposé.

Par conséquent, le présent paragraphe définit la fonction nodale 1 (voir B.4.2) et les procédures de la fonction nodale 2 (voir B.4.3). Ces dernières, de par leur nature, ne sont pas spécifiées en SDL; il convient toutefois d'avoir une idée précise des fonctionnalités que l'on attend de ces procédures pour mettre au point la définition des diagrammes SDL de l'entité de signalisation AAL de type 2.

Les processus USER (c'est-à-dire les processus intéressant l'utilisateur AAL de type 2 desservi) ne sont pas définis non plus, ni l'interaction de ces processus avec l'environnement (en ce qui concerne, par exemple, les stimuli de simulation). On part du principe qu'à chaque connexion AAL de type 2 correspond un processus USER distinct (à l'une et l'autre extrémités de la connexion) et que l'identité (de ce processus SDL) est utilisée pour router les signaux.

NOTE – Les processus des entités de protocole sont décrits au 8.3.

B.4.2 Diagrammes SDL de la fonction nodale 1

B.4.2.1 Structure des données

Pour le modèle SDL, l'entité de signalisation AAL de type 2 tient à jour un registre de type "CRec" pour chaque connexion AAL de type 2 à compter de l'instance temporelle à laquelle la connexion a été établie jusqu'à sa libération. Les termes "précédente" et "suivante" dans la suite de notre exposé se rapportent au sens d'établissement de la connexion.

Aux fins de sa consultation (au moyen de l'identificateur de l'instance d'entité de procédure protocolaire), le registre est organisé de telle manière que la partie "données entrantes" se rapporte à la liaison en provenance de laquelle un message a été reçu, comme en rendent compte les valeurs d'état indiquées dans le Tableau B.1.

Tableau B.1/Q.2630.1 – Valeurs d'état des registres CRec

Description	Valeurs d'état	La partie "données entrantes" décrit	La partie "données sortantes" décrit
Etablissement en cours	2	La liaison précédente	La liaison suivante
	3	La liaison suivante	La liaison précédente
Connexion établie	4	La liaison précédente	La liaison suivante
	5	La liaison suivante	La liaison précédente
Libération en cours	6	La liaison précédente	La liaison suivante
	7	La liaison suivante	La liaison précédente

La structure du registre de type "CRec" est définie dans le fragment ASN.1 ci-dessous:

```
CRec ::= SEQUENCE {
    status      CRecStatus,      -- état de la connexion
    incoming    HRec,            -- caractéristiques de la liaison
                                entrante
    outgoing    HRec }           -- caractéristiques de la liaison
                                sortante

HRec ::= SEQUENCE {
    peer        ENUMERATED {user,remote,none},
    ppus        PID,             -- identificateur de l'entité de
                                protocole ou de l'utilisateur
    srid        PID,             -- identificateur de la relation de
                                signalisation
    ceid        CEID }          -- identificateur d'élément de connexion
```

Pour le modèle SDL, l'entité de signalisation AAL de type 2 tient à jour un registre de type "MRec" pour chaque action de maintenance de liaison sortante à compter de l'instance temporelle à laquelle l'action de maintenance a été lancée et jusqu'à ce qu'elle prenne fin. La structure du registre MRec est définie dans le fragment ASN.1 ci-dessous:

```
MRec ::= SEQUENCE {
    status      MRecStatus,      -- état de l'action de maintenance
    ppus        PID,             -- identificateur d'entité de protocole
    srid        PID,             -- l'identificateur de la relation de
                                signalisation est l'identificateur de
                                l'entité de transport de
                                signalisation
    ceid        CEID,           -- identificateur d'élément de connexion
    disp        BOOLEAN }       -- la TRUE: l'expéditeur est la gestion
                                de couche
```

B.4.2.2 Primitives entre la fonction nodale 1 et les entités de protocole

L'interface avec l'utilisateur AAL de type 2 desservi est définie au 5.1. L'interface avec la gestion de couche est définie au 5.3. Les primitives et les paramètres échangés entre l'entité de signalisation AAL de type 2 et les entités de protocole AAL de type 2 sont récapitulés dans le Tableau B.2 et définis après ce tableau.

Tableau B.2/Q.2630.1 – Primitives et paramètres échangés entre l'entité de la fonction nodale 1 et les entités de protocole

Primitive Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
IPP-ESTABLISH	Non définie	ERQmsg, sri	ECFmsg	Non définie
IPP-REJECT	RLCmsg	Non définie	Non définie	Non définie
OPP-ESTABLISH	ERQmsg	Non définie	Non définie	ECFmsg
OPP-REJECT	Non définie	RLCmsg	Non définie	Non définie
XPP-RELEASE	RELmsg	RELmsg	RLCmsg	RLCmsg
XPP-TIMEOUT	Non définie	–	Non définie	Non définie
XPP-CONFUSE	CNFmsg	CNFmsg	Non définie	Non définie

Tableau B.2/Q.2630.1 – Primitives et paramètres échangés entre l'entité de la fonction nodale 1 et les entités de protocole (*fin*)

Primitive Nom générique	Type			
	Demande	Indication	Réponse	Confirmation
XPP-UNRECOGNIZED	msg	msg	Non définie	Non définie
MPP-BLOCK	BLOmsg, sqc	BLOmsg, sri	BLCmsg	BLCmsg
MPP-UNBLOCK	UBLmsg	UBLmsg, sri	UBCmsg	UBCmsg
MPP-RESET	RESmsg	RESmsg, sri	RSCmsg, sqc	RSCmsg
MPP-ERROR	Non définie	cause	Non définie	Non définie
TERMINATE	–	Non définie	Non définie	Non définie
– Cette primitive n'a pas de paramètres.				

a) **IPP-ESTABLISH**

Une entité de protocole d'arrivée récemment créée indique le message de demande d'établissement (ERQ, *establish request message*) à la fonction nodale 1 (au moyen de la primitive d'indication) ainsi que l'identificateur d'association de signalisation nodale "sri". Si l'établissement permet de desservir un utilisateur relevant de ce nœud ou si les nœuds AAL de type 2 situés au-delà de la liaison suivante ont accusé réception de l'établissement, le message de confirmation d'établissement (ECF, *establish confirm message*) est communiqué à l'entité de protocole d'arrivée (au moyen de la primitive de réponse) pour transmission sur la liaison précédente.

b) **IPP-REJECT**

S'il ne peut être mené à bien au niveau de ce nœud AAL de type 2 ou de tout autre nœud AAL de type 2 situé au-delà de la liaison suivante, l'établissement est rejeté moyennant l'envoi d'un message de confirmation de libération (RLC, *release confirm message*) à l'entité de protocole d'arrivée (au moyen de la primitive de demande) pour transmission sur la liaison précédente.

c) **OPP-ESTABLISH**

Si une opération d'établissement doit être poursuivie sur une liaison suivante, le message de demande d'établissement (ERQ) est envoyé à l'entité de protocole de départ récemment créée (au moyen de la primitive de demande). Si le message de confirmation d'établissement (ECF) est reçu par cette entité de protocole de départ, il est communiqué à la fonction nodale 1 (au moyen de la primitive de confirmation).

d) **OPP-REJECT**

(Si les nœuds AAL de type 2 situés au-delà de la liaison suivante ne sont pas en mesure d'accepter l'établissement, l'entité de protocole de départ reçoit un message de confirmation de libération RLC) qui est communiqué à la fonction nodale 1 (au moyen de la primitive d'indication).

e) **XPP-RELEASE**

L'entité de protocole d'arrivée ou de départ est chargée de lancer les procédures de libération au moyen d'un message de demande de libération (REL, *release request message*). Ce message est communiqué à l'entité de protocole (au moyen de la primitive de demande). Si une entité de protocole d'arrivée ou de départ reçoit un tel message de demande de libération (REL), ce message est envoyé à la fonction nodale 1 (au moyen de la primitive d'indication).

Après réception du message de demande de libération (REL), la fonction nodale libère les ressources et confirme la libération au moyen d'un message de confirmation de libération (RLC); ce message est envoyé à l'entité de protocole (au moyen de la primitive de réponse). Si une entité de protocole d'arrivée ou de départ reçoit un message de confirmation de libération (RLC), ce message est envoyé à la fonction nodale 1 (au moyen de la primitive de confirmation).

f) **XPP-TIMEOUT**

A différentes phases, une entité de protocole d'arrivée ou de départ actionne une temporisation. Si cette temporisation vient à expirer, la fonction nodale 1 en est informée (au moyen de la primitive d'indication); aucun paramètre ne doit être envoyé.

g) **XPP-CONFUSE**

Si deux nœuds AAL de type 2 d'un niveau fonctionnel différent communiquent, entre eux, il se peut que les informations transmises par l'un de ces nœuds ne soient pas intégralement comprises par l'autre. En pareil cas, le mécanisme de compatibilité peut exiger la transmission d'un message "confusion" (CNF, *confuse message*); ce message est envoyé à l'entité de protocole d'arrivée ou de départ (au moyen de la primitive de demande) pour transmission. A la réception d'un message "confusions" (CNF) par une entité de protocole d'arrivée ou de départ, ce message est envoyé à la fonction nodale 1 (au moyen de la primitive d'indication).

h) **XPP-UNRECOGNIZED**

Si deux nœuds AAL de type 2 d'un niveau fonctionnel différent communiquent entre eux, il se peut que les informations transmises par l'un de ces nœuds ne soient pas intégralement comprises par l'autre. En pareil cas, un message non reconnu peut être reçu par une entité de protocole d'arrivée ou de départ; ce message est communiqué à la fonction nodale 1 (au moyen de la primitive d'indication). Le mécanisme de compatibilité peut exiger la retransmission de ce message sur une autre liaison; le message non reconnu est transmis à l'entité de protocole d'arrivée ou de départ (au moyen de la primitive de demande) pour transmission.

i) **MPP-BLOCK**

Une entité de protocole de maintenance récemment créée est chargée de transmettre un message de demande de blocage (BLO, *block request*) (au moyen de la primitive de demande). Lorsqu'elle reçoit un tel message de demande de blocage (BLO), une entité de protocole de maintenance récemment créée envoie ce message à la fonction nodale 1 (au moyen de la primitive d'indication). Le message de confirmation de blocage (BLC, *block confirm message*) à renvoyer est communiqué à l'entité de protocole de maintenance (au moyen de la primitive de réponse) pour transmission. Si une entité de protocole de maintenance reçoit un message de confirmation de blocage (BLC, *block confirm message*), ce message est communiqué à la fonction nodale 1 (au moyen de la primitive de confirmation).

j) **MPP-UNBLOCK**

Une entité de protocole de maintenance récemment créée est chargée de transmettre un message de demande de déblocage (UBL) (au moyen de la primitive de demande). Lorsqu'elle reçoit un tel message de demande de déblocage (UBL, *unblock request message*), une entité de protocole de maintenance récemment créée envoie ce message à la fonction nodale 1 (au moyen de la primitive d'indication). Le message de confirmation de déblocage (UBC, *unblock confirm message*) à renvoyer est communiqué à l'entité de protocole de maintenance (au moyen de la primitive de réponse) pour transmission. Si une entité de protocole de maintenance reçoit un message de confirmation de déblocage (UBC), ce message est envoyé à la fonction nodale 1 (au moyen de la primitive de confirmation).

k) **MPP-RESET**

Une entité de protocole de maintenance récemment créée est chargée de transmettre un message de demande de réinitialisation (RES, *reset request message*) (au moyen de la primitive de demande). Lorsqu'elle reçoit un tel message de demande de réinitialisation (RES), une entité de protocole de maintenance récemment créée envoie ce message à la fonction nodale 1 (au moyen de la primitive d'indication). Le message de confirmation de réinitialisation (RSC, *reset confirm message*) à renvoyer est communiqué à l'entité de protocole de maintenance (au moyen de la primitive de réponse) pour transmission. Si une entité de protocole de maintenance reçoit un message de confirmation de réinitialisation (RSC), ce message est envoyé à la fonction nodale 1 (au moyen de la primitive de confirmation).

l) **MPP-ERROR**

Les erreurs détectées par une entité de protocole de maintenance sont signalées à la fonction nodale 1 (au moyen de la primitive d'indication); ces erreurs comprennent les fins de temporisation.

m) **TERMINATE**

A tout moment, la fonction nodale 1 peut supprimer une entité de protocole de maintenance (au moyen de la primitive de demande); aucun paramètre ne doit être envoyé.

Outre les messages de signalisation AAL de type 2, les paramètres suivants sont envoyés:

aa) **sri**

Le paramètre "sri" est du type PID (identificateur de processus SDL); utilisé dans des primitives d'indication, il indique la relation de signalisation nodale.

ab) **sqc**

Le paramètre "sqc" est utilisé dans les primitives de demande MPP-BLOCK et de réponse MPP-RESET pour assurer l'intégrité de la séquence des messages de demande de blocage (BLO) et de confirmation de réinitialisation (RSC).

ac) **cause**

Le paramètre "cause" est utilisé pour indiquer le type d'erreur signalée par une entité de protocole de maintenance.

ad) **msg**

Le paramètre "msg" contient un message de signalisation AAL de type 2 non reconnu complet.

La réaction aux événements de signaux d'entrée est décrite dans les feuillets 1 à 15 (de 25) de la Figure B.3.

B.4.2.3 Procédures

Les procédures sont décrites dans les feuillets 16 à 21 (de 25) de la Figure B.3.

Les fonctions "**Compatibility**" (feuillets 19 et 20 de la Figure B.3) et "**MsgCompatibility**" (feuille 21 de la Figure B.3) assurent le contrôle de compatibilité des paramètres (dans les messages reconnus) et décident de l'action de compatibilité à appliquer aux messages non reconnus; elles renvoient une valeur définie par la structure ASN.1 suivante:

```
Compat ::= SEQUENCE {
    course          ENUMERATED {pass,passcnf,dcrd,dcrdcnf,release},
    cause          CAUSE }
-- renvoyer la valeur de la procédure de
-- compatibilité
-- cause si autre que "pass (transfert)"
```

L'action prend les valeurs suivantes:

- pass (transfert): les paramètres sont tous reconnus ou peuvent être transférés; les paramètres non reconnus qui doivent être ignorés sont supprimés du message à l'aide de cette fonction;
- passcnf: le message contenait des paramètres non reconnus qui ont été supprimés du message à l'aide de cette fonction; une notification a été demandée;
- dcrd: le message contenait des paramètres non reconnus; la non-reconnaissance (d'au moins un) des paramètres a rendu nécessaire la mise à l'écart de la totalité du message;
- dcrdcnf: même chose que pour "dcrd", mais avec en plus demande de notification;
- libération: le message contenait des paramètres non reconnus; la non-reconnaissance (d'au moins un) des paramètres a rendu nécessaire la libération de la connexion.

NOTE – La valeur "passcnf" n'est pas renvoyée par "MsgCompatibility".

La fonction "**LookupCRec**" (feuillet 16 de la Figure B.3) explore tous les registres de type "CRec" pour trouver celui dont le paramètre d'entrée correspond au paramètre `crec.incoming.ppus` ou au paramètre `crec.outgoing.ppus`; en outre, la partie `crec...peer` doit indiquer la valeur "remote". Un seul registre réunissant toutes ces conditions est trouvé.

Si le paramètre d'entrée correspond au paramètre `crec.outgoing.ppus`, les parties "données entrantes" et "données sortantes" des registres sont échangées. A l'issue d'un tel échange, la partie indiquant l'état du registre (`crec.status`) est également modifiée comme suit:

- si la valeur (`crec.status`) est paire,
 - augmenter cette valeur de 1
- sinon
 - diminuer la valeur `crec.status` de 1
- mettre fin à cette fonction si

La valeur renvoyée peut être comprise comme donnant une indication sur le registre proprement dit.

La fonction "**LookupMRec**" (feuillet 16 de la Figure B.3) explore tous les registres de type "MRec" pour trouver celui qui correspond au paramètre `mrec.ppus`. Un seul registre correspondant à ce paramètre est trouvé. La valeur renvoyée peut être interprétée comme donnant une indication sur le registre proprement dit.

La fonction "**FindNextCRec**" (feuillet 16 de la Figure B.3) explore tous les registres de type "CRec" pour trouver le prochain qui correspondra aux paramètres "ceid" et "sri". La valeur renvoyée peut être interprétée comme donnant une indication sur le registre proprement dit, sauf s'il n'est pas trouvé d'autres registres (auquel cas la valeur "null" est renvoyée). Avant de renvoyer l'indication, on procède au réglage du contenu du registre comme indiqué précédemment pour la procédure "LookupCRec".

La fonction "**FindNextPath**" (feuillet 16 de la Figure B.3) explore tous les conduits attribués de la relation de signalisation indiquée par le paramètre "sri": la valeur renvoyée est la valeur de l'identificateur de conduit AAL de type 2, sauf s'il n'est pas trouvé d'autres conduits (auquel cas la valeur "null" est renvoyée).

La fonction "**GetNextParam**" (feuillet 16 de la Figure B.3) analyse le message et localise le paramètre suivant. La valeur renvoyée est le paramètre (ou s'y réfère) sauf s'il n'est pas trouvé d'autres paramètres (auquel cas la valeur "null" est renvoyée).

La fonction "**GetNextField**" (feuillet 16 de la Figure B.3) analyse le paramètre et localise le champ suivant. La valeur renvoyée est le champ (ou s'y réfère), sauf s'il n'existe pas d'autres champs (auquel cas la valeur "null" est renvoyée).

La fonction "**allocate**" et la procédure "**release**" (feuillet 16 de la Figure B.3) tiennent lieu de système de gestion de mémoire qui attribue et libère les registres de type "CRec" et "MRec".

La procédure "**StartReset**" (feuillet 17 de la Figure B.3) fait appel à la procédure "ResetConn" pour libérer les connexions AAL de type 2 visées par la réinitialisation. Si plusieurs canaux sont réinitialisés, il est fait appel à la procédure "ResetUnblock" pour positionner tous les conduits AAL de type 2 visés sur "déblocage distant". Un message de demande de réinitialisation (RES) est créé et soumis à une entité de protocole de gestion récemment créée. Avant de renvoyer la valeur requise, un nouveau registre de type "MRec" est attribué et rempli.

La procédure "**StartBlocking**" (feuillet 17 de la Figure B.3) fait appel à la procédure "BLOCKING" pour enregistrer le blocage demandé avant de créer un message de demande de blocage (BLO, *blocking message*) et de soumettre ce message à une entité de protocole de gestion récemment créée. Avant de renvoyer la valeur requise, un nouveau registre de type "MRec" est attribué et rempli.

La procédure "**StartUnblocking**" (feuillet 17 de la Figure B.3) vérifie (en faisant appel à la procédure "Blocked") si le ou les conduits indiqués sont déjà à l'état "déblocage local"; si tel est le cas, la primitive de confirmation de déblocage UNBLOCK.confirm est émise, ce qui met fin à l'état de "déblocage". Sinon, un message de demande de déblocage (UBL, *unblocking message*) est créé et soumis à une entité de protocole de gestion récemment créée. Avant de renvoyer la valeur requise, un nouveau registre de type "MRec" est attribué et rempli.

La procédure "**ResetConn**" (feuillet 18 de la Figure B.3) fait appel à la fonction "FindNextCRec" pour trouver les liaisons AAL de type 2 visées par l'opération de réinitialisation. Chacune de ces liaisons aboutit à l'entité de protocole associée. A une extrémité du service AAL de type 2, une primitive de confirmation de libération RELEASE.confirm est envoyée à l'utilisateur desservi. Dans un commutateur AAL de type 2, les procédures de libération lancées concernent l'utilisateur desservi distant. On s'assure que les ressources attribuées à ces liaisons soient libérées; les ressources du ou des conduits AAL de type 2 qui sont réinitialisées sont libérées par la procédure de réinitialisation proprement dite.

La procédure "**ResetBlock**" (feuillet 18 de la Figure B.3) fait en sorte que pour tous les conduits AAL de type 2 se trouvant à l'état "blocage local" et visés par une opération de réinitialisation (lancée par un autre nœud) une procédure de blocage soit lancée (avant que la réinitialisation ne soit confirmée).

La procédure "**ResetUnblock**" (feuillet 18 de la Figure B.3) positionne tous les conduits AAL de type 2 se trouvant à l'état "blocage distant" et visés par l'opération de réinitialisation sur la position "déblocage distant".

B.4.2.4 Macros

Les macros sont décrites dans les feuillets 22 à 25 (de 25) de la Figure B.3.

La macro "**Construct ERQmsg**" (feuillet 22 de la Figure B.3) indique les caractéristiques nécessaires pour la création du message ERQ. En particulier, des paramètres sont ajoutés au message en fonction des paramètres figurant dans la primitive de demande d'établissement ESTABLISH émanant de l'utilisateur desservi.

La macro "**ReturnConfuse**" (feuillet 23 de la Figure B.3) renvoie un message "confusion" (CNF) à l'expéditeur du dernier message à l'origine de la confusion.

La macro "**ReturnReject**" (feuillet 23 de la Figure B.3) crée un message de confirmation de libération (RLC) comportant un paramètre cause (CAU) et le renvoie à l'expéditeur du message de demande d'établissement (ERQ); la macro "**SendReject**" crée un message de confirmation de libération (RLC) comportant un paramètre cause (CAU) et l'envoie à l'expéditeur du message de demande d'établissement (ERQ), une fois que le message ERQ a été traité, par exemple si la liaison suivante a été réinitialisée.

La macro "**ReturnRelease**" (feuillet 23 de la Figure B.3) crée un message de libération (REL) et le renvoie à l'expéditeur du dernier message; ce message doit être envoyé en réponse à un message non reconnu ou à un message qui a donné lieu à confusion. La macro "**SendRelease**" crée un message de libération (REL) et l'envoie sur la liaison précédente ou suivante; ce message peut être envoyé en réponse à un message non reconnu ou à l'occasion d'une opération de réinitialisation.

La macro "**Extract ERQparameters**" (feuillet 24 de la Figure B.3) extrait les informations figurant dans un message de demande d'établissement (ERQ) et prépare les paramètres de la primitive d'indication d'établissement ESTABLISH.

