



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**Q.774**

(06/97)

SÉRIE Q: COMMUTATION ET SIGNALISATION

Spécifications du système de signalisation n° 7 –  
Sous-système application de gestion des transactions

---

**Procédures du gestionnaire de transactions**

Recommandation UIT-T Q.774

(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
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N ° 7	Q.700–Q.849
Généralités	Q.700
Sous-système transport de messages	Q.701–Q.709
Sous-système commande des connexions sémaphores	Q.711–Q.719
Sous-système utilisateur téléphonie	Q.720–Q.729
Services complémentaires du RNIS	Q.730–Q.739
Sous-système utilisateur données	Q.740–Q.749
Gestion du système de signalisation n° 7	Q.750–Q.759
Sous-système utilisateur du RNIS	Q.760–Q.769
<b>Sous-système application de gestion des transactions</b>	<b>Q.770–Q.779</b>
Spécification des tests	Q.780–Q.799
Interface Q3	Q.800–Q.849
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
Généralités	Q.850–Q.919
Couche Liaison de données	Q.920–Q.929
Couche Réseau	Q.930–Q.939
Gestion usager-réseau	Q.940–Q.949
Description d'étape 3 des services complémentaires utilisant le système DSS 1	Q.950–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
Interfonctionnement avec le service INMARSAT de norme A	Q.1100–Q.1109
Interfonctionnement avec le service INMARSAT de norme B	Q.1110–Q.1149
Interfonctionnement avec le système du service mobile aéronautique par satellite d'INMARSAT	Q.1150–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1999
RNIS À LARGE BANDE	Q.2000–Q.2999

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

## **RECOMMANDATION UIT-T Q.774**

### **PROCEDURES DU GESTIONNAIRE DE TRANSACTIONS**

#### **Résumé**

La présente Recommandation a été révisée en vue d'apporter de nouvelles précisions (tant dans le texte que dans les diagrammes SDL) concernant les procédures de commande de dialogue utilisées pour mettre fin de façon normale ou anormale au dialogue. Des modifications ont également été apportées aux diagrammes SDL de la présente Recommandation, destinées à représenter de manière appropriée la création dynamique et l'arrêt des processus.

#### **Source**

La Recommandation UIT-T Q.774, révisée par la Commission d'études 11 de l'UIT-T (1997-2000), a été approuvée le 5 juin 1997 selon la procédure définie dans la Résolution n° 1 de la CMNT.

## 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.

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

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

Dans certains secteurs de la technologie 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 avait/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 1998

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

	<b>Page</b>
1	Introduction..... 1
1.1	Principes de base..... 1
1.2	Aperçu général..... 1
2	Adressage..... 1
3	Gestionnaire de transactions utilisant un service de réseau en mode sans connexion 1
3.1	Définition des sous-couches du TCAP ..... 1
3.2	Procédures de la sous-couche composant..... 2
3.2.1	Procédures normales..... 2
3.2.2	Procédures anormales ..... 13
3.2.3	Question de compatibilité..... 17
3.3	Procédures de la sous-couche transaction..... 18
3.3.1	Généralités ..... 18
3.3.2	Mappage entre les primitives de service de la sous-couche transaction et les types de message..... 18
3.3.3	Procédures normales..... 19
3.3.4	Procédures anormales relatives à la commande de transaction..... 22
	Annexe A – Diagrammes SDL du gestionnaire de transactions..... 24
A.1	Généralités ..... 24
A.2	Conventions utilisées ..... 29
A.3	Création dynamique de processus..... 29
A.4	Abréviations utilisées dans les diagrammes SDL..... 61



## **Recommandation Q.774**

### **PROCEDURES DU GESTIONNAIRE DE TRANSACTIONS**

*(Melbourne, 1988; révisée en 1993 et 1997)*

#### **1 Introduction**

Le gestionnaire de transactions (TC, *transaction capabilities*) permet à ses utilisateurs d'échanger des composants via des messages du sous-système application pour la gestion des transactions (TCAP, *transaction capabilities application part*). Il permet également, en tant qu'option, le transfert du nom du contexte d'application et d'informations d'utilisateur (c'est-à-dire des données qui ne sont pas des composants) entre deux utilisateurs TC. Les procédures décrites dans le présent paragraphe spécifient les règles gérant le contenu des informations et l'échange des messages TCAP entre utilisateurs du TC.