La macro "**ValidREL**" (feuillet 24 de la Figure B.3) fait en sorte qu'un paramètre cause (CAU) figure dans le message de libération (REL); si ce paramètre est absent, il est ajouté dans cette macro ("normal, non spécifié", pas de diagnostic). La macro "**ValidRLC**" vérifie si un paramètre cause (CAU) figure dans le message de fin de libération (RLC) lorsque aucun paramètre cause n'est prévu; si ce paramètre est présent, la cause (découlant d'une confusion) est communiquée à la gestion de couche. La macro "**ValidRLCR**" fait en sorte qu'un paramètre cause (CAU), lorsqu'il en est prévu un, figure dans le message de fin de libération (RLC); si ce paramètre est absent, il est ajouté dans cette macro ("défaillance temporaire", pas de diagnostic). La macro "**ValidCNF**" vérifie si un paramètre cause (CAU) figure dans le message "confusion" (CNF); aucun paramètre n'est ajouté en pareil cas.

Les macros "**Construct ECFmsg**", "**Construct RLCmsg**", "**Construct RSCmsg**", "**Construct BLCmsg**", et "**Construct UBCmsg**" (feuillet 25 de la Figure B.3) indiquent que les messages correspondants créés ne comportent pas de paramètres.

Les macros "**Construct RESmsg**", "**Construct BLOmsg**", et "**Construct UBLmsg**" (feuillet 25 de la Figure B.3) indiquent que les messages correspondants créés comportent un paramètre CEID, identificateur d'élément de connexion (CEID, *connection element identifier*).

Les macros "**Construct RELmsg**", "**Construct RLCmsgR**", "**Construct RSCmsgC**", "**Construct BLCmsgC**", "**Construct UBCmsgC**", "**Construct RELmsg**", et "**Construct CNFmsg**" (feuillet 25 de la Figure B.3) indiquent que les messages correspondants créés comportent un paramètre cause (CAU).

NOTE – Le message de libération (REL) ou confusion (CNF) contient toujours un paramètre cause. Le message de confirmation de libération (RLC) contient un paramètre cause en réponse à un message de demande d'établissement (ERQ) (rejet de l'établissement) ou en réponse à un message de libération associé à des notifications de compatibilité. Les messages de confirmation de réinitialisation (RSC), de confirmation de blocage (BLC) et de confirmation de déblocage (UBC) contiennent un paramètre cause associé à des notifications de compatibilité.

B.4.3 Procédures de la fonction nodale

La fonction "**PathRes**" effectue le contrôle d'admission de la connexion puis la réservation des ressources sur une liaison entrante (la liaison précédente) pendant l'établissement de la connexion; elle renvoie une valeur définie par la structure ASN.1 suivante:

```

PathRes ::= SEQUENCE {
    course          ENUMERATED {success, fail},
    cause          CAUSE }
-- renvoyer la valeur de la procédure
-- PathResource
-- cause si l'action échoue
-- ("course = fail")

```

NOTE – Le contrôle d'admission de la connexion et la réservation des ressources ne sont pas spécifiés de manière détaillée dans la présente Recommandation.

La fonction "**SelectRoute**" effectue une décision de routage puis la réservation des ressources sur la liaison sortante (la liaison suivante) pendant l'établissement de la connexion.

La fonction nodale détermine l'existence d'une route disposant de suffisamment de ressources par un conduit AAL de type 2 jusqu'au nœud AAL de type 2 suivant. Cette fonction peut comporter une fonction de sélection d'une route disposant de ressources utilisables jusqu'au nœud AAL de type 2 suivant. Elle sélectionne ensuite, d'après cette route, un conduit AAL de type 2 à même de prendre en charge la nouvelle connexion.

Le routage est généralement assuré d'après:

- les informations d'adressage (dans le cas du mode commutation);
- l'indicateur connexion d'essais;
- les informations relatives à la liaison (caractéristiques de la liaison);
- d'autres informations [dont l'information de sous-couche de convergence propre au service (SSCS)].

Cette fonction renvoie une valeur définie par la structure ASN.1 suivante:

```
Route ::= SEQUENCE {
    course          ENUMERATED {remote,local,fail},
    ceid            CEID,
    sri             PID,
    cause           CAUSE }
-- renvoyer la valeur de la
-- procédure SelectRoute
-- identificateur d'élément de
-- connexion
-- identificateur d'association de
-- signalisation nodale
-- cause en cas d'échec ("fail")
```

La fonction "**SwitchRoute**" (interne à la fonction nodale) effectue une décision de routage puis la réservation des ressources internes d'un nœud AAL de type 2.

Cette route est établie entre l'utilisateur AAL de type 2 desservi demandeur ou la liaison entrante (précédente), d'une part, et l'utilisateur AAL de type 2 desservi de destination ou la liaison sortante (suivante), d'autre part, pendant l'établissement de la connexion. Cette fonction renvoie une valeur qui est définie par la structure ASN.1 suivante:

```
Switch ::= SEQUENCE {
    course          ENUMERATED {success,fail},
    cause           CAUSE }
-- renvoyer la valeur de la
-- procédure SwitchRoute
-- cause en cas d'échec
-- ("fail")
```

La procédure "**PathRel**" libère les ressources associées à un conduit AAL de type 2; ces ressources sont désignées par le paramètre "CRec.incoming" ou "CRec.outgoing".

La procédure "**SwitchRel**" libère les ressources associées à une connexion AAL de type 2 interne à un nœud AAL de type 2.

La procédure "**ResetRel**" libère les ressources de connexion AAL de type 2 associées au canal (à un des canaux) qui fait (font) l'objet d'une procédure de maintenance de réinitialisation.

La procédure "**AddCompatibility**" complète un message en ajoutant l'information de compatibilité appropriée au champ de compatibilité de ce message ainsi qu'à tous les champs de compatibilité des paramètres.

La fonction "**PassOnPossible**" renvoie la valeur "TRUE" si le transfert du message non reconnu ou d'un paramètre non reconnu d'un message reconnu est possible; sinon, la valeur "FALSE" est renvoyée.

La fonction "**ParamKnown**" renvoie la valeur "TRUE" si le paramètre est reconnu; sinon la valeur "FALSE" est renvoyée.

La fonction "**FieldRecognized**" renvoie la valeur "TRUE" si la valeur du champ est reconnue; sinon, la valeur "FALSE" est renvoyée.

La fonction "**PassConfuse**" renvoie la valeur "TRUE" si un message CNF reçu doit être transféré; sinon, la valeur "FALSE" est renvoyée.

La fonction "**Valid**" renvoie la valeur "TRUE" si l'identificateur CEID sur une relation de signalisation donnée est valable; sinon, la valeur "FALSE" est renvoyée.

La fonction "**ValidateLink**" détermine si l'information CEID figurant dans le message ERQmsg désigne une liaison valable de l'association de signalisation nodale "sri". Elle renvoie la valeur "TRUE" si l'information est valable (sinon, elle renvoie la valeur "FALSE").

La procédure "**BLOCKING**" positionne un conduit (d'une relation de signalisation) sur "blocage ou déblocage local" et "blocage ou déblocage distant".

La fonction "**Blocked**" renvoie la valeur "TRUE" si le conduit AAL de type 2 indiqué est positionné sur "blocage local ou distant" (comme indiqué dans le deuxième paramètre); sinon, la valeur "FALSE" est renvoyée.

B.4.4 Structure des données des messages et paramètres de signalisation AAL de type 2

Les diagrammes SDL utilisent les structure et définition ASN.1 pour les messages et paramètres de signalisation AAL de type 2.

B.4.4.1 Structure des messages et paramètres

```

Message ::= SEQUENCE {
    dsaid          SAID,          -- identificateur
                                d'association de
                                signalisation de
                                destination
    msgID          MessageID,    -- identificateur de
                                message
    msgcompat      BIT STRING (SIZE (8)), -- compatibilité des
                                message
    parameters     SET OF Parameter } -- paramètres de message

```

```

Parameter ::= SEQUENCE {
    paramid       BIT STRING (SIZE (8)), -- identificateur de
                                paramètre
    paramcompat   BIT STRING (SIZE (8)), -- compabilité des
                                paramètres
    paramlength   INTEGER (0 .. 255),   -- longueur de paramètre
    fields        SEQUENCE OF Fields }  -- champs de paramètre

```

```

Fields ::= CHOICE {
    fldtyp1       FixedSizeField,
    fldtyp2       VariableSizeField }
FixedSizeField ::= fixedfield OCTET STRING (SIZE (1 .. 255))
VariableSizeField ::= SEQUENCE {
    fieldlength   INTEGER (0 .. 254),   -- longueur
                                de champ
    variablefield OCTET STRING (SIZE (0 .. 254)) }

```

MessageID	BIT STRING (SIZE (8))	-- identificateur de message
erq	MessageID ::= '00000101'H	-- Demande d'établissement
ecf	MessageID ::= '00000100'H	-- Confirmation d'établissement
rel	MessageID ::= '00000111'H	-- Demande de libération
rlc	MessageID ::= '00000110'H	-- Confirmation de libération
res	MessageID ::= '00001001'H	-- Demande de réinitialisation
rsc	MessageID ::= '00001000'H	-- Confirmation de réinitialisation
blo	MessageID ::= '00000010'H	-- Demande de blocage
blc	MessageID ::= '00000001'H	-- Confirmation de blocage
ubl	MessageID ::= '00001011'H	-- Demande déblocage
ubc	MessageID ::= '00001010'H	-- Confirmation de déblocage
cnf	MessageID ::= '00000011'H	-- Confusion

B.4.4.2 Structure détaillée des paramètres

CAU ::= SEQUENCE {	-- définition des éléments essentiels du paramètre cause (CAU)	
org	BIT STRING (SIZE (2)) ('00'B),	-- origine UIT-T
value	INTEGER (1 .. 127),	-- valeur de cause
diagnostics	OCTET STRING (SIZE (0 .. 252)) }	

CEID ::= SEQUENCE {	-- définition des éléments essentiels du paramètre CEID	
path	AAL2Path,	-- conduit AAL de type 2
cid	CID }	-- identificateur de canal (CID)

OSAID ::=	SAID	-- définition du paramètre OSAID
SAID ::=	OCTET STRING (SIZE (4))	-- définition du paramètre SAID
unknown	SAID ::= '00000000'H	-- définition de la valeur "inconnu"

TCI ::= OCTET STRING (SIZE (0))	-- définition des éléments essentiels du paramètre TCI
---------------------------------	---

```

-- Les paramètres suivants sont traités dans la définition SDL mais n'y sont
  jamais interprétés; ils n'appellent donc pas de précisions
ESEA ::= OCTET STRING (SIZE (3 .. 17))    -- définition des éléments
                                          essentiels du paramètre ESEA
NSEA ::= OCTET STRING (SIZE (20))         -- définition des éléments
                                          essentiels du paramètre NSEA
ALC  ::= OCTET STRING (SIZE (12))         -- définition des éléments
                                          essentiels du paramètre ALC
SUGR ::= OCTET STRING (SIZE (4))          -- définition des éléments
                                          essentiels du paramètre SUT
SUT  ::= OCTET STRING (SIZE (1 .. 255))  -- définition des éléments
                                          essentiels du paramètre SUGR
SSIA ::= OCTET STRING (SIZE (8))         -- définition des éléments
                                          essentiels du paramètre SSIA
SSIM ::= OCTET STRING (SIZE (3))         -- définition des éléments
                                          essentiels du paramètre SSIM
SSISA ::= OCTET STRING (SIZE (14))       -- définition des éléments
                                          essentiels du paramètre SSISA
SSISU ::= OCTET STRING (SIZE (7))        -- définition des éléments
                                          essentiels du paramètre SSISU

```

NOTE – Les paramètres non interprétés par la fonction nodale 1 ne sont représentés que de manière succincte; toutefois, à l'exception des paramètres "SUGR" et "SUT", ces paramètres sont interprétés par la fonction nodale 2.

B.4.4.3 Structure récapitulative détaillée des paramètres des messages

```

ERQmsg ::= SEQUENCE {
  ceid          CEID,
  a2ea          CHOICE {
    esea        ESEA,
    nsea        NSEA },
  alc           ALC OPTIONAL,
  osaid         OSAID,
  sugr          SUGR OPTIONAL,
  sut           SUT OPTIONAL,
  ssis         CHOICE {
    ssia        SSIA,
    ssim        SSIA,
    ssisa       SSIA,
    ssisu       SSIA } OPTIONAL,
  tci          TCI OPTIONAL }
-- définition des éléments essentiels du message ERQ
-- identificateur d'élément de
-- connexion
-- adresse de point d'extrémité AAL
-- de type 2
-- adresse de point
-- d'extrémité E.164 de destination
-- adresse de point
-- d'extrémité NSAP de destination
-- caractéristiques des liaisons
-- identificateur d'association
-- de signalisation d'origine
-- référence générée par
-- l'utilisateur desservi
-- transport d'utilisateur desservi
-- information SSCS
-- informations propres au service
-- (audio)
-- informations propres au service
-- (multidébit)
-- informations propres au service
-- (SAR assuré)
-- informations propres au service
-- (SAR non assuré)
-- indicateur de connexion d'essais

```

```

ECFmsg ::= SEQUENCE {
  osaid         OSAID }
-- définition des éléments essentiels du message ECF
-- identificateur d'association
-- de signalisation d'origine

```

```

RELmsg ::= SEQUENCE {
  cause        CAU }
-- définition des éléments essentiels du message REL
-- cause

```



```
RLCmsg ::= SEQUENCE { -- définition des éléments essentiels du message RLC
  cause          CAU OPTIONAL } -- cause
```

```
RESmsg ::= SEQUENCE { -- définition des éléments essentiels du message RES
  ceid          CEID, -- identificateur d'élément de
                   connexion
  osaid         OSAID } -- identificateur d'association de
                   signalisation d'origine
```

```
RSCmsg ::= SEQUENCE { -- définition des éléments essentiels du message RSC
  cause          CAU OPTIONAL } -- cause
```

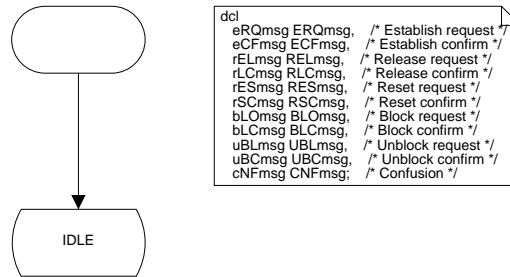
```
BLOmsg ::= SEQUENCE { -- définition des éléments essentiels du message BLO
  ceid          CEID, -- identificateur d'élément de
                   connexion
  osaid         OSAID } -- identificateur d'association de
                   signalisation d'origine
```

```
BLCmsg ::= SEQUENCE { -- définition des éléments essentiels du message BLC
  cause          CAU OPTIONAL } -- cause
```

```
UBLmsg ::= SEQUENCE { -- définition des éléments essentiels du message UBL
  ceid          CEID, -- identificateur d'élément de
                   connexion
  osaid         OSAID } -- identificateur d'association de
                   signalisation d'origine
```

```
UBCmsg ::= SEQUENCE { -- définition des éléments essentiels du message UBC
  cause          CAU OPTIONAL } -- cause
```

```
CNFmsg ::= SEQUENCE { -- définition des éléments essentiels du message CNF
  cause          CAU } -- cause
```



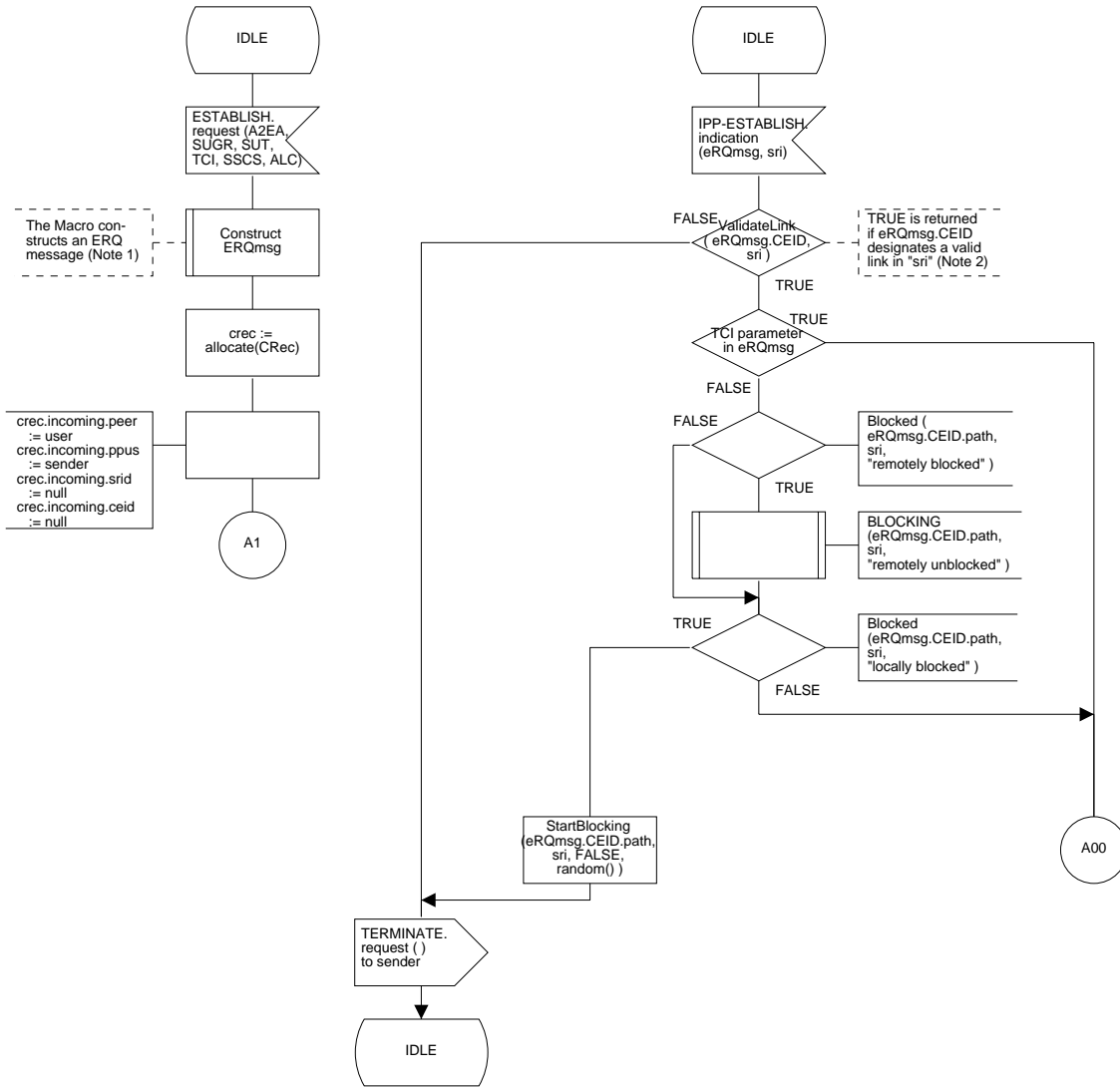
```

dcl
eRQmsg ERQmsg, /* Establish request */
eCFmsg ECFmsg, /* Establish confirm */
rELmsg RELmsg, /* Release request */
rLCmsg RLCmsg, /* Release confirm */
rESmsg RESmsg, /* Reset request */
rSCmsg RSCmsg, /* Reset confirm */
bLOmsg BLOmsg, /* Block request */
bLCmsg BLCmsg, /* Block confirm */
uBLmsg UBLmsg, /* Unblock request */
uBCmsg UBCmsg, /* Unblock confirm */
cNFmsg CNFmsg: /* Confusion */
  
```

NOTE

At start-up, it is assumed that the STI (Signalling Transport Interfaces) to each existing STC (Signalling Transport Converter, there exists one per nodal signalling relation) are created. The addition or removal of nodal signalling relations together with the creation or destruction of the STI and STC processes is not shown in the SDL diagrams of this Annex.

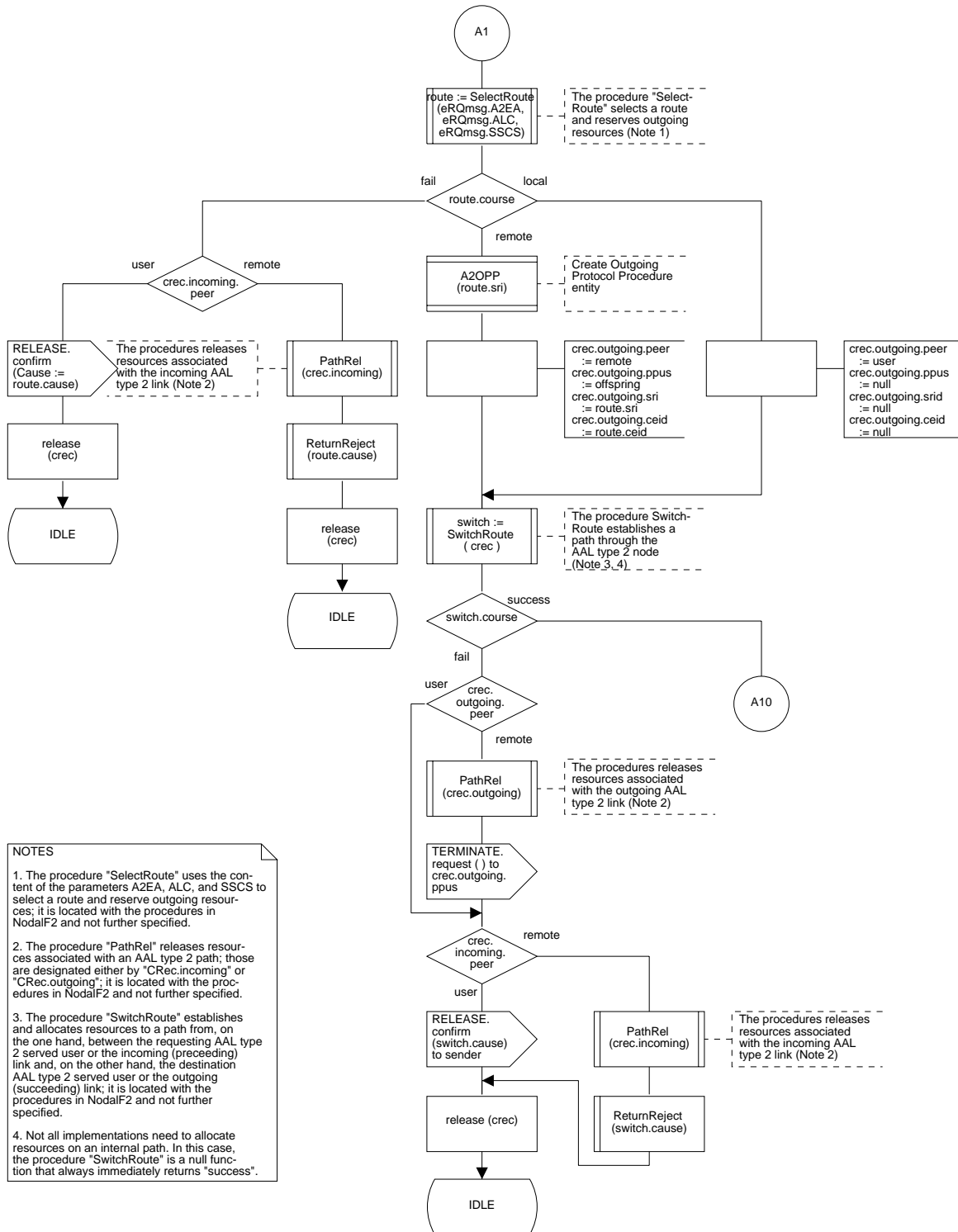
Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 1 de 25)



NOTES

1. The ERQ message includes the required parameters but without compatibility information filled in.
2. The procedure "ValidateLink" determines whether the CEID information in the ERQmsg designates a valid link in the nodal signalling association "sri"; it is located with the procedures in NodalF2 and not further specified.
3. The procedures and functions "Blocked" and "BLOCKING", are located in NodalF2. "StartBlocking" is located in NodalF1.