##### **1.1 Principes de base**

Afin de maximiser la flexibilité dans l'architecture des services et dans la manière de les mettre en œuvre, les procédures du TCAP se limitent à l'échange de composants et si l'option a été choisie à l'échange du nom du contexte d'application et d'informations d'utilisateur (c'est-à-dire des données qui ne sont pas des composants) entre utilisateurs du TC. Les procédures spécifiques aux applications (utilisateurs du TC) ne font pas partie du TCAP.

Lorsque la sélection d'une valeur de paramètre associée à une primitive qui est demandée par une couche inférieure (sous-couche) ne concerne pas cette couche (sous-couche), la valeur est simplement transmise à travers l'interface de service. La même remarque s'applique aux paramètres provenant d'une couche inférieure à travers l'interface de service, qui ne sont pas nécessaires aux fonctions du TCAP.

##### **1.2 Aperçu général**

Le paragraphe 2 présente les règles d'adressage pour les messages du TC. Le paragraphe 3 décrit le gestionnaire de transactions utilisant un service de réseau en mode sans connexion.

#### **2 Adressage**

Dans l'environnement du système de signalisation n° 7 en mode sans connexion, les messages du TC utiliseront n'importe quelles options d'adressage offertes par le sous-système commande des connexions sémaphores (SCCP, *signalling connection control part*). L'attribution et l'utilisation d'appellations globales peuvent dépendre du réseau et d'une application spécifique. Elles ne font pas partie de la présente spécification de protocole et ne sont pas examinées plus avant dans la présente Recommandation.

Les options d'adressage, lorsque d'autres fournisseurs de service de réseau sont utilisés, sont pour étude ultérieure.

#### **3 Gestionnaire de transactions utilisant un service de réseau en mode sans connexion**

##### **3.1 Définition des sous-couches du TCAP**

Les procédures TCAP sont divisées en procédure pour la sous-couche composant et procédure pour la sous-couche transaction. La procédure de la sous-couche composant fournit à un utilisateur du TC la possibilité d'invoquer des opérations distantes et de recevoir les réponses. La sous-couche composant reçoit également des informations de commande de dialogue et des informations d'utilisateur d'un

utilisateur du TC et produit des APDU de commande de dialogue. Elle utilise des fonctions de la sous-couche transaction pour la commande des transactions qui correspond à la capacité d'acheminer une séquence de composants et, à titre d'option, une partie de dialogue, dans des messages de la sous-couche transaction sur une connexion de bout en bout entre deux utilisateurs du TC.

## 3.2 Procédures de la sous-couche composant

La sous-couche composant fournit deux sortes de procédures:

- la gestion de dialogue;
- la gestion de composants.

### 3.2.1 Procédures normales

#### 3.2.1.1 Procédures de gestion des composants

##### 3.2.1.1.1 Mappage entre les primitives du service de gestion des composants du TC et les types de composant

La Recommandation Q.771 décrit les services fournis par la sous-couche composant en définissant l'interface de service entre l'utilisateur du TC et la sous-couche composant et l'interface entre la sous-couche composant et la sous-couche transaction. La gestion des composants correspond à la capacité de l'utilisateur du TC d'invoquer une procédure éloignée et de recevoir une réponse. Les procédures de gestion des composants traduisent les primitives de service de gestion des composants en composants qui constituent les unités de données du protocole (PDU, *protocol data unit*) de la sous-couche composant. Un mappage entre ces primitives et les PDU de la sous-couche composant est présentée au Tableau 1.

**Tableau 1/Q.774 – Mappage entre les primitives du service de gestion des composants du TC et les composants**

Primitives de service	Abréviation	Type de composant
TC-INVOKE	INV	invocation (Note 1)
TC-RESULT-L	RR-L	retour résultat (dernier) (Note 1)
TC-U-ERROR	RE	retour erreur (Note 1)
TC-U-REJECT	RJ	rejet (Note 1)
TC-R-REJECT	RJ	rejet (Note 1)
TC-L-REJECT	(Note 2)	
TC-RESULT-NL	RR-NL	retour résultat (non dernier)
TC-L-CANCEL	(Note 3)	
TC-U-CANCEL	(Note 3)	
NOTE 1 – Compatible avec les Recommandations X.219 et X.229.		
NOTE 2 – Le traitement de cette primitive est décrit en 3.2.2.2.		
NOTE 3 – Il n'y a pas de type de composant associé à cette primitive étant donné que l'effet est purement local.		

##### 3.2.1.1.2 Gestion des identificateurs d'invocation

Les identificateurs d'invocation sont attribués par l'extrémité initiatrice au moment de l'invocation. Un utilisateur du TC n'a pas besoin d'attendre qu'une opération soit terminée pour en invoquer une autre. A tout moment, un utilisateur du TC peut avoir un nombre quelconque d'opérations en cours avec une

extrémité distante (bien que cette dernière puisse rejeter un composant d'invocation par manque de ressources).

Chaque valeur d'identificateur d'invocation est associée à une invocation d'opération et à l'automate de la sous-couche composant correspondant. La gestion de cet automate a lieu seulement à l'extrémité qui invoque l'opération. L'autre extrémité réutilise cet identificateur d'invocation dans sa réponse à l'invocation d'opération et ne gère pas d'automate pour cet identificateur. Chaque extrémité peut invoquer des opérations de manière bidirectionnelle: chacune gère les automates des opérations qu'elle a invoquées, et est libre d'attribuer les identificateurs d'invocation indépendamment de l'autre.

Une valeur d'identificateur d'invocation peut être réattribuée lorsque l'automate correspondant revient à l'état repos. Cependant, une réattribution immédiate peut soulever des difficultés lorsque certaines situations anormales se produisent. Une valeur d'identificateur libérée (lorsque l'automate revient au repos) ne doit donc pas être réattribuée immédiatement; la manière dont cela est fait est dépendant de l'application et par conséquent n'est pas décrit dans la présente Recommandation.

Les deux applications homologues doivent avoir *a priori* une connaissance des classes et des temporisateurs associés à chaque opération.

Les états de la sous-couche composant et les transitions d'états sont décrits en 3.2.1.1.3.

### 3.2.1.1.3 Classes d'opérations

Voir Tableau 2.

**Tableau 2/Q.774 – Classes d'opérations**

Classe d'opération	Description
1	signalisation des succès et des échecs
2	signalisation des échecs uniquement
3	signalisation des succès seulement
4	aucune signalisation

Un type différent d'automate est défini pour chaque classe d'opération, les transitions d'états sont représentées aux Figures 1 à 4. Ces automates sont décrits ici du point de vue du protocole (composants émis/reçus), alors qu'ils sont décrits du point de vue du service (primitives) dans la Recommandation Q.771.

Les états de l'automate de chaque composant sont définis comme suit:

- *repos*: la valeur d'identificateur d'invocation n'est attribuée à aucune opération en cours.
- *opération envoyée*: la valeur d'identificateur d'invocation est attribuée à une opération qui n'a été ni exécutée, ni rejetée. L'état "opération envoyée" est déclenché lorsque le composant invocation est transmis.
- *attente pour rejet*: lorsqu'un composant indiquant l'achèvement d'une opération est reçu, l'utilisateur du TC récepteur peut rejeter ce résultat. L'état attente pour rejet est introduit afin que l'identificateur d'invocation soit gelé pour quelque temps, rendant ainsi le rejet possible.

Les transitions d'états sont déclenchées par:

- une primitive reçue de l'utilisateur du TC, entraînant la construction et éventuellement l'envoi d'un composant;
- la réception d'un composant de l'entité homologue;
- un certain nombre de situations indiquées dans les Figures 1 à 4, et correspondant aux situations suivantes:

- *annulation* – Une temporisation associée à une invocation. Cette temporisation d'invocation démarre lorsque le composant d'invocation est transmis à la sous-couche transaction. La primitive de demande TC-INVOKE indique une valeur de temporisation. Une situation d'annulation se produit lorsque l'utilisateur du TC qui invoque l'opération décide de l'annuler (primitive de demande TC-U-CANCEL) avant que le résultat final ne soit reçu (s'il y en a) ou avant la chute de la temporisation. A la réception d'une demande TC-U-CANCEL, la sous-couche composant arrête la temporisation; aucune réponse ultérieure ne sera remise à l'utilisateur du TC, et le TCAP réagira en accord avec les situations anormales décrites en 3.2.2.2.
- *situation de terminaison* – Lorsqu'un message "End" ou "Abort" est reçu, ou lorsqu'une terminaison prédéterminée est utilisée, le TCAP remet les opérations en suspens au repos.
- *fin de la temporisation d'invocation* – Une situation de fin de temporisation se présente lorsque la temporisation associée à une invocation d'opération prend fin: l'automate revient au repos et l'utilisateur du TC est notifié par une primitive d'indication TC-L-CANCEL (en cas d'opération de classes 1, 2 ou 3). Cette notification indique une situation anormale pour une opération de classe 1, ou termine une opération de classe 2 ou 3 pour laquelle aucun résultat n'a été reçu (situation normale).
- *fin de la temporisation de rejet* – Une situation de fin de temporisation de rejet se présente lorsque la temporisation associée à l'état attente pour rejet prend fin. Si cela se produit, la sous-couche composant suppose que l'utilisateur du TC a accepté le composant.