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 2 de 25)



NOTES

1. The procedure "SelectRoute" uses the content of the parameters A2EA, ALC, and SSCS to select a route and reserve outgoing resources; it is located with the procedures in NodalF2 and not further specified.
2. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified.
3. The procedure "SwitchRoute" establishes and allocates resources to a path from, on the one hand, between the requesting AAL type 2 served user or the incoming (preceding) link and, on the other hand, the destination AAL type 2 served user or the outgoing (succeeding) link; it is located with the procedures in NodalF2 and not further specified.
4. Not all implementations need to allocate resources on an internal path. In this case, the procedure "SwitchRoute" is a null function that always immediately returns "success".

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 4 de 25)

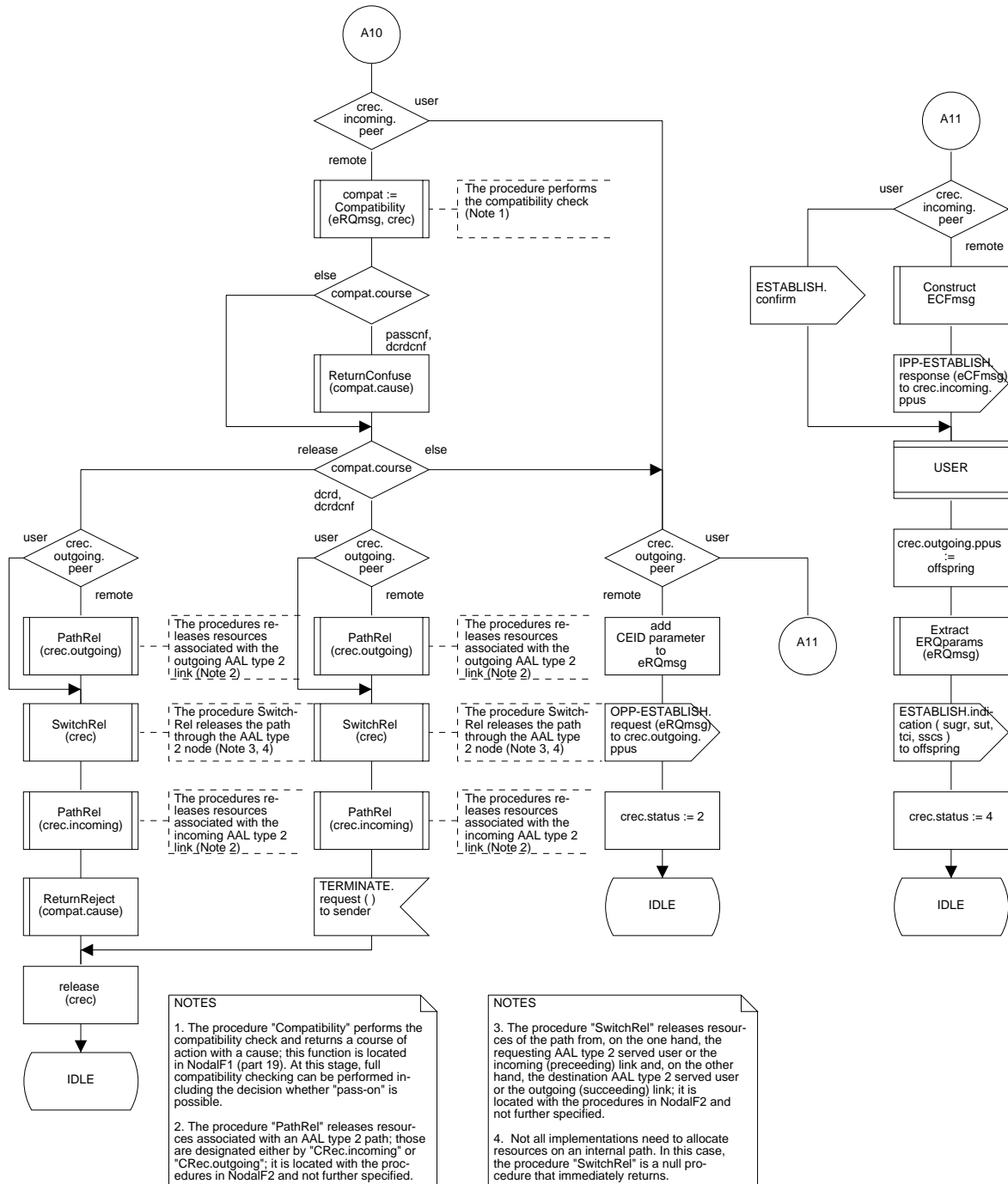


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 5 de 25)

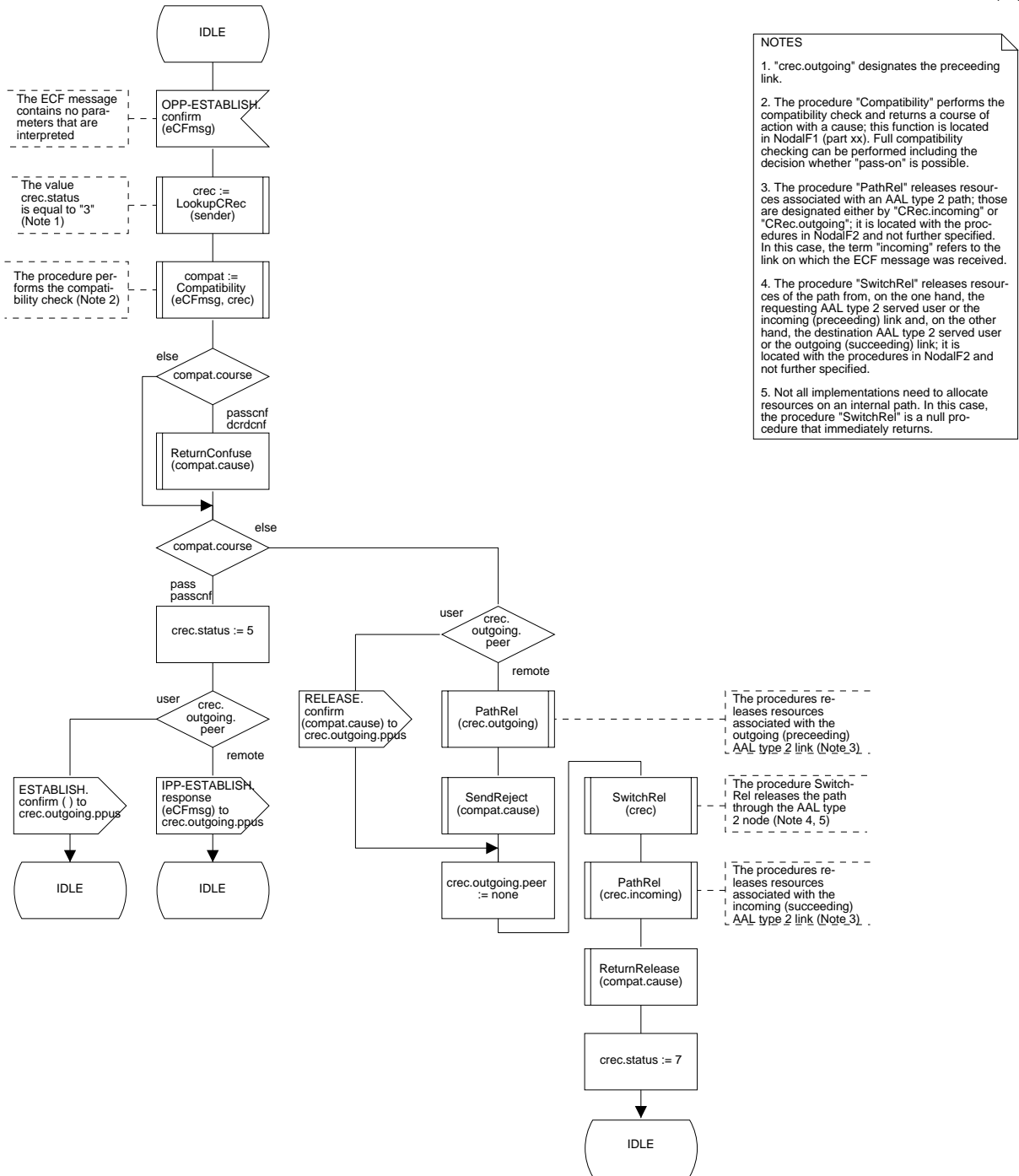
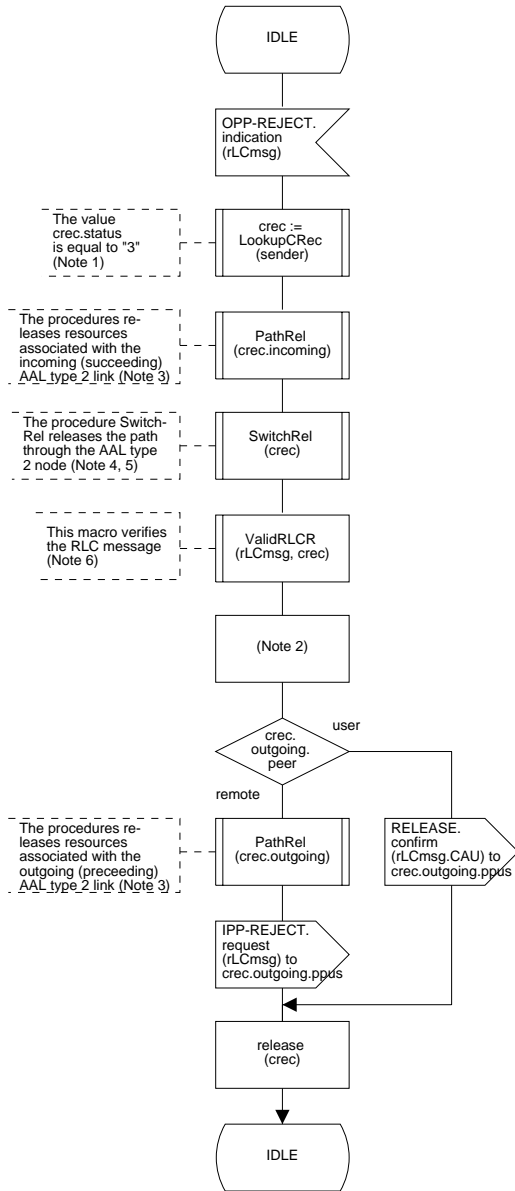
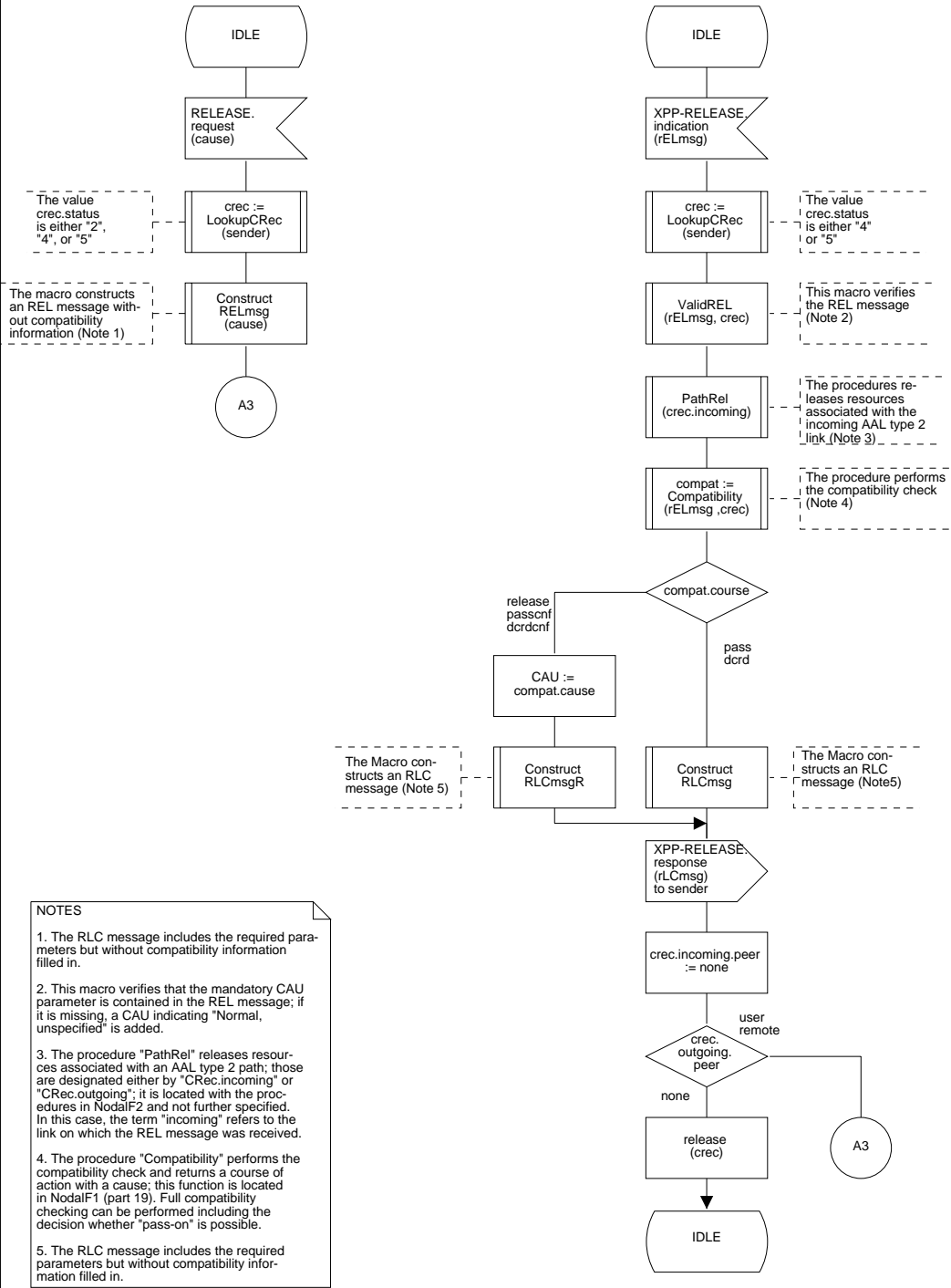


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 6 de 25)



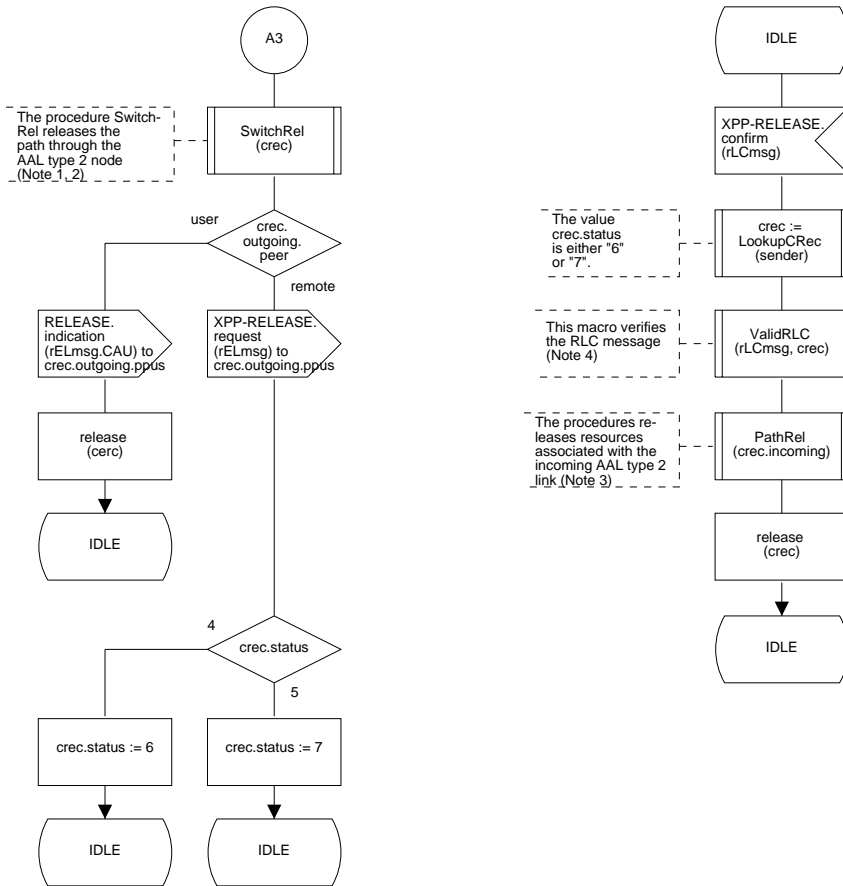
- NOTES**
1. "crec.outgoing" designates the preceding link.
 2. Features that enable a further connection attempt, involving the selection of a different AAL type 2 path within the same route or of an alternative route, may be implemented; however, such features are not specified.
 3. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified. In this case, the term "incoming" refers to the link on which the ECF message was received.
 4. The procedure "SwitchRel" releases resources of the path from, on the one hand, the requesting AAL type 2 served user or the incoming (preceding) link and, on the other hand, the destination AAL type 2 served user or the outgoing (succeeding) link; it is located with the procedures in NodalF2 and not further specified.
 5. Not all implementations need to allocate resources on an internal path. In this case, the procedure "SwitchRel" is a null procedure that immediately returns.
 6. This macro verifies that the mandatory CAU parameter is contained in the RLC message; if it is missing, a CAU indicating "Temporary failure" is added.

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 7 de 25)



- NOTES**
1. The RLC message includes the required parameters but without compatibility information filled in.
 2. This macro verifies that the mandatory CAU parameter is contained in the REL message; if it is missing, a CAU indicating "Normal, unspecified" is added.
 3. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified. In this case, the term "incoming" refers to the link on which the REL message was received.
 4. The procedure "Compatibility" performs the compatibility check and returns a course of action with a cause; this function is located in NodalF1 (part 19). Full compatibility checking can be performed including the decision whether "pass-on" is possible.
 5. The RLC message includes the required parameters but without compatibility information filled in.

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 8 de 25)



NOTES

1. The procedure "SwitchRel" releases resources of the path from, on the one hand, the requesting AAL type 2 served user or the incoming (preceeding) link and, on the other hand, the destination AAL type 2 served user or the outgoing (succeeding) link; it is located with the procedures in NodalF2 and not further specified.
2. Not all implementations need to allocate resources on an internal path. In this case, the procedure "SwitchRel" is a null procedure that immediately returns.
3. The procedure "PathRel" releases resources associated with an AAL type 2 path; those are designated either by "CRec.incoming" or "CRec.outgoing"; it is located with the procedures in NodalF2 and not further specified. In this case, the term "incoming" refers to the link on which the RLC message was received.
4. This macro notifies layer management if the optional CAU parameter is contained in the RLC message.

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 9 de 25)

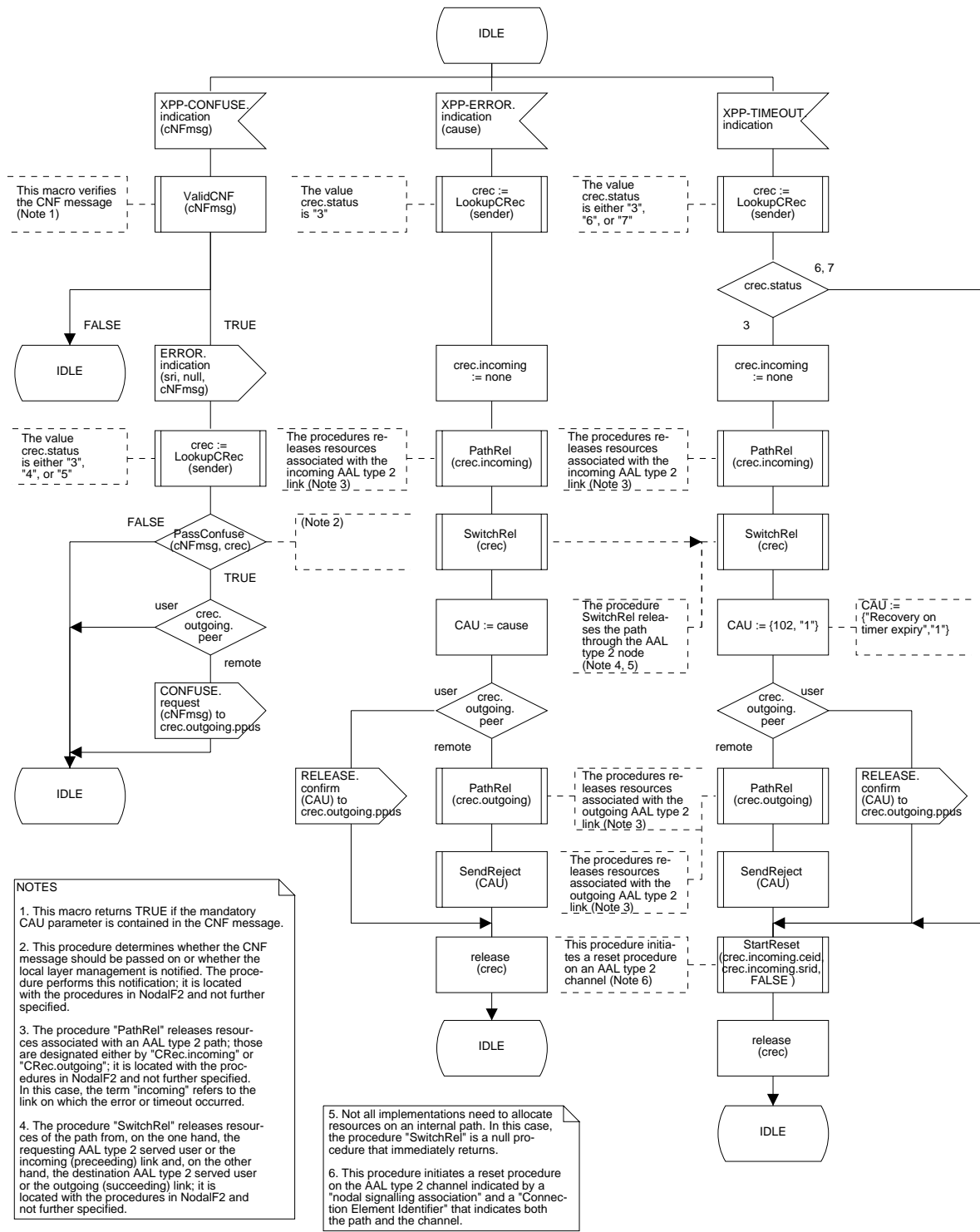


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 10 de 25)

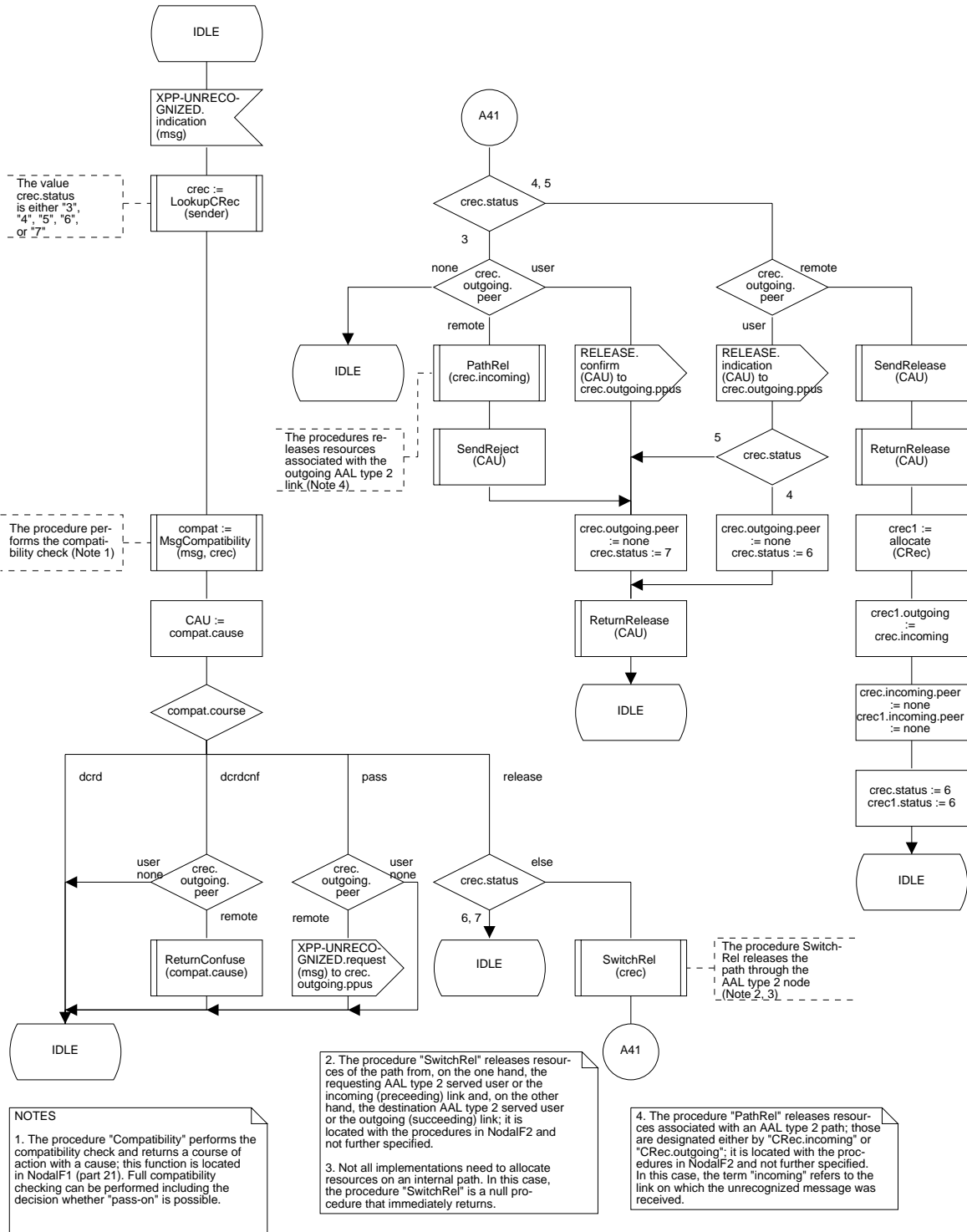
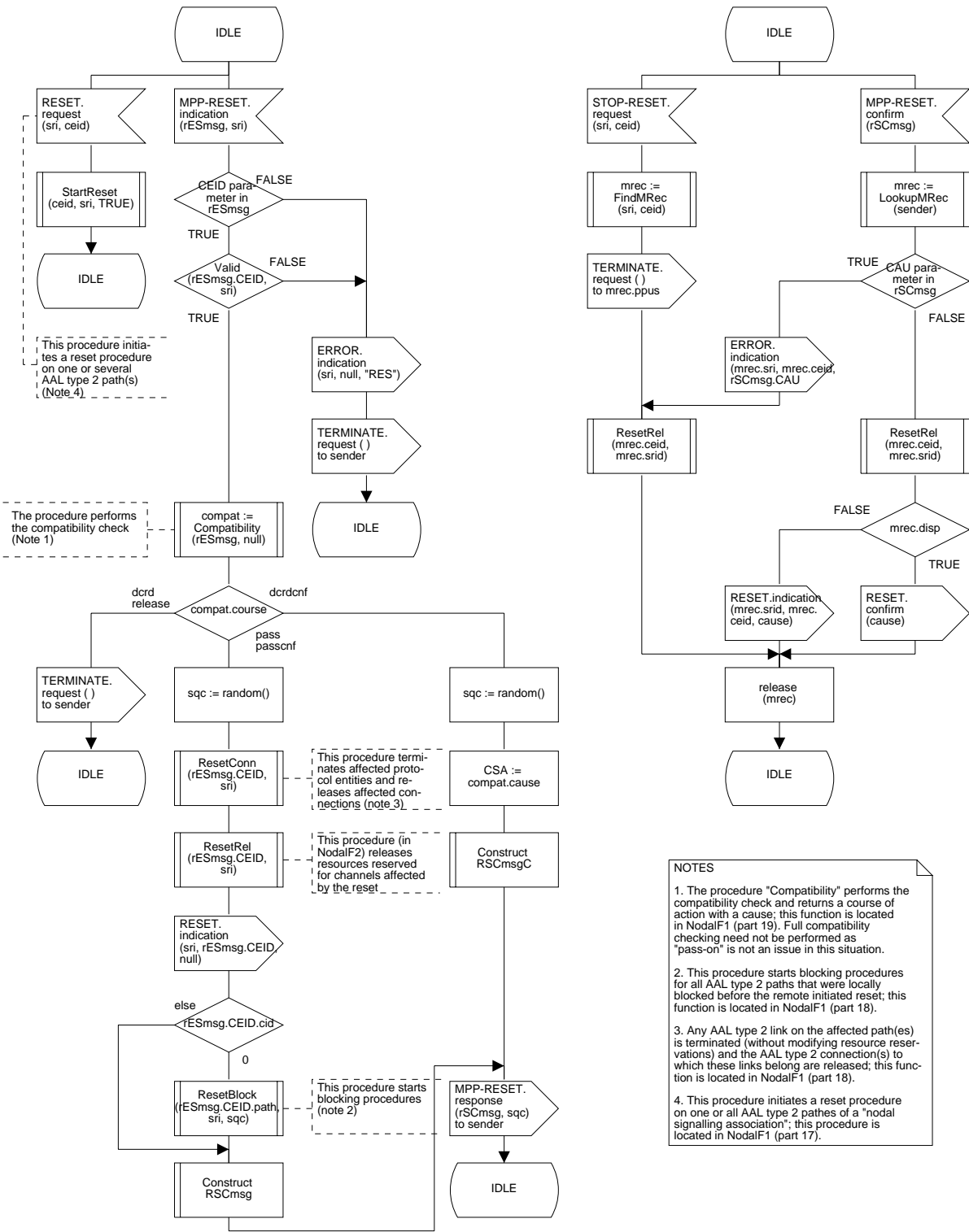
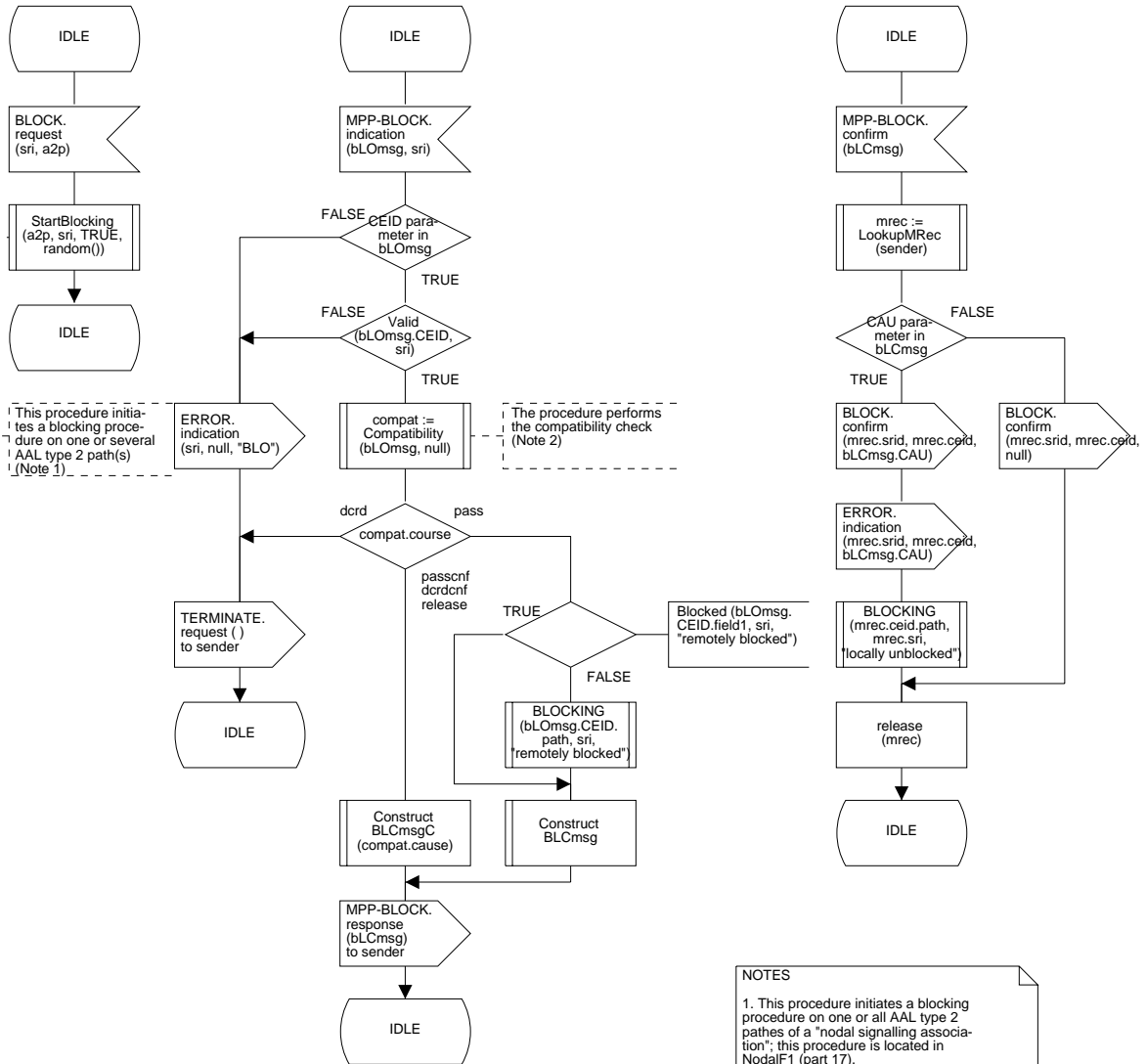


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 11 de 25)



- NOTES
1. The procedure "Compatibility" performs the compatibility check and returns a course of action with a cause; this function is located in NodalF1 (part 19). Full compatibility checking need not be performed as "pass-on" is not an issue in this situation.
 2. This procedure starts blocking procedures for all AAL type 2 paths that were locally blocked before the remote initiated reset; this function is located in NodalF1 (part 18).
 3. Any AAL type 2 link on the affected path(es) is terminated (without modifying resource reservations) and the AAL type 2 connection(s) to which these links belong are released; this function is located in NodalF1 (part 18).
 4. This procedure initiates a reset procedure on one or all AAL type 2 paths of a "nodal signalling association"; this procedure is located in NodalF1 (part 17).

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 12 de 25)



NOTES

1. This procedure initiates a blocking procedure on one or all AAL type 2 paths of a "nodal signalling association"; this procedure is located in NodalF1 (part 17).
2. The procedure "Compatibility" performs the compatibility check and returns a course of action with a cause; this function is located in NodalF1 (part 19). Full compatibility checking need not be performed as "pass-on" is not an issue in this situation.

Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 13 de 25)

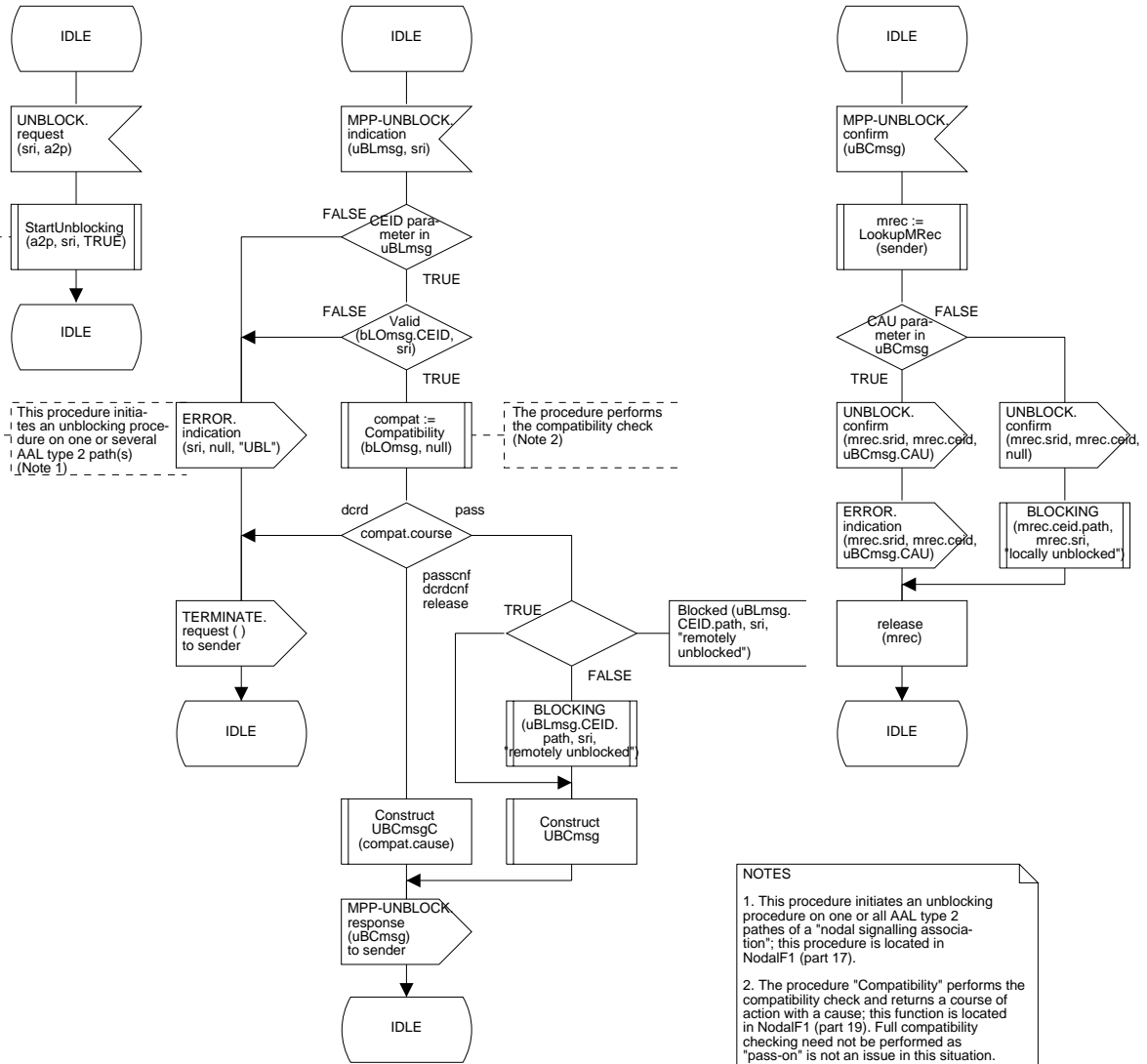


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 14 de 25)

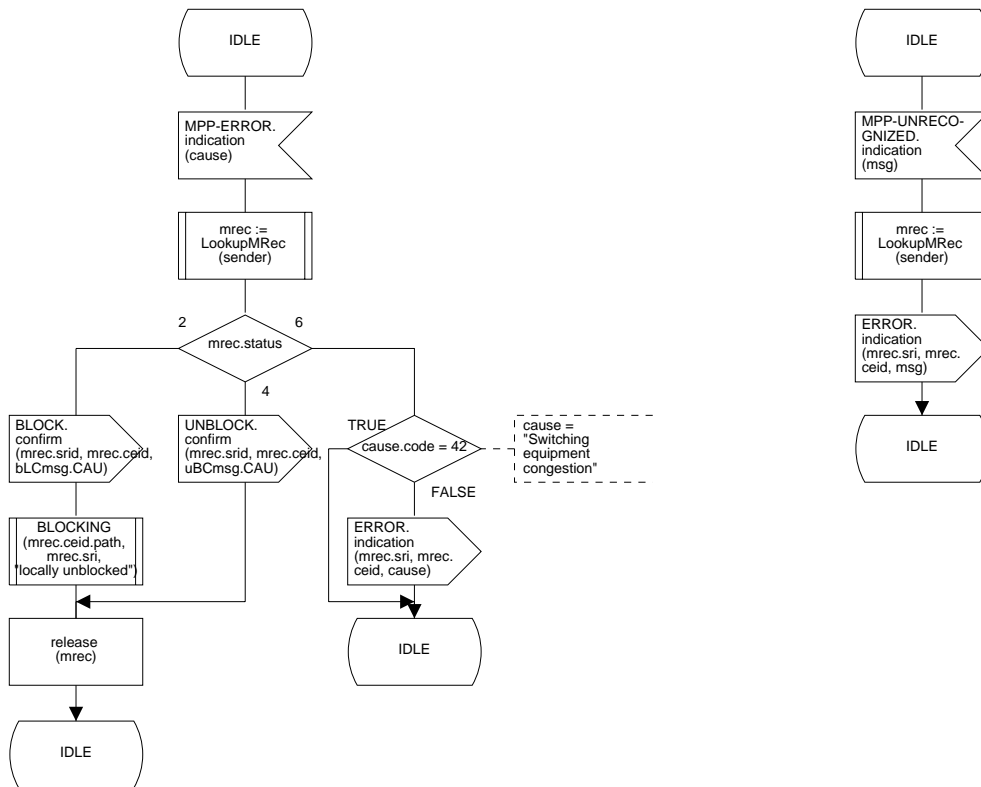
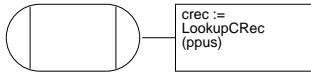


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (feuille 15 de 25)



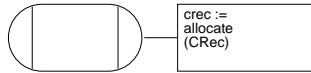
This function searches all records of type "CRec" to find the one that matches either the crec.incoming.ppus or the crec.outgoing.ppus with the input parameter. Exactly one such record is found.

If the input parameter matches the crec.outgoing.ppus, the incoming and outgoing parts of the record are exchanged. If such an exchange took place, the status part of the record is also modified as follows:

```

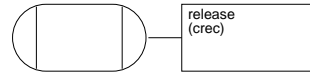
if even(crec.status) then
  increment crec.status by 1
else
  decrement crec.status by 1
endif
    
```

The value returned can be understood to be a pointer to the record itself.

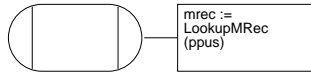


This function allocates a record of type "CRec" or "MRec".

The value returned can be understood to be a pointer to the record itself.

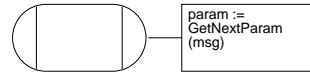


This function deallocates a record of type "CRec" or "MRec" referenced by the parameter "crec" or "mrec". The record becomes unavailable.



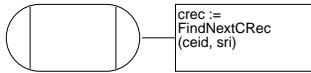
This function searches all records of type "MRec" to find the one that matches the mrec.ppus parameter. Exactly one such record is found.

The value returned can be understood to be a pointer to the record itself.



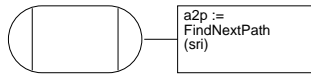
This function parses the message and isolates the next parameter.

The value returned is (a reference to) the parameter, unless no further parameters are found (in which case the value "null" is returned).