Dans les Figures 1 à 4, les composants contiennent soit des valeurs d'identificateur unique, soit des paires ordonnées d'identificateurs (i, y), où i représente l'identificateur d'invocation d'opération et y l'identificateur lié. Les diagrammes de transitions d'états sont modélisés pour une invocation d'opération individuelle avec l'identificateur i. La valeur de y ne se rapporte pas à l'identificateur i. Une opération d'invocation liée ne peut être acceptée que si l'automate lié est à l'état opération envoyée.

Les composants peuvent être reçus "bien formés" ou "mal formés". Les diagrammes montrent si cela est important. Si le fait qu'un composant soit reçu bien ou mal formé est important, le diagramme indique alors "réception" seulement.

Ces figures indiquent aussi des transitions normales et anormales. Les transitions anormales aboutissent aux procédures anormales examinées en 3.2.2. Par exemple, s'il y a fin de temporisation pour une opération de classe 1, cela dénote une situation anormale. La sous-couche composant libérera l'identificateur d'invocation et en informera l'application. Il appartient alors à l'application de décider si elle doit abandonner la transaction ou signaler une erreur à l'application homologue. Dans un autre exemple, la réception d'un composant de retour résultat négatif ne donne pas lieu à une transition valable pour une opération de classe 3. Il résulterait de ce type d'erreur que la sous-couche composant formerait un composant rejet, avec un code de problème "retour erreur problème – retour erreur non attendue". Elle informerait l'utilisateur du TC local par une primitive TC-L-REJECT de sorte que le composant rejet pourrait être envoyé (si l'utilisateur le souhaitait) avec la prochaine primitive de gestion de dialogue émise.

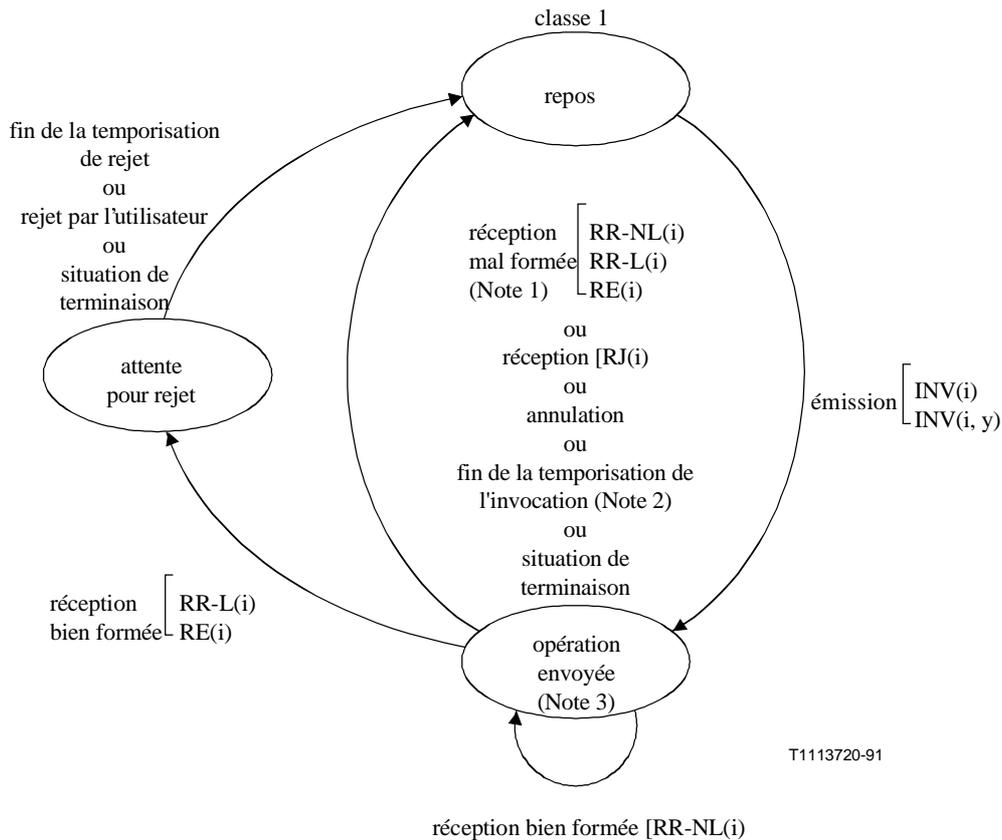
Les opérations de classe 1 notifient une réponse d'échec ou de succès. Un rejet en cas d'erreur de protocole peut aussi se produire. Lors de l'invocation d'une opération de classe 1, l'extrémité initiatrice gardera l'identificateur i actif jusqu'à ce qu'une "dernière" réponse, qui ne puisse plus être rejetée, soit reçue. Un identificateur peut être libéré localement, à la discrétion de l'utilisateur du TC. Un identificateur peut aussi être libéré à la fin de temporisation de l'invocation de l'opération. Cela est indiqué sur la Figure 1.