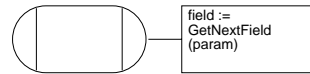
This function searches all records of type "MRec" to find the next one that matches the "ceid" and "sri" parameters.

The value returned can be understood to be a pointer to the record itself, unless no further records are found (in which case the value "null" is returned).



This function searches for all assigned paths of the signalling relationship indicated with the "sri" parameter.

The value returned is the value of an AAL type 2 path identifier, unless no further paths are found (in which case the value "null" is returned).



This function parses the parameter and isolates the next field

The value returned is (a reference to) the field, unless no further fields exist (in which case the value "null" is returned).



Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (procédures) (feuille 16 de 25)

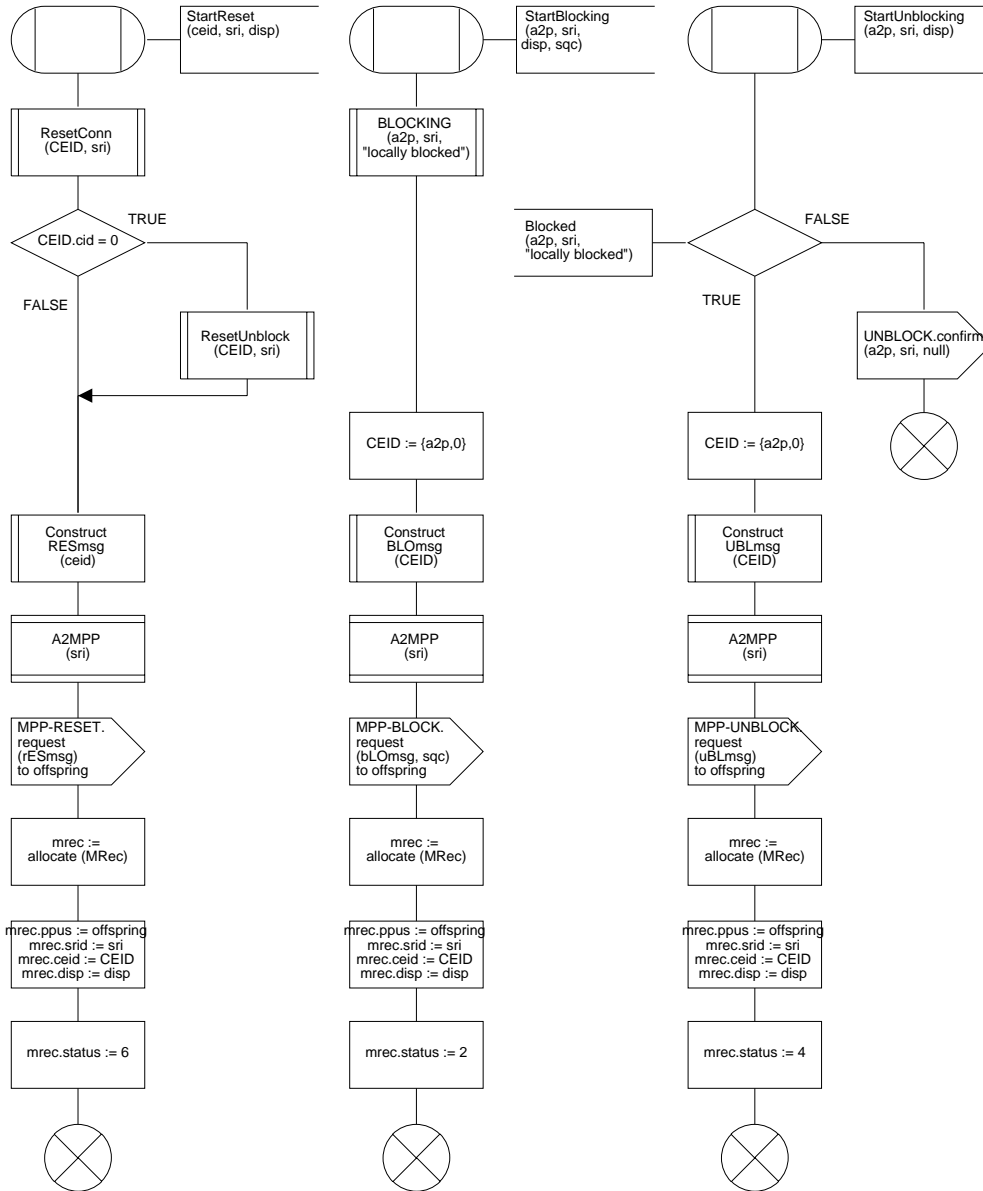


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (procédures) (feuille 17 de 25)

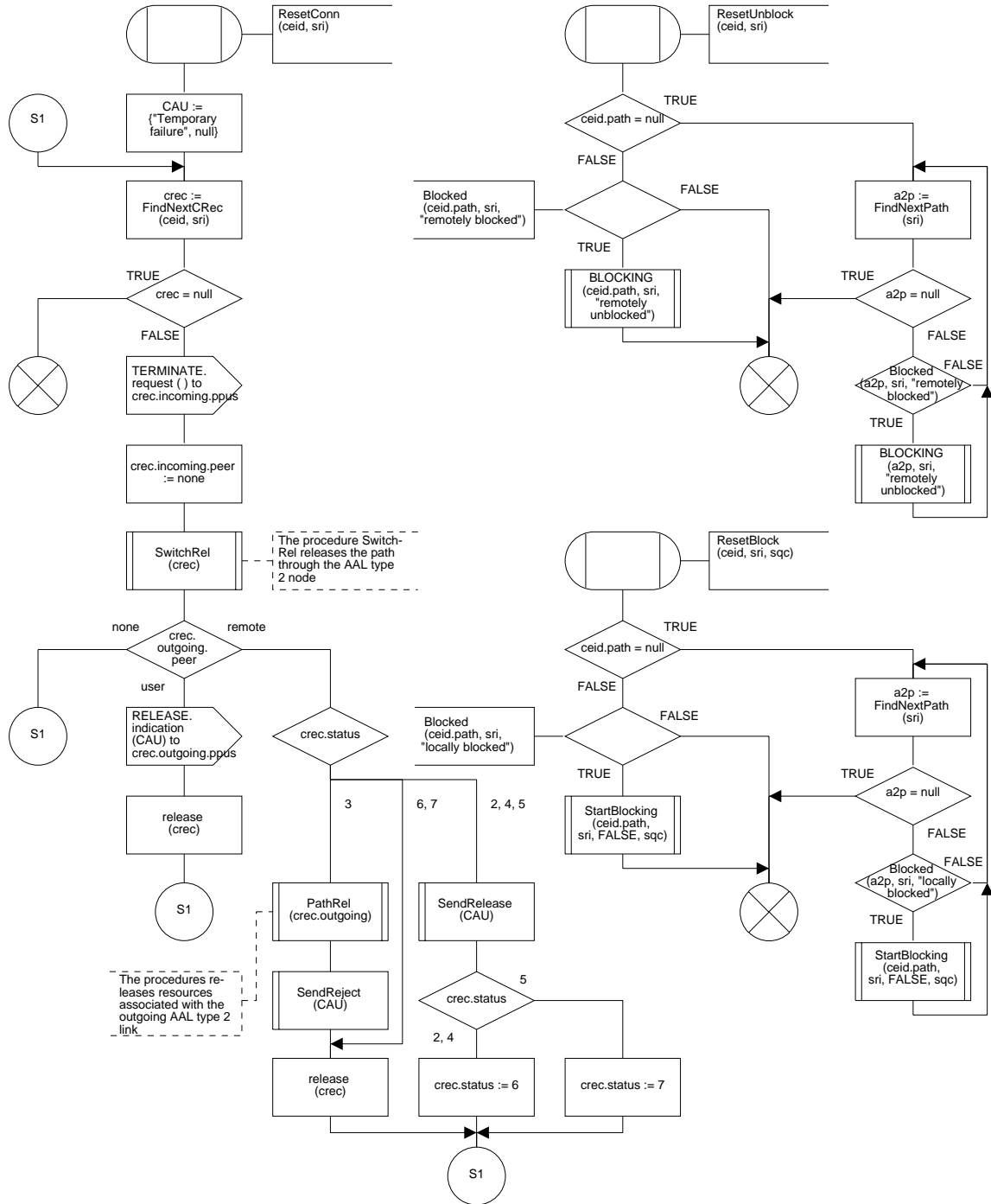


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (procédures) (feuille 18 de 25)

This function checks the compatibility of parameters in recognized messages. It returns a structured value with a course and a cause.

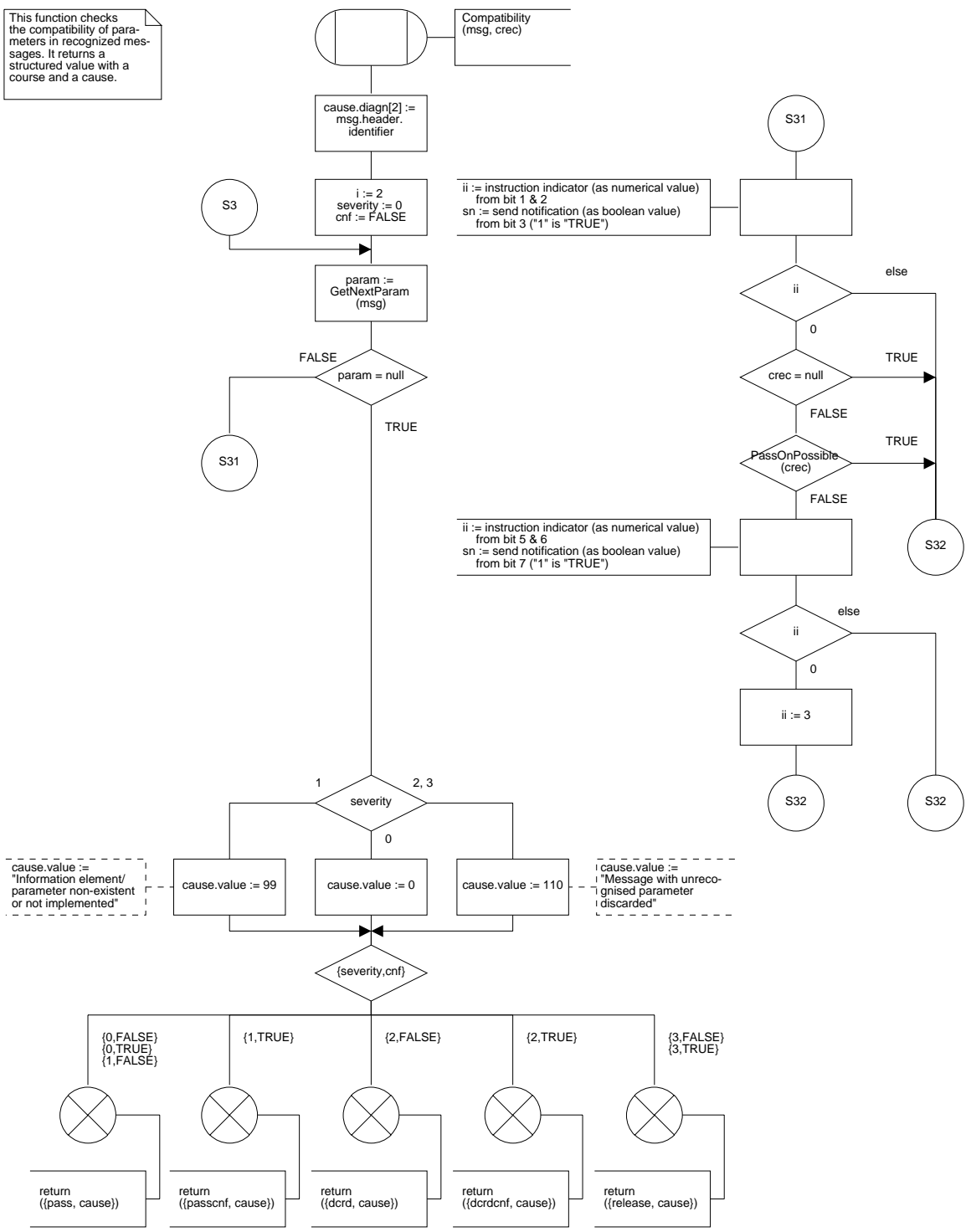


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (procédures) (feuille 19 de 25)

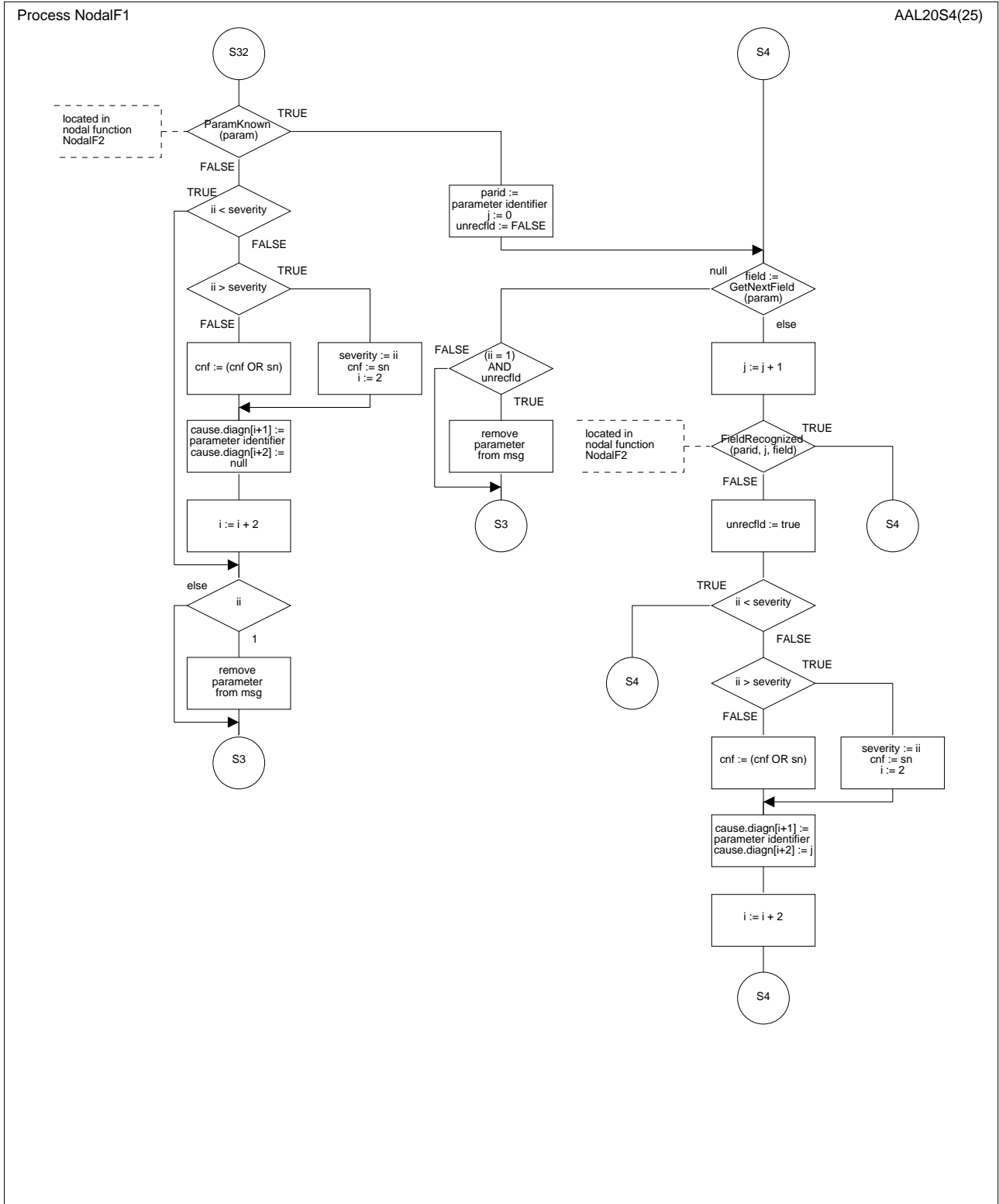


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (procédures) (feuille 20 de 25)

This function checks the compatibility of unrecognized messages. It returns a structured value with a cause and a cause.

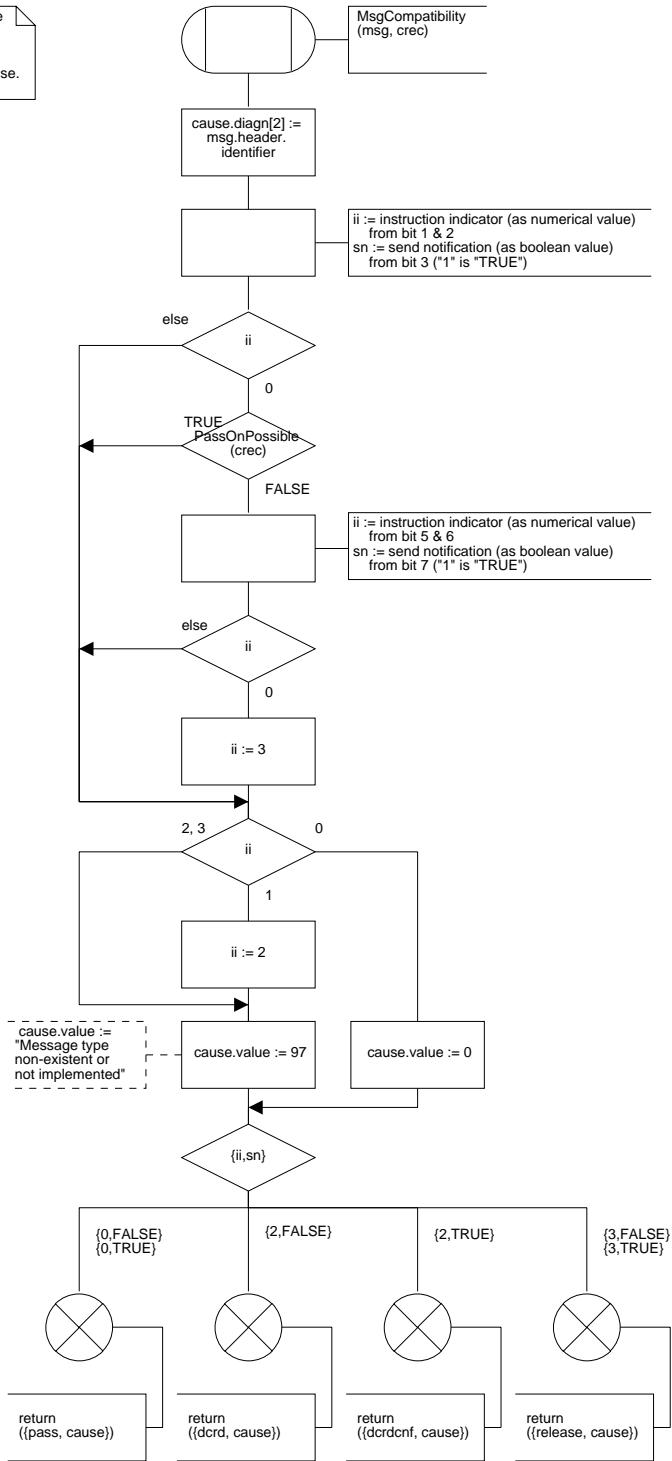


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (procédures) (feuille 21 de 25)

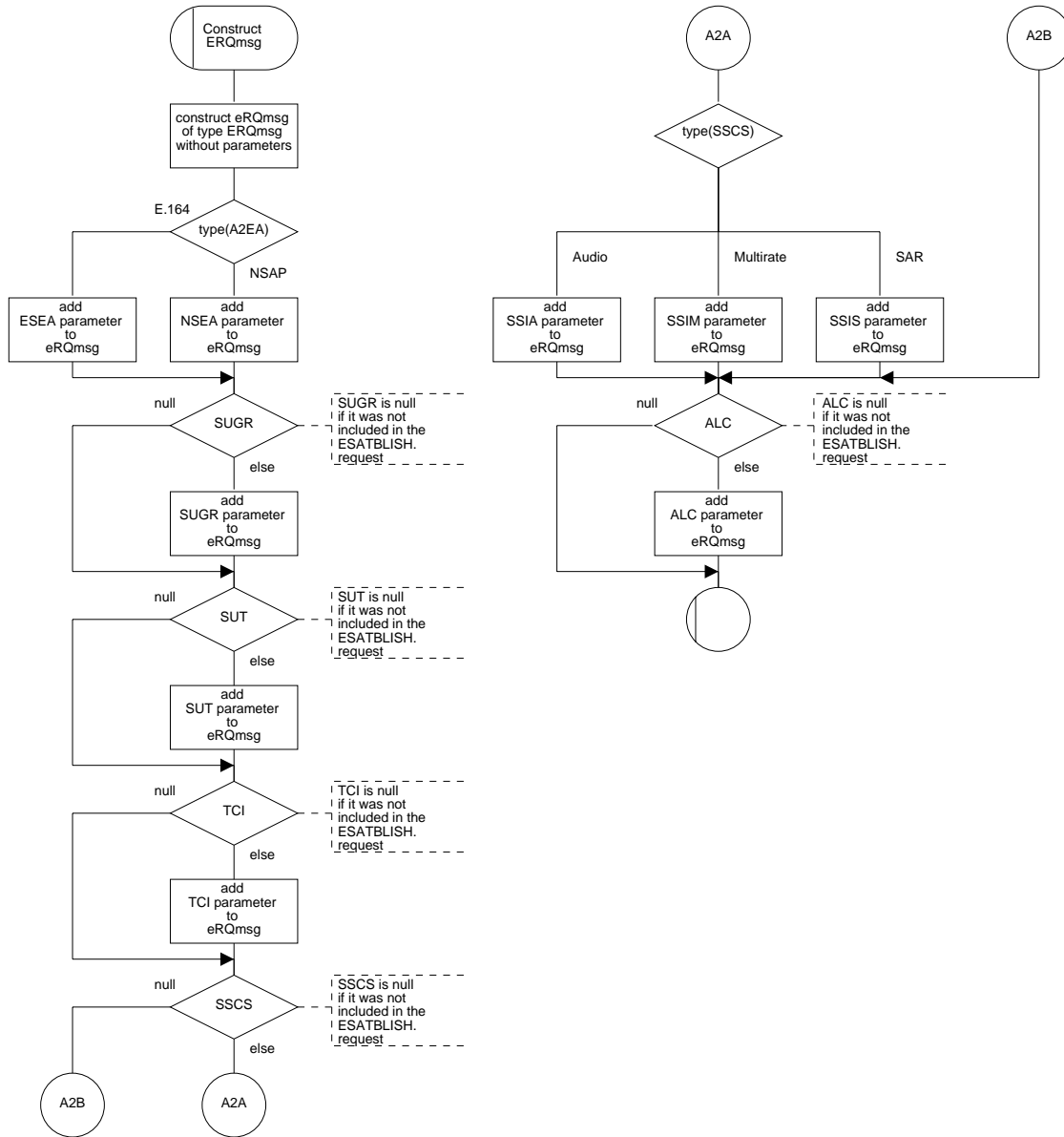


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (macros) (feuille 22 de 25)