Les opérations de classe 2 signalent seulement les échecs. Un rejet en cas d'erreur de protocole peut aussi se produire. Lors de l'invocation d'une opération de classe 2, l'extrémité initiatrice gardera l'identificateur i actif, jusqu'à ce qu'une réponse, qui ne puisse plus être rejetée, soit reçue, ou jusqu'à ce

que la temporisation<sup>1</sup> fin ou jusqu'à ce qu'une situation de fin ou d'annulation apparaisse. Ceci est décrit dans la Figure 2.

Les opérations de classe 3 ne notifient que le succès. Un rejet en cas d'erreur de protocole peut aussi se produire. Lors de l'invocation d'une opération de classe 3, l'extrémité initiatrice gardera l'identificateur i actif jusqu'à ce qu'une réponse, qui ne puisse plus être rejetée, soit reçue, jusqu'à ce que la temporisation<sup>2</sup> chute ou jusqu'à ce qu'une situation de fin ou d'annulation apparaisse. Ceci est décrit dans la Figure 3.

Les opérations de classe 4 ne notifient pas leur résultat. Un rejet en cas d'erreur de protocole peut aussi se produire. Lors de l'invocation d'une opération de classe 4, l'extrémité initiatrice gardera l'identificateur i actif jusqu'à ce qu'un rejet soit reçu, ou jusqu'à ce que la temporisation<sup>3</sup> chute ou jusqu'à ce qu'une situation d'annulation ou de fin apparaisse. Ceci est décrit dans la Figure 4.