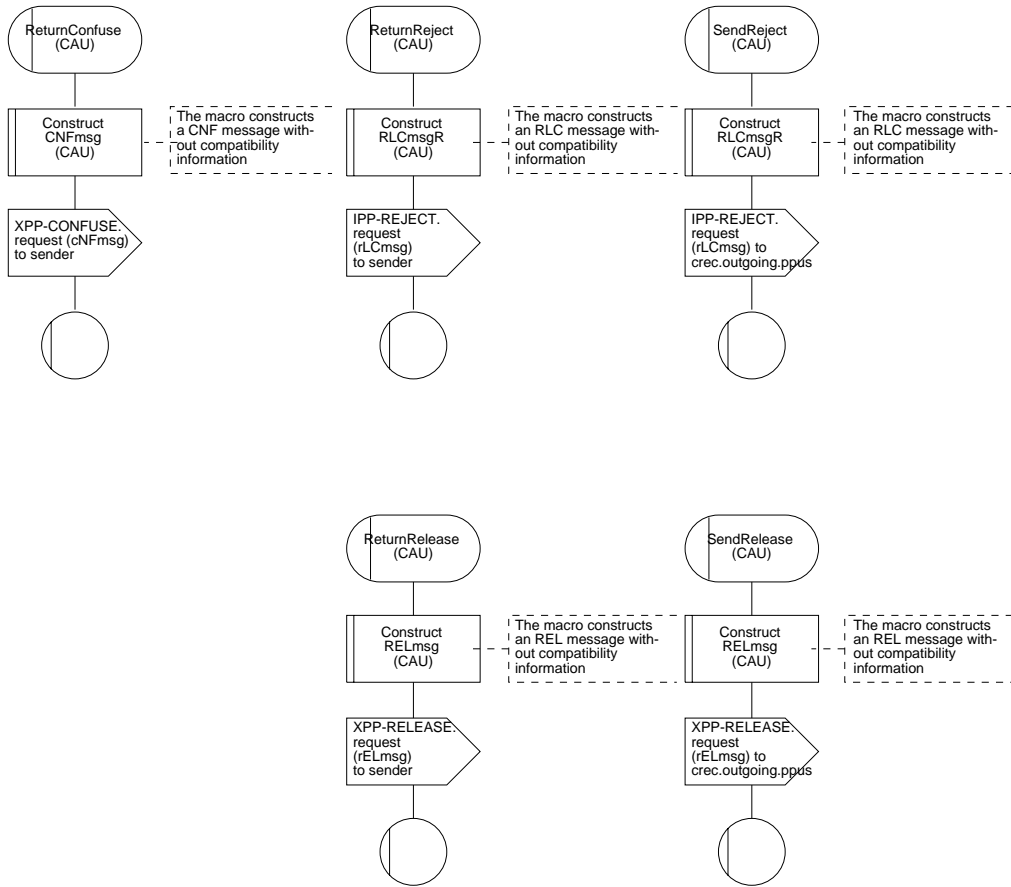


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (macros) (feuille 23 de 25)

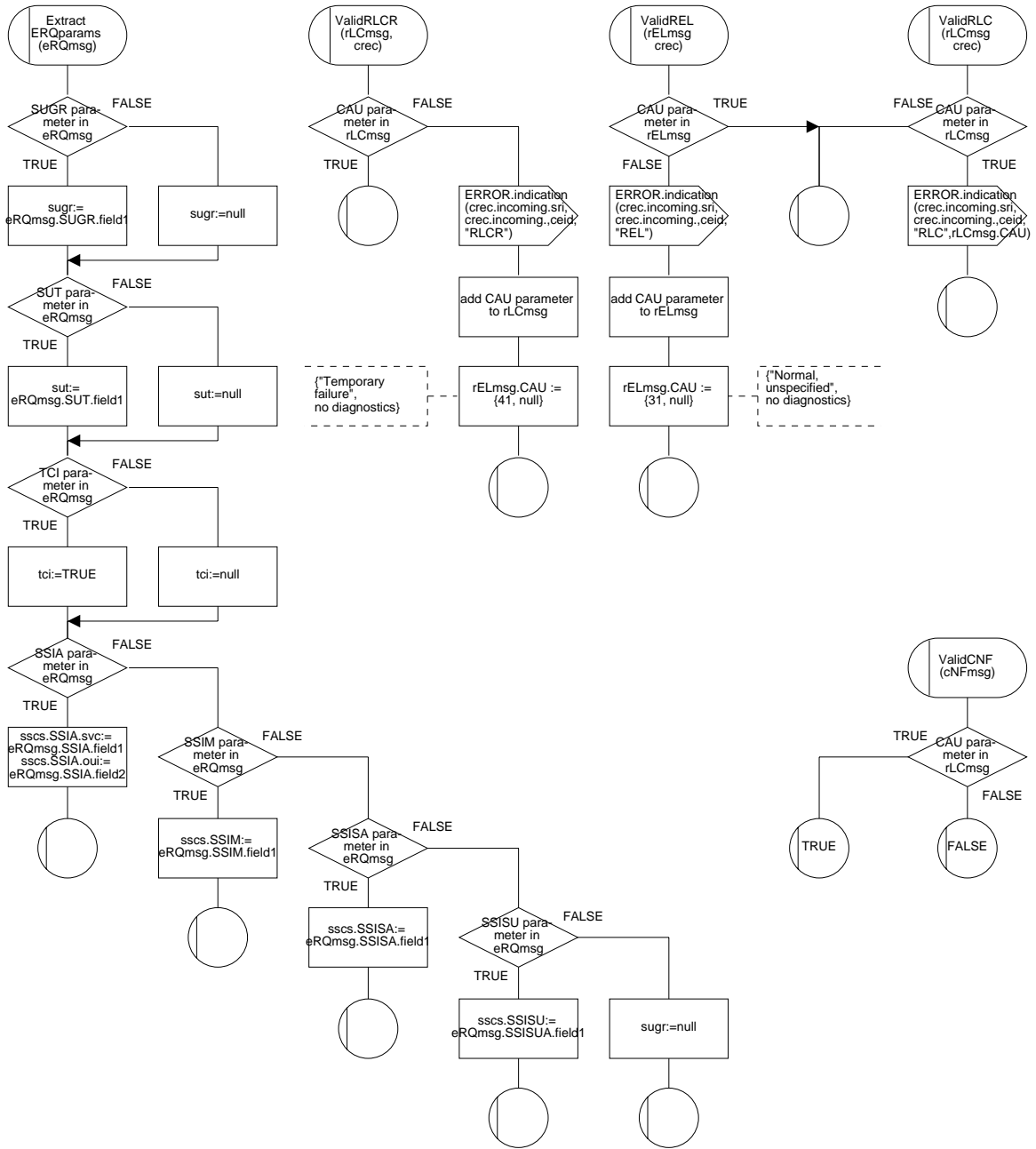


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (macros) (feuille 24 de 25)

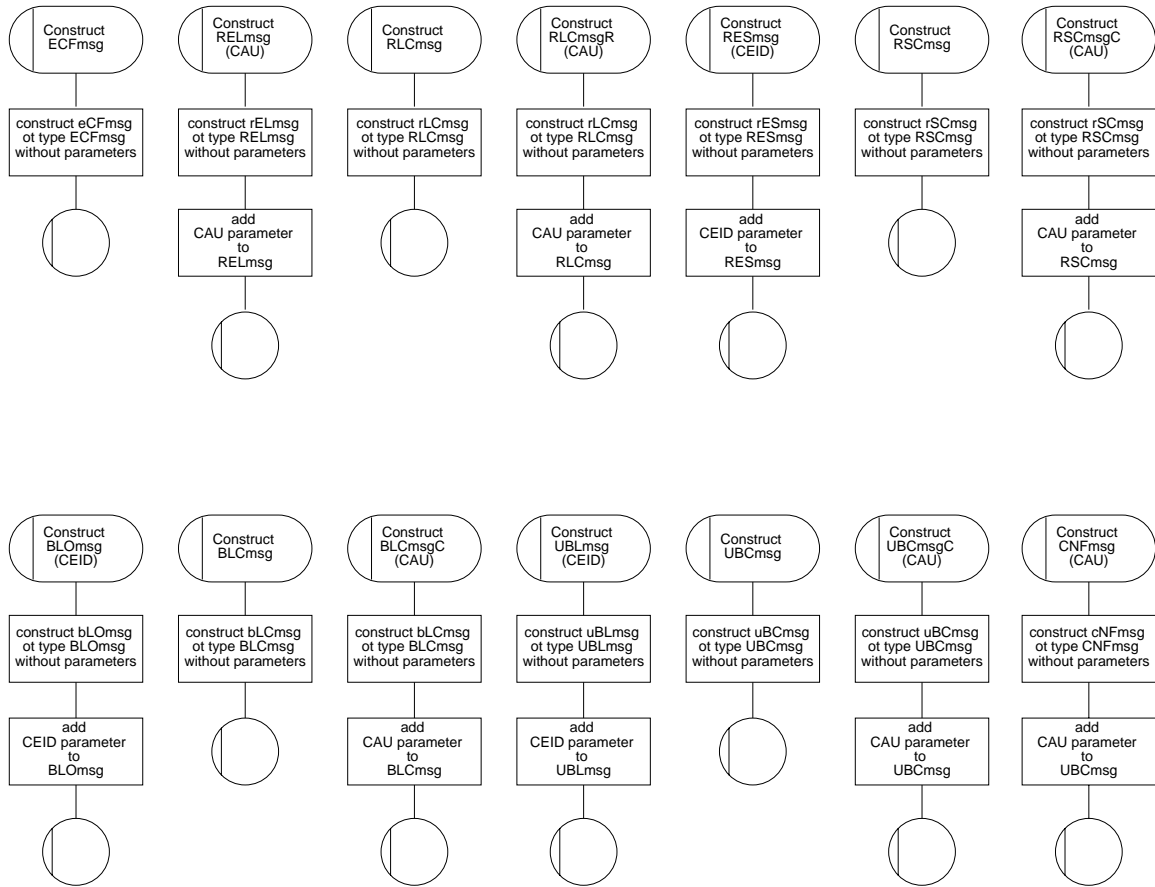


Figure B.3/Q.2630.1 – Diagramme SDL de la fonction nodale 1 (macros) (feuille 25 de 25)

B.5 Diagrammes SDL des entités de protocole

B.5.1 Introduction

On voit sur la Figure B.2 (feuillet 3 de 4) que l'entité de protocole est subdivisée en quatre éléments:

a) *les procédures protocolaires sortantes*

Cette entité contient toutes les fonctionnalités définies de manière détaillée au 8.3.2 du texte principal de la présente Recommandation. Elle peut donc être spécifiée de manière précise dans les diagrammes SDL.

b) *les procédures protocolaires entrantes*

Cette entité contient toutes les fonctionnalités définies de manière détaillée au 8.3.3 du texte principal de la présente Recommandation. Elle peut donc être spécifiée de manière précise dans les diagrammes SDL.

c) *les procédures protocolaires de maintenance*

Cette entité contient toutes les fonctionnalités définies de manière détaillée au 8.3.4 du texte principal de la présente Recommandation. Elle peut donc être spécifiée de manière précise dans les diagrammes SDL.

d) *l'interface de transport de signalisation*

Cette entité n'est pas expressément spécifiée dans la présente Recommandation. Il lui appartient toutefois de recevoir les primitives et les messages en provenance de l'interface de transport de signalisation, de les acheminer jusqu'à l'instance d'entité de protocole appropriée et de transmettre les signaux "IN-SERVICE", "OUT-OF-SERVICE", et "CONGESTION" à la fonction globale 2. Dans certains cas, ces instances d'entité de protocole doivent être créées par l'interface de transport de signalisation (à la réception d'un message ERQ, RES, BLO ou UBL, par exemple). Cette entité est également spécifiée dans les diagrammes SDL.

NOTE – Dans le sens émission, cette entité ne remplit aucune fonctionnalité.

Le présent paragraphe définit donc l'entité procédures protocolaires sortantes (voir B.5.2.2), l'entité procédures protocolaires entrantes (voir B.5.2.3), l'entité procédures protocolaires de maintenance (voir B.5.2.4) et l'entité interface de transport de signalisation (voir B.5.2.5).

B.5.2 Diagrammes SDL des procédures protocolaires sortantes, entrantes et de maintenance

B.5.2.1 Structure des données

Les diagrammes SDL figurant dans le présent paragraphe utilisent les définitions ASN.1 données au B.4.2.5.

B.5.2.2 Diagrammes SDL des procédures protocolaires sortantes

Les diagrammes SDL de la procédure protocolaire sortante sont décrits dans les feuillets 1 à 5 de la Figure B.4.

B.5.2.3 Diagrammes SDL des procédures protocolaires entrantes

Les diagrammes SDL de la procédure protocolaire entrante sont décrits dans les feuillets 1 à 5 de la Figure B.5.

B.5.2.4 Diagrammes SDL des procédures protocolaires de maintenance

Les diagrammes SDL de la procédure protocolaire de maintenance sont décrits dans les feuillets 1 à 5 de la Figure B.6.

B.5.2.5 Diagrammes SDL de l'interface de transport de signalisation

Les diagrammes SDL de l'interface de transport de signalisation sont décrits dans les feuillets 1 à 2 de la Figure B.7.

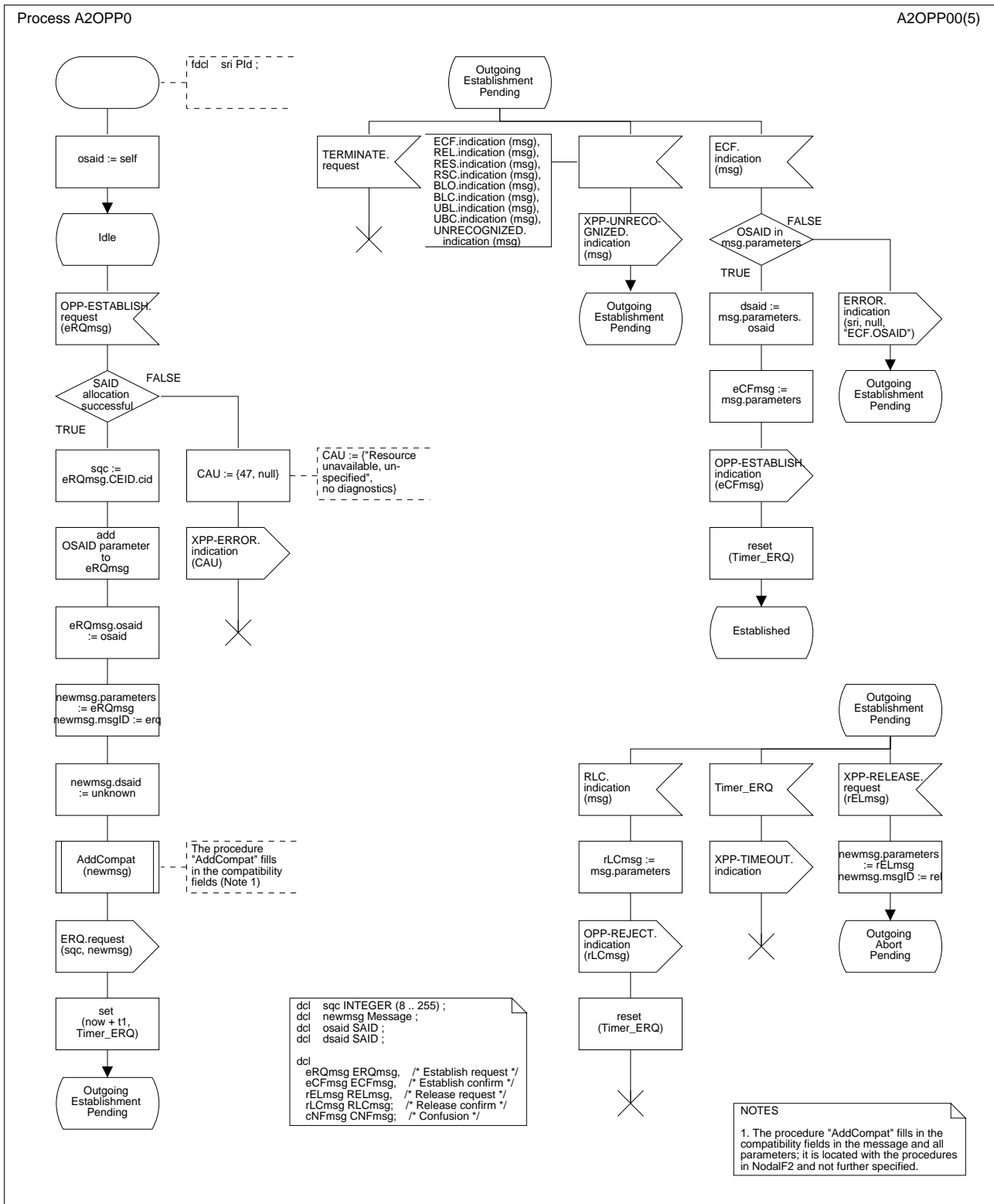
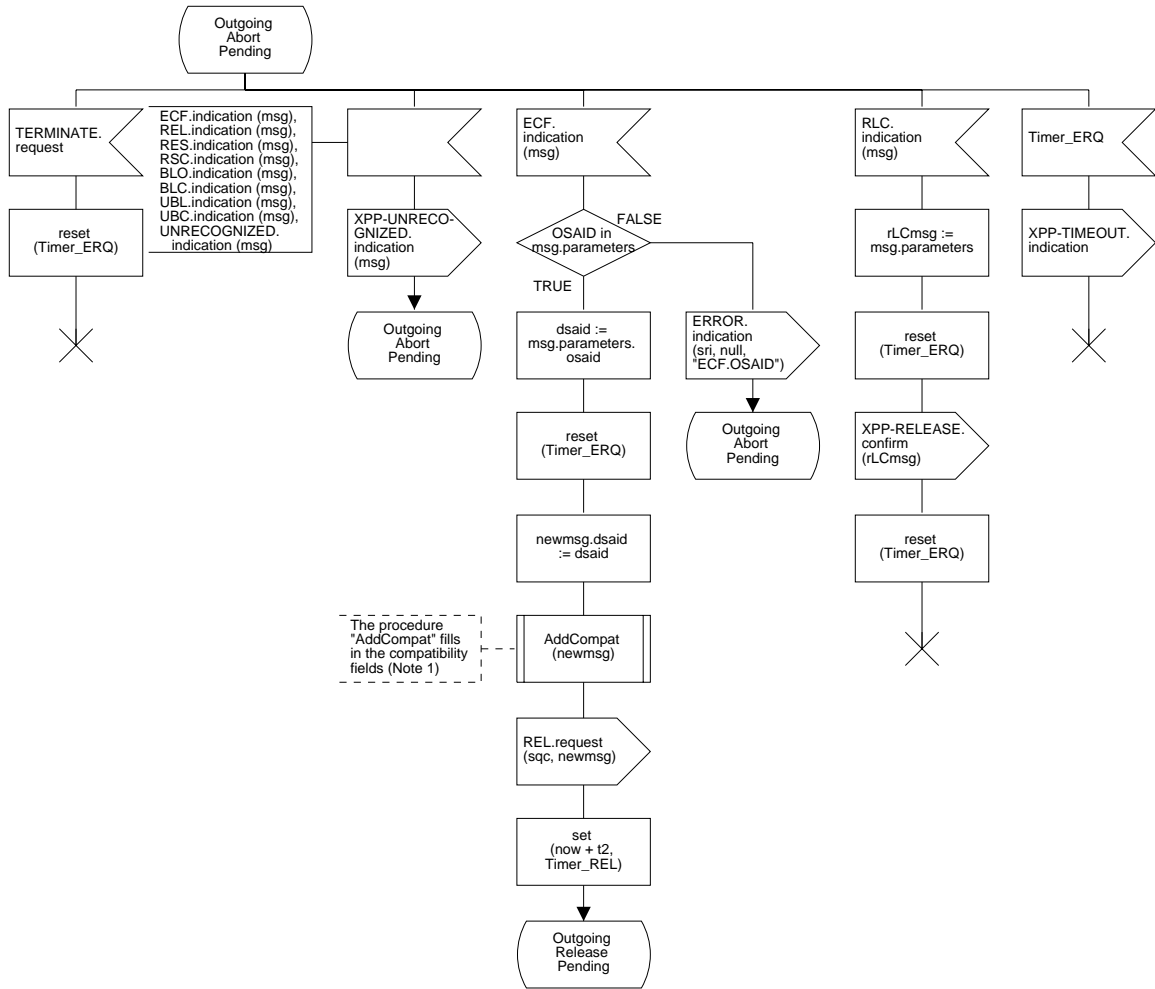
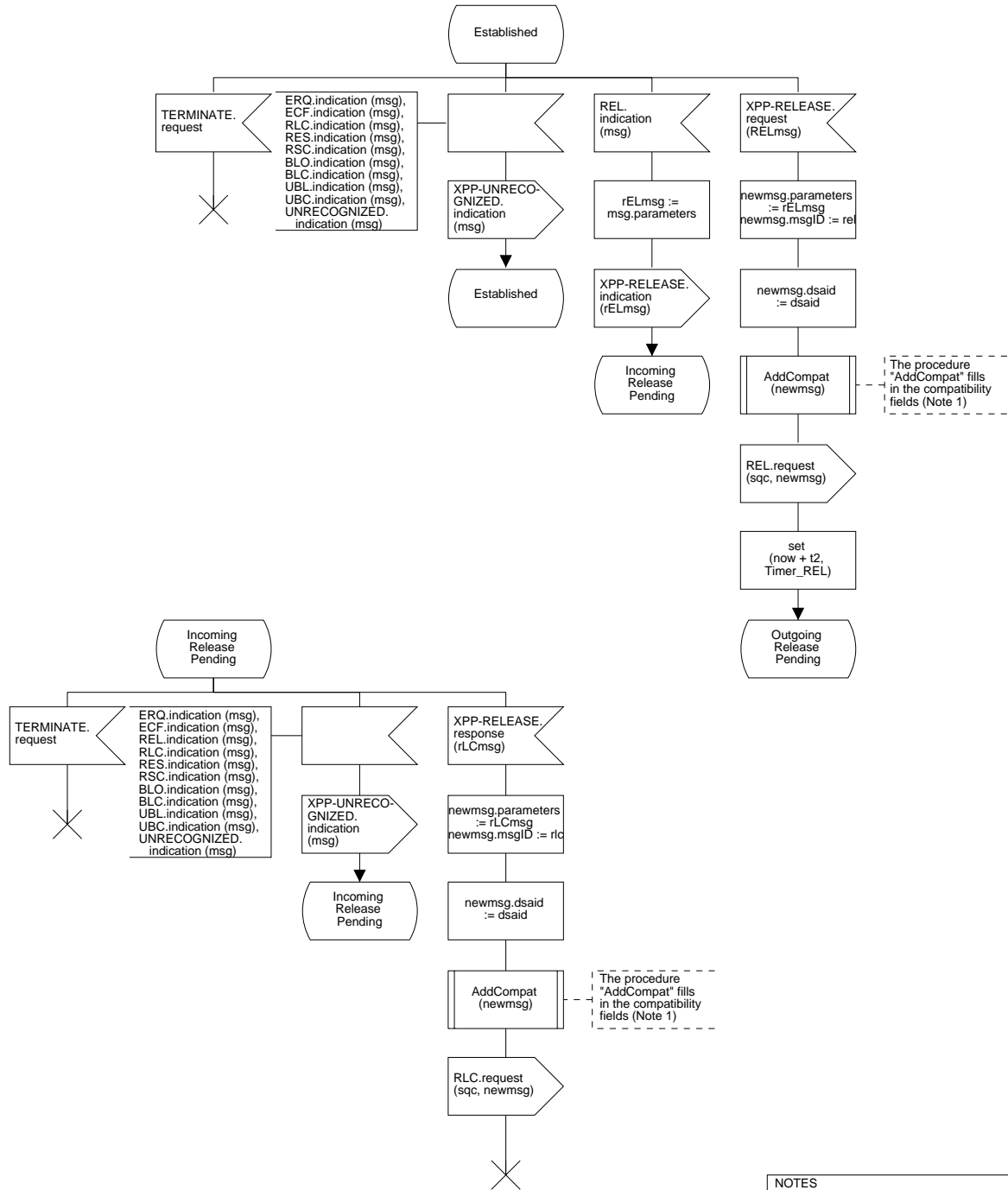


Figure B.4/Q.2630.1 – Diagramme SDL de la procédure protocolaire sortante (feuille 1 de 5)



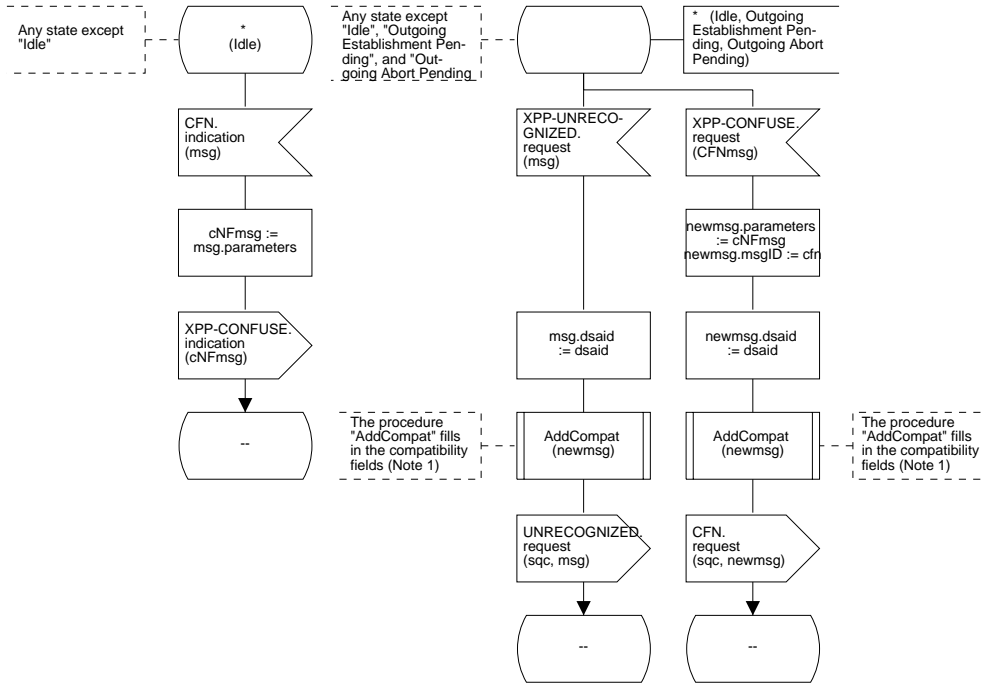
NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.4/Q.2630.1 – Diagramme SDL de la procédure protocolaire sortante (feuille 2 de 5)



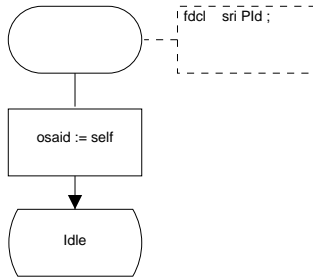
NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.4/Q.2630.1 – Diagramme SDL de la procédure protocolaire sortante (feuille 3 de 5)



NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

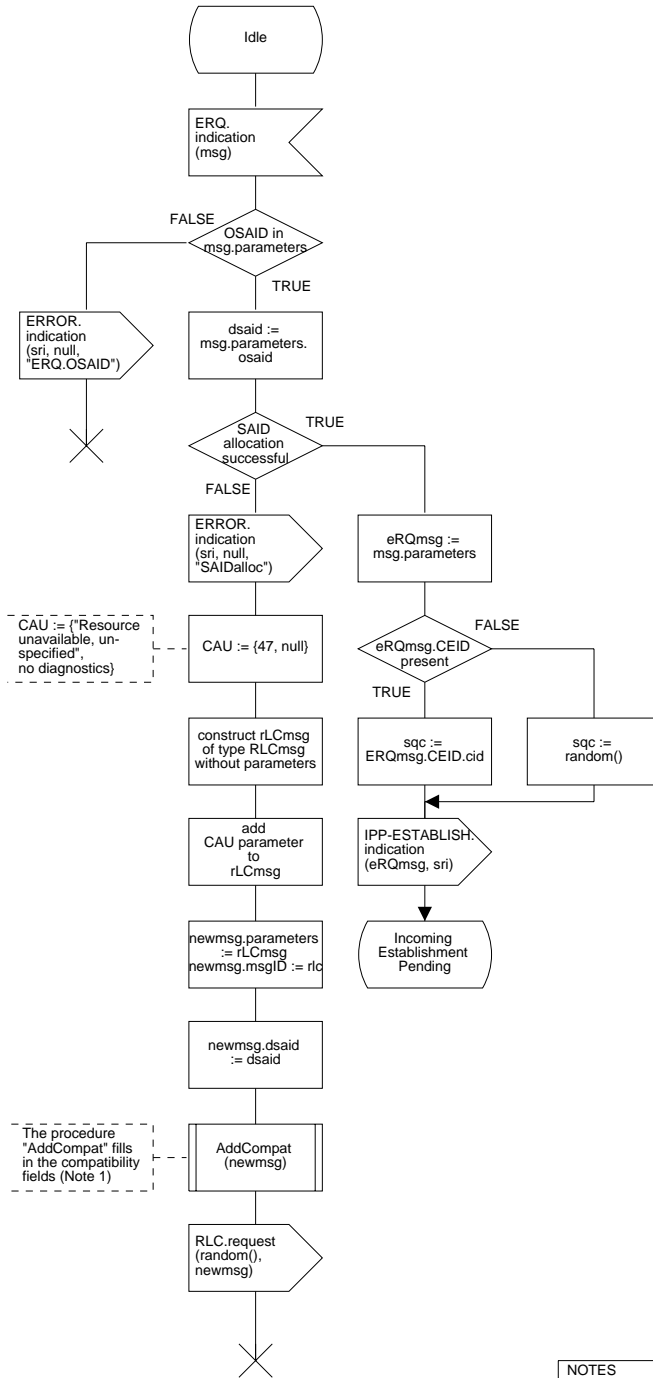
Figure B.4/Q.2630.1 – Diagramme SDL de la procédure protocolaire sortante (feuille 5 de 5)



```

dcl sqc INTEGER (8 .. 255);
dcl newmsg Message;
dcl osaid SAID;
dcl dsaid SAID;

dcl eRQmsg ERQmsg; /* Establish request */
dcl eCFmsg ECFmsg; /* Establish confirm */
dcl rELmsg RELmsg; /* Release request */
dcl rLCmsg RLCmsg; /* Release confirm */
dcl cNFmsg CNFmsg; /* Confusion */
    
```

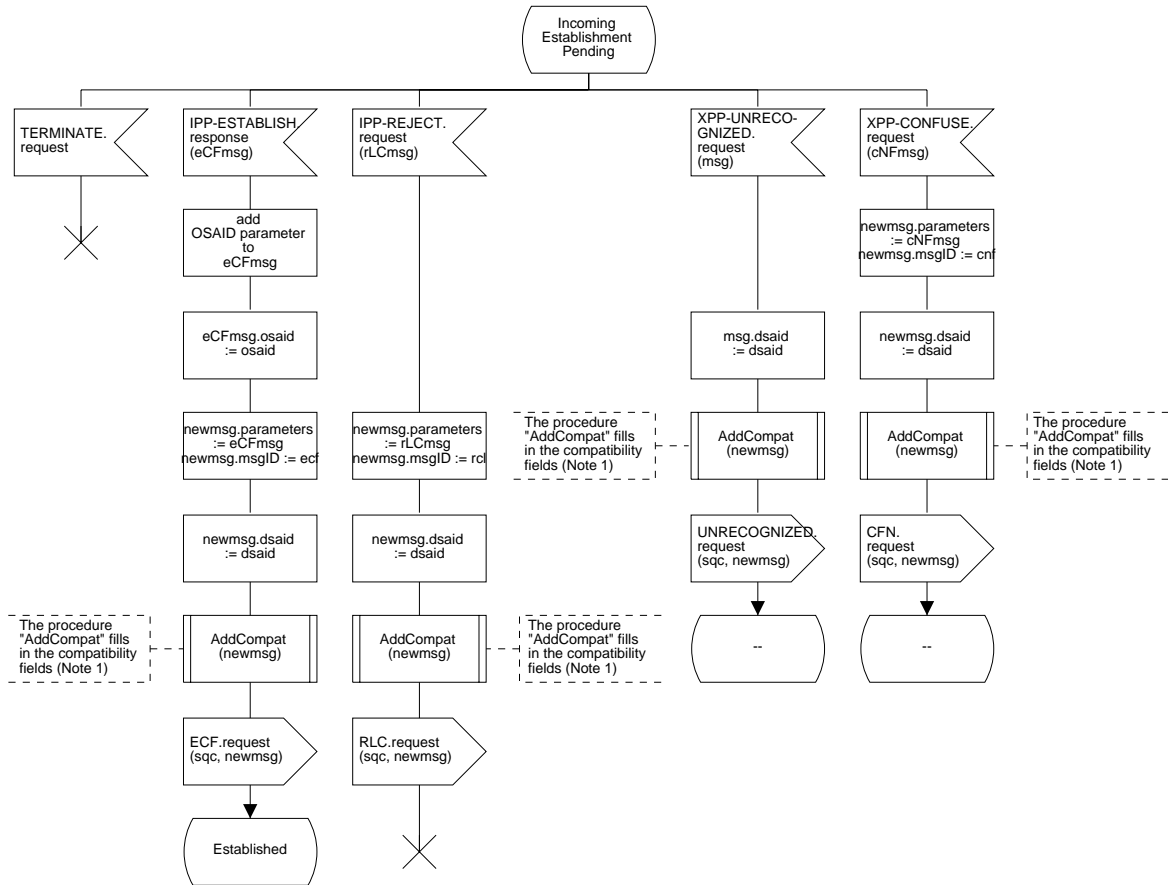


CAU := ("Resource unavailable, unspecified", no diagnostics)

The procedure "AddCompat" fills in the compatibility fields (Note 1)

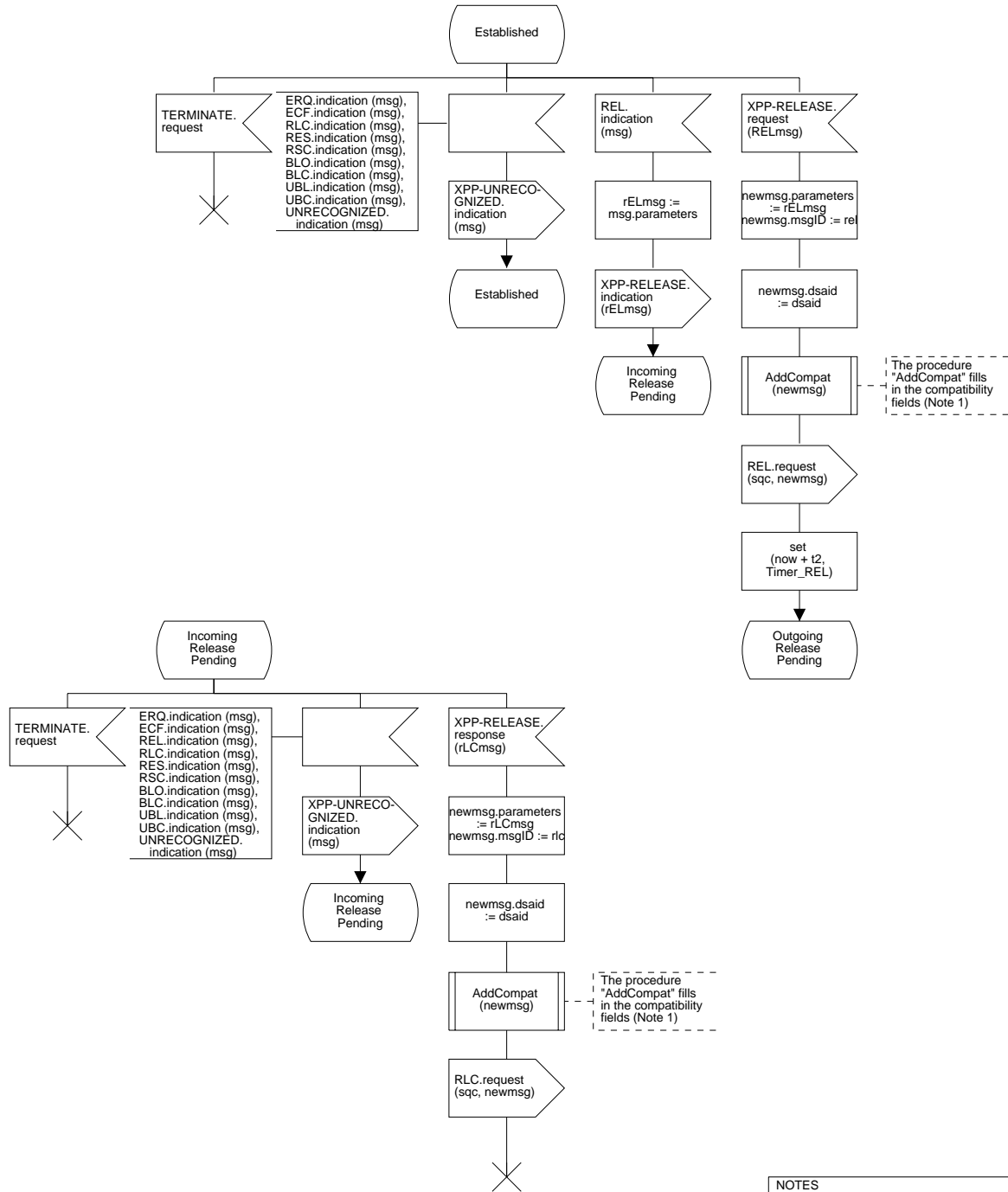
NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.5/Q.2630.1 – Diagramme SDL de la procédure protocolaire entrante (feuille 1 de 5)



NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.5/Q.2630.1 – Diagramme SDL de la procédure protocolaire entrante (feuille 2 de 5)



NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.5/Q.2630.1 – Diagramme SDL de la procédure protocolaire entrante (feuille 3 de 5)

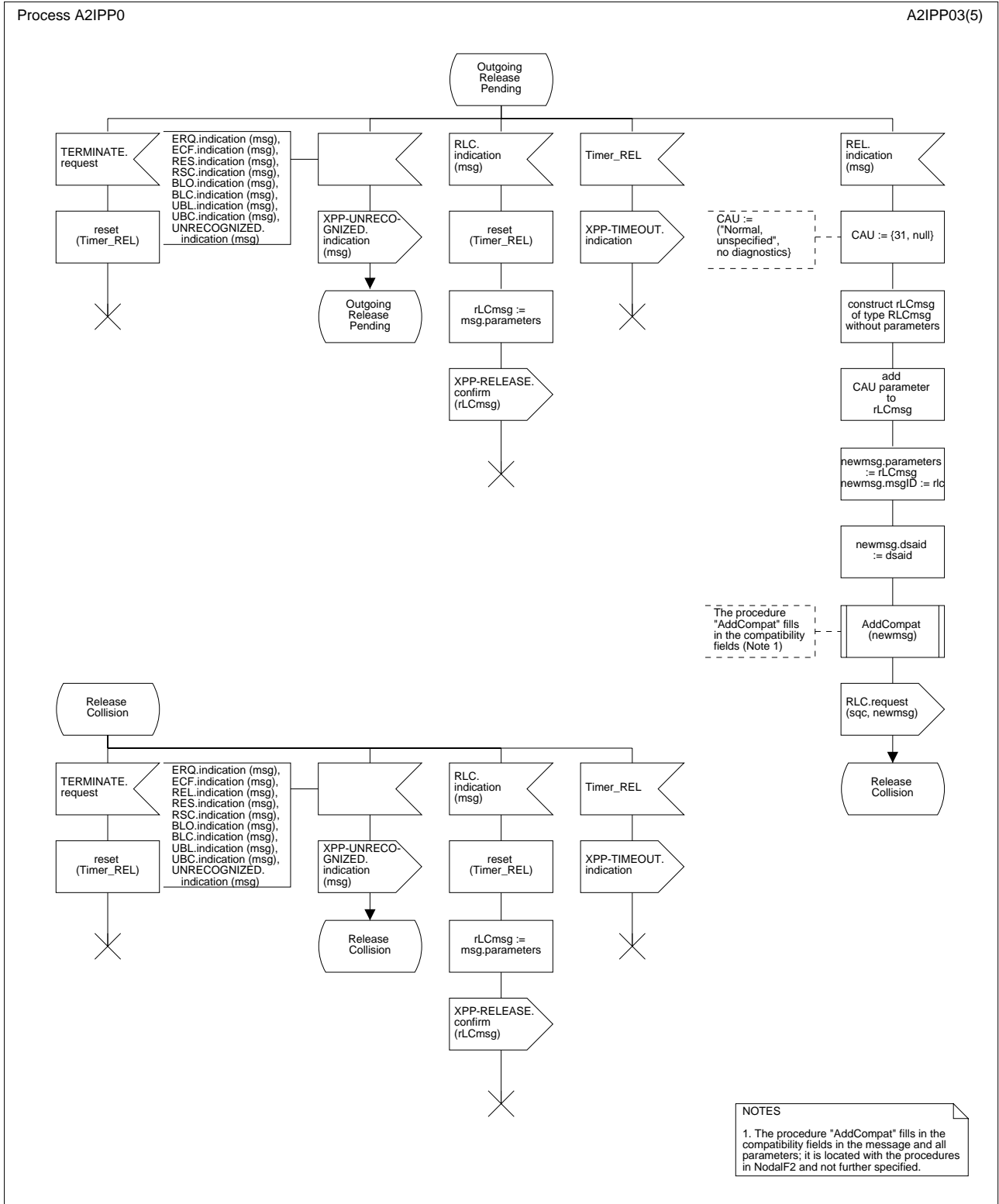
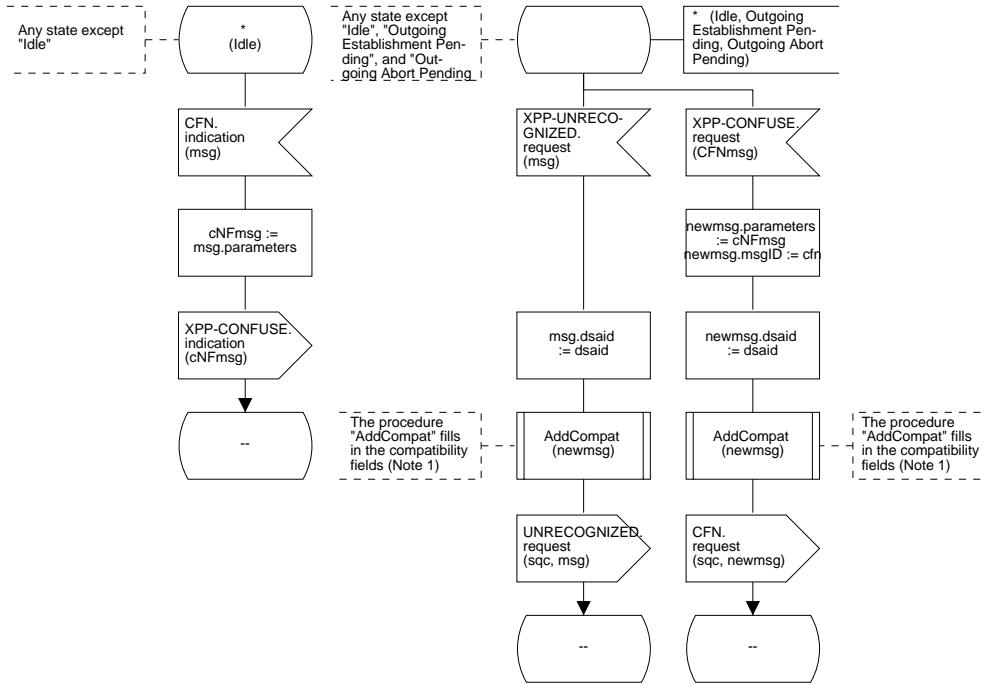