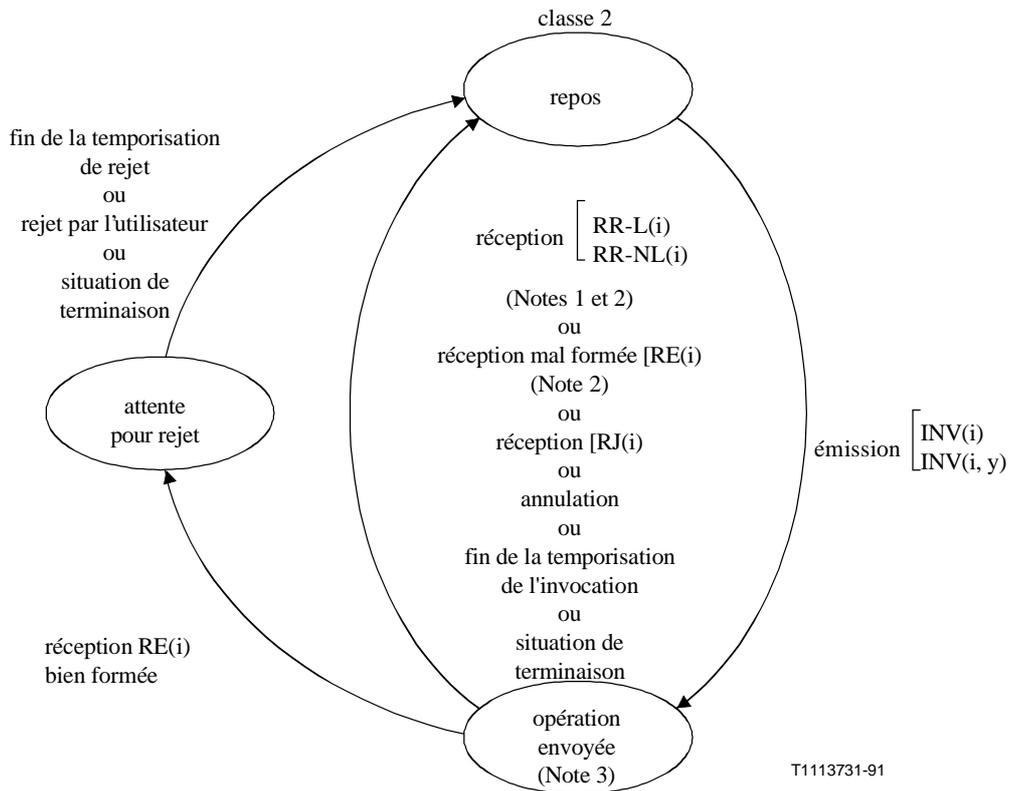
NOTE 1 – Dans ces situations, l'utilisateur du TC est informé et la transition intervient lorsque l'envoi du rejet est déclenché.

NOTE 2 – Il s'agit d'opérations anormales.

NOTE 3 – Quand une invocation liée INV(x, i) est reçue, l'existence de l'automate i est vérifiée pour s'assurer qu'il se trouve à l'état opération envoyée, mais il n'y a pas de conséquence pour l'automate.

**Figure 1/Q.774 – Classe d'opération 1**

- <sup>1</sup> Une fin de temporisation pour une opération de classe 2 est une situation "normale".
- <sup>2</sup> Une fin de temporisation pour une opération de classe 3 est une situation "normale".
- <sup>3</sup> Une fin de temporisation pour une opération de classe 4 est une situation "normale".