Figure B.5/Q.2630.1 – Diagramme SDL de la procédure protocolaire entrante (feuille 4 de 5)



NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.5/Q.2630.1 – Diagramme SDL de la procédure protocolaire entrante (feuille 5 de 5)

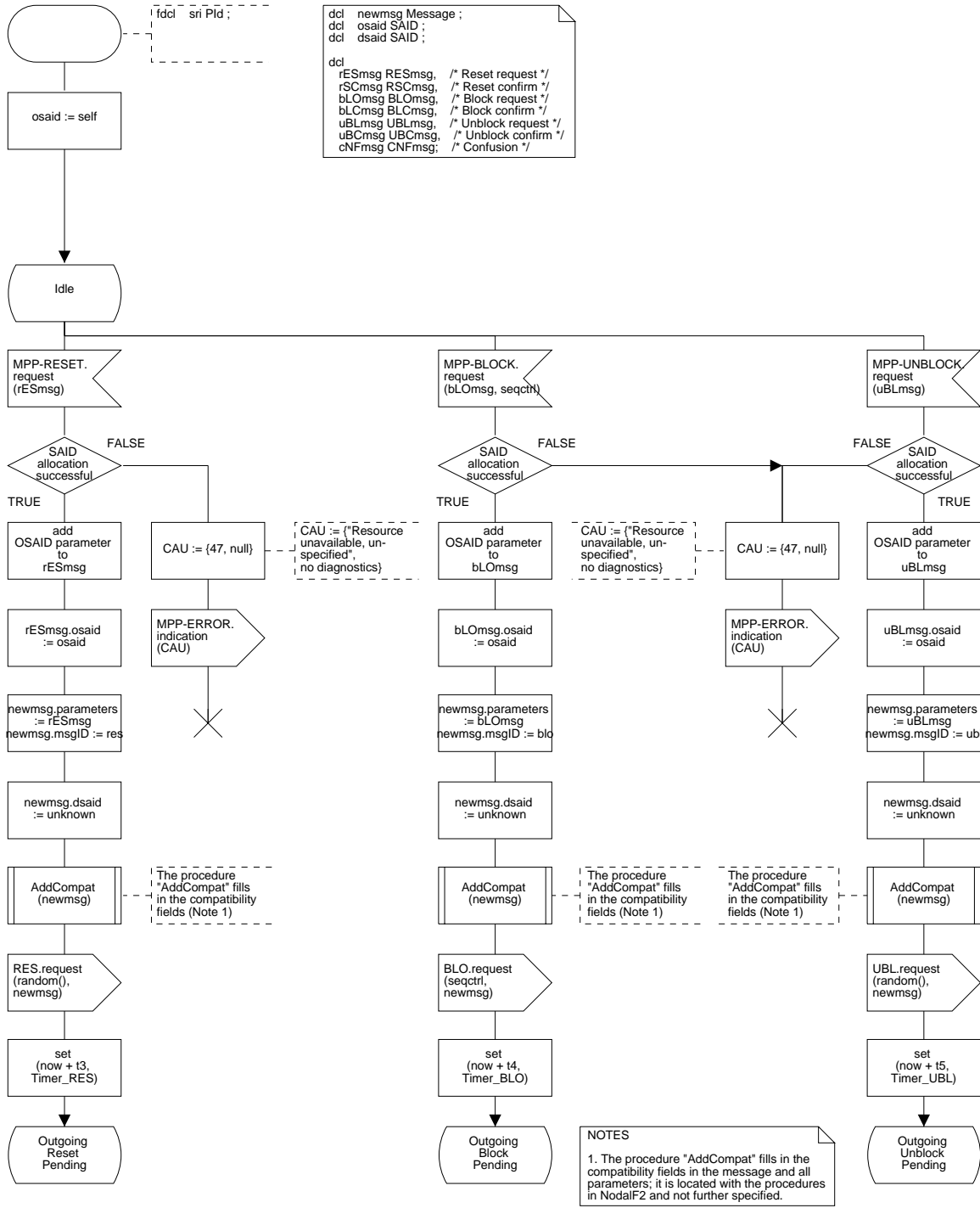


Figure B.6/Q.2630.1 – Diagramme SDL de la procédure protocolaire de maintenance (feuille 1 de 5)

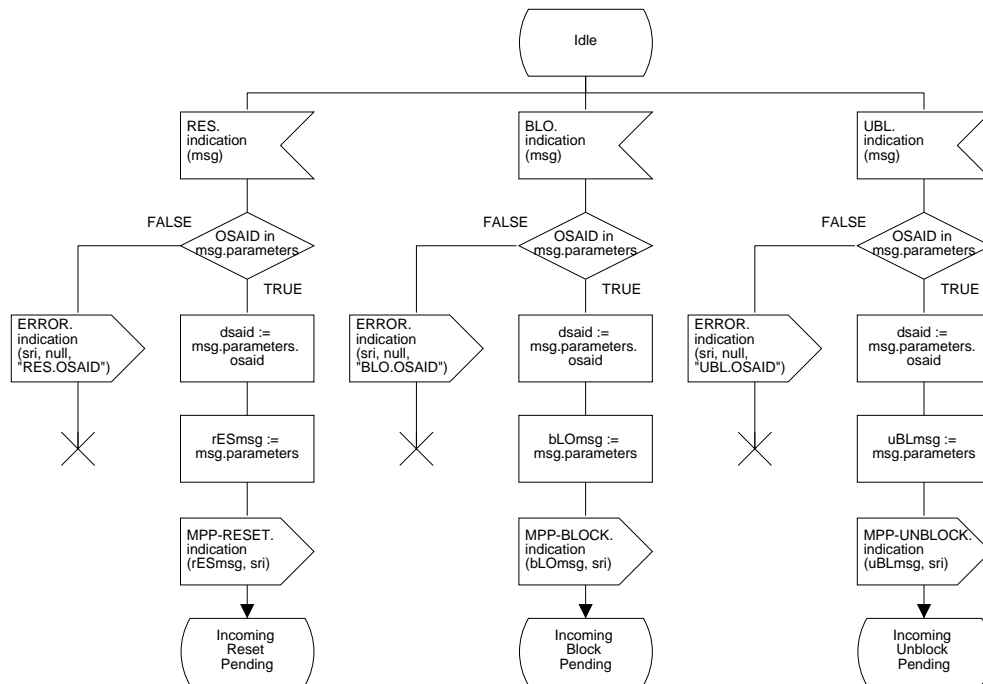


Figure B.6/Q.2630.1 – Diagramme SDL de la procédure protocolaire de maintenance (feuille 2 de 5)

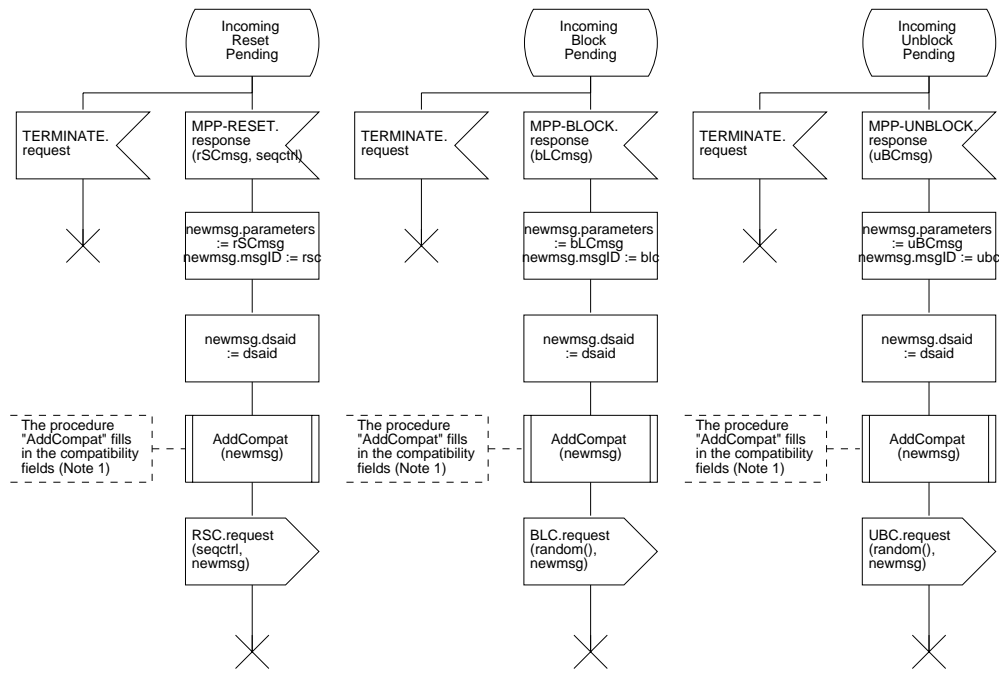
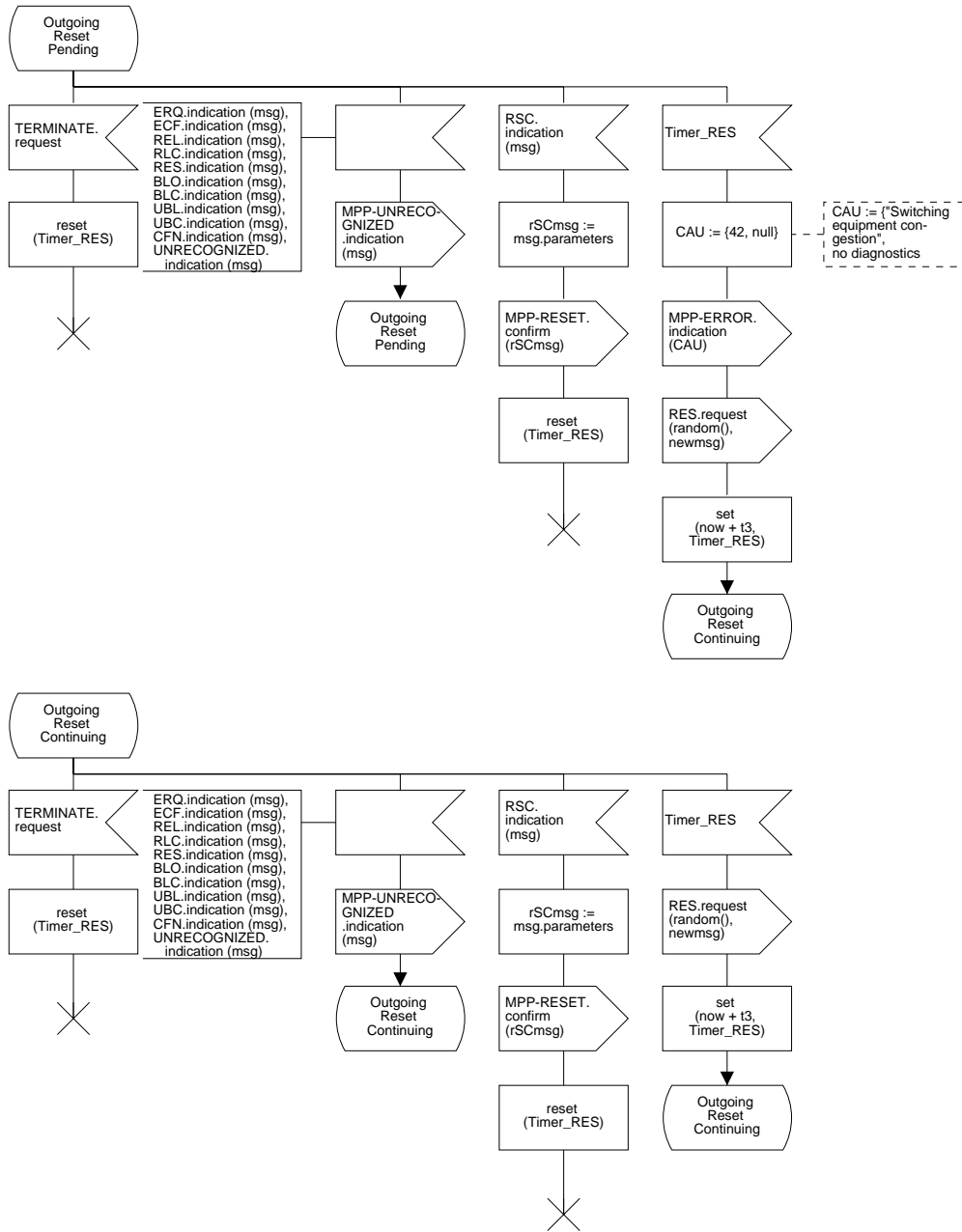


Figure B.6/Q.2630.1 – Diagramme SDL de la procédure protocolaire de maintenance (feuille 3 de 5)



NOTES
 1. The procedure "AddCompat" fills in the compatibility fields in the message and all parameters; it is located with the procedures in NodalF2 and not further specified.

Figure B.6/Q.2630.1 – Diagramme SDL de la procédure protocolaire de maintenance (feuille 4 de 5)

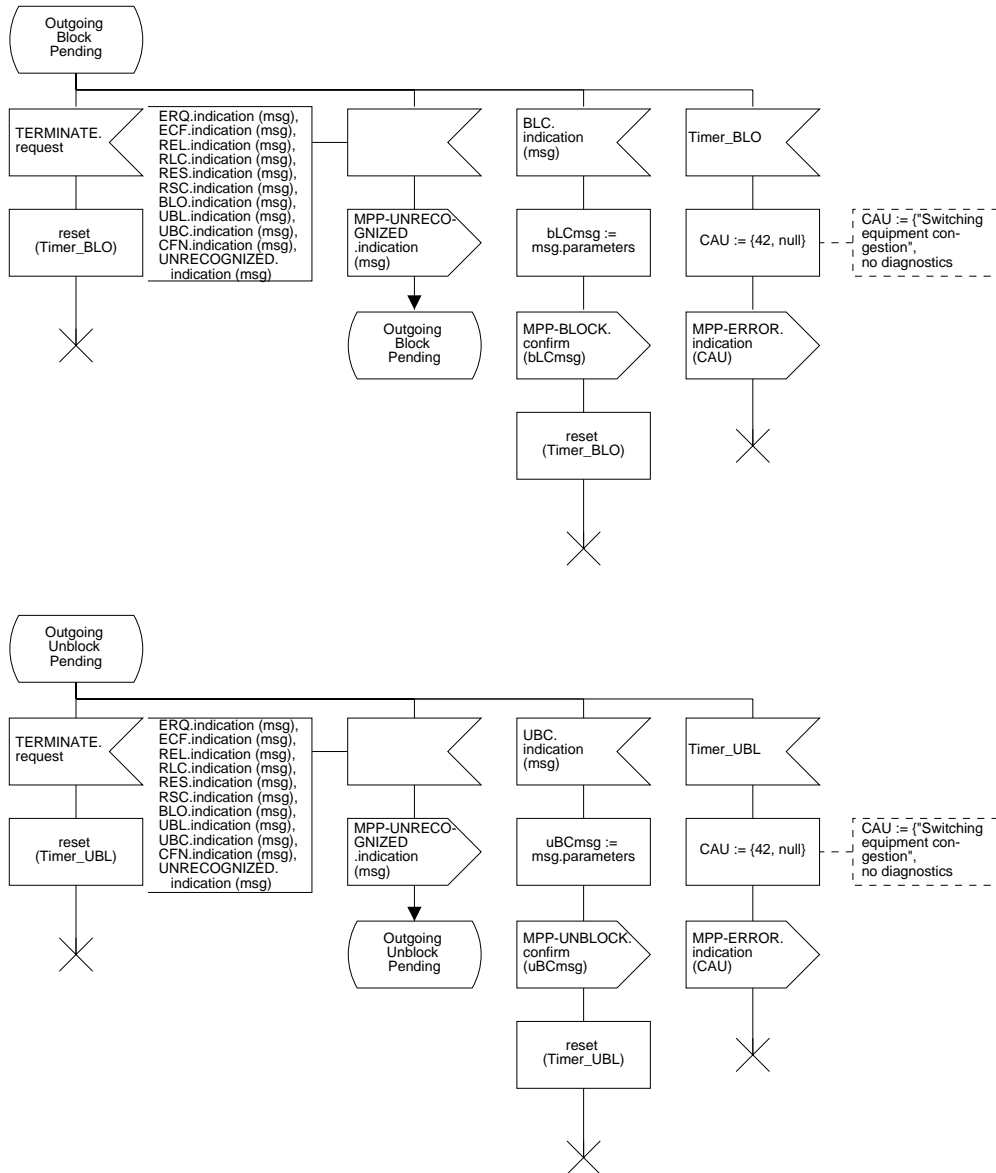


Figure B.6/Q.2630.1 – Diagramme SDL de la procédure protocolaire de maintenance (feuille 5 de 5)

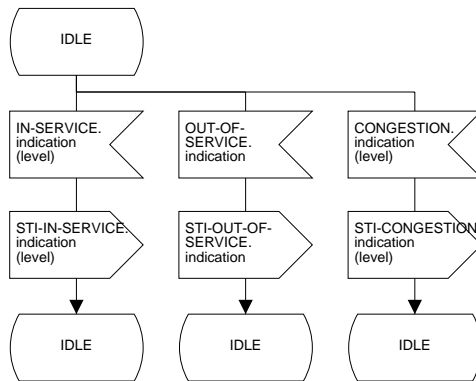
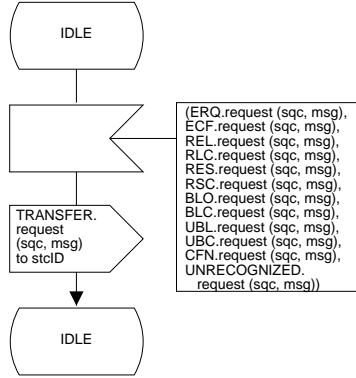
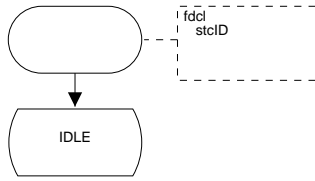


Figure B.7/Q.2630.1 – Diagramme SDL de l'interface de signalisation de transport (feuille 1 de 2)

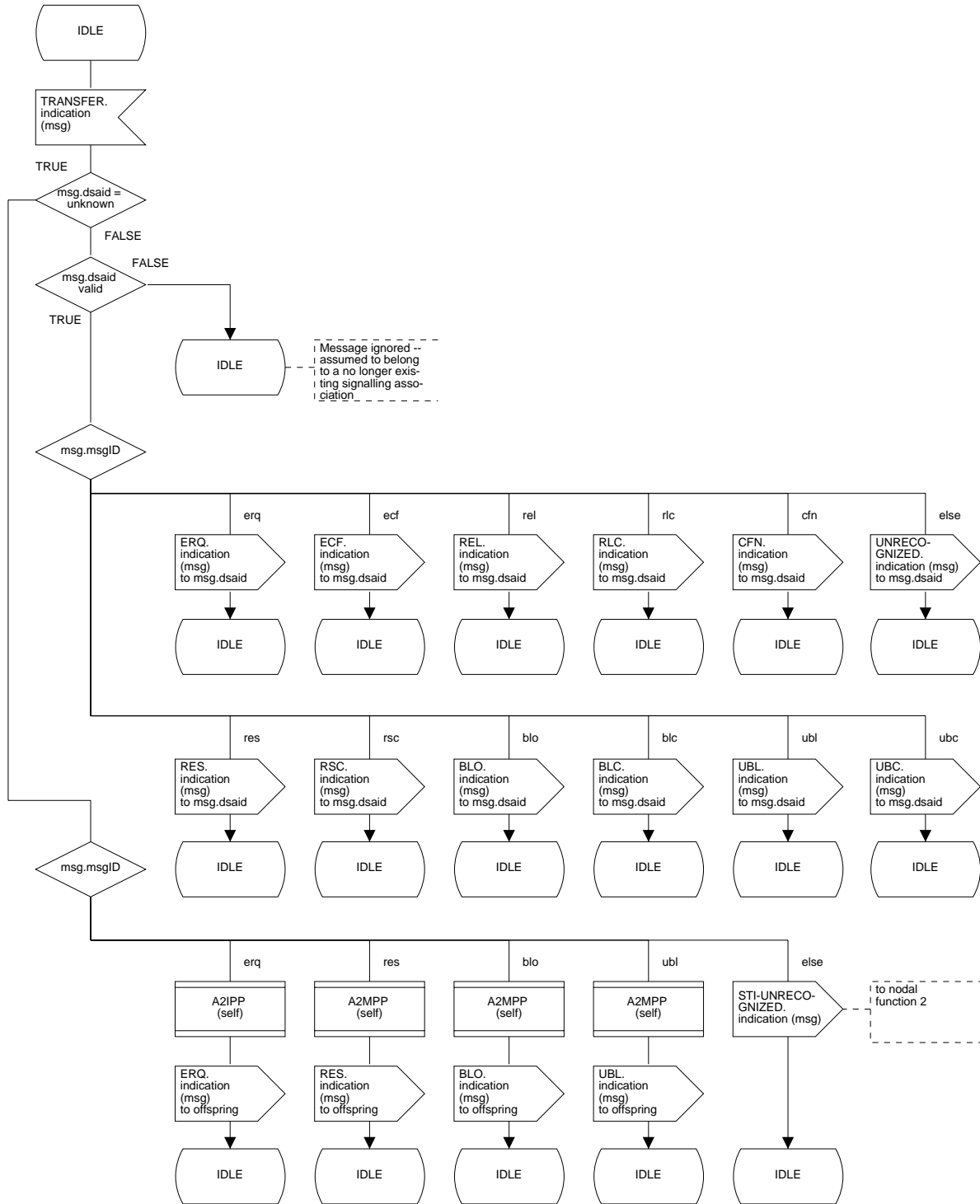


Figure B.7/Q.2630.1 – Diagramme SDL de l'interface de signalisation de transport (feuille 2 de 2)

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Réseaux câblés et transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, circuits téléphoniques, télégraphie, télécopie et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects généraux logiciels des systèmes de télécommunication

20868