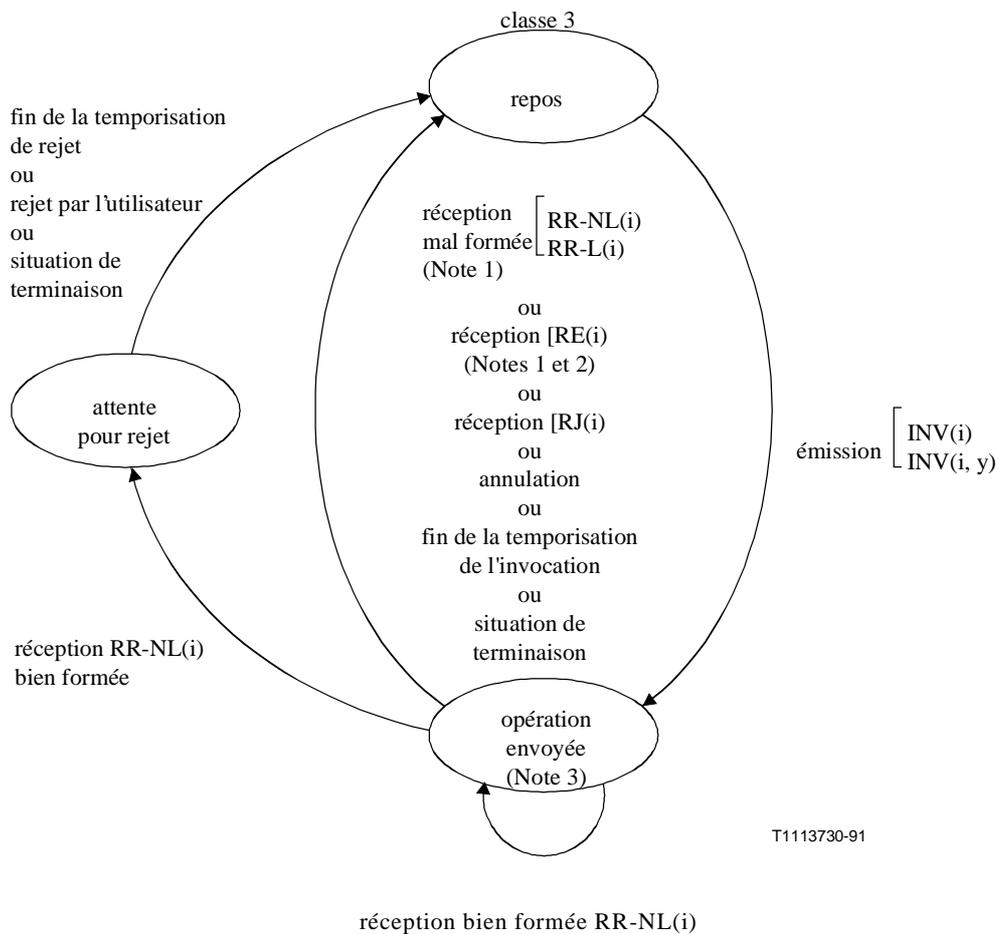


NOTE 1 – Il s'agit de situations anormales.

NOTE 2 – Dans ces situations, l'utilisateur du TC est informé et la transition intervient lorsque l'envoi du rejet est déclenché.

NOTE 3 – Quand une invocation liée INV(x, i) est reçue, l'existence de l'automate i est vérifiée pour s'assurer qu'il se trouve à l'état opération envoyée, mais il n'y a pas de conséquence pour l'automate.

**Figure 2/Q.774 – Classe d'opération 2**

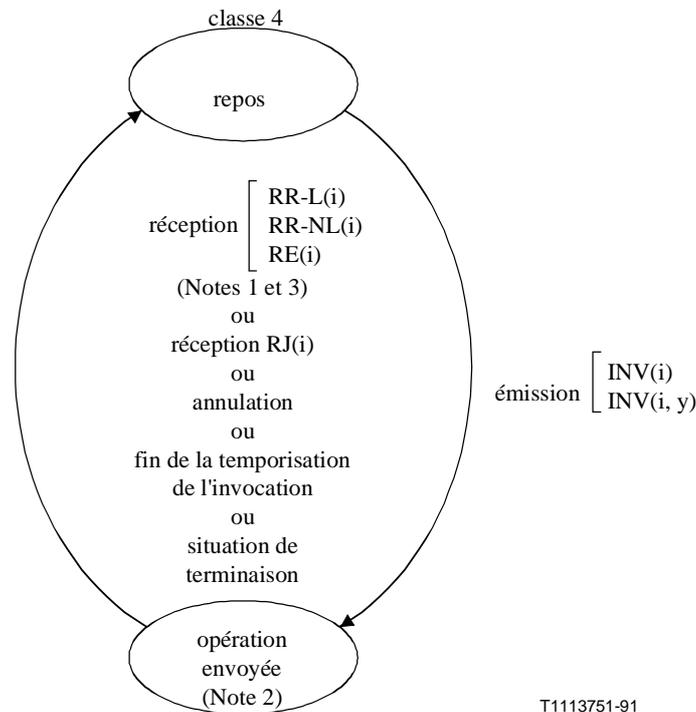


NOTE 1 – Dans ces situations, l'utilisateur du TC est informé et la transition intervient lorsque l'envoi du rejet est déclenché.

NOTE 2 – Il s'agit de situations anormales.

NOTE 3 – Quand une invocation liée INV(x, i) est reçue, l'existence de l'automate i est vérifiée pour s'assurer qu'il est à l'état opération envoyée, mais il n'y a pas de conséquence pour l'automate.

**Figure 3/Q.774 – Classe d'opération 3**



T1113751-91

NOTE 1 – Il s'agit de situations anormales.

NOTE 2 – Quand une invocation liée INV(x, i) est reçue, l'existence de l'automate i est vérifiée pour s'assurer qu'il se trouve à l'état opération envoyée, mais il n'y a pas de conséquence pour l'automate.

NOTE 3 – Dans ces situations, l'utilisateur du TC en est informé et la transition se produit lorsque l'émission du rejet est initialisée.

**Figure 4/Q.774 – Classe d'opération 4**

#### 3.2.1.1.4 Exemples d'échange de composants

Quelques exemples d'échanges de composants compatibles avec la Recommandation X.229 (serveur d'opérations distantes) sont présentés dans la Figure 5. Les échanges montrent des cas de séquences de composants valables liés à une opération invoquée.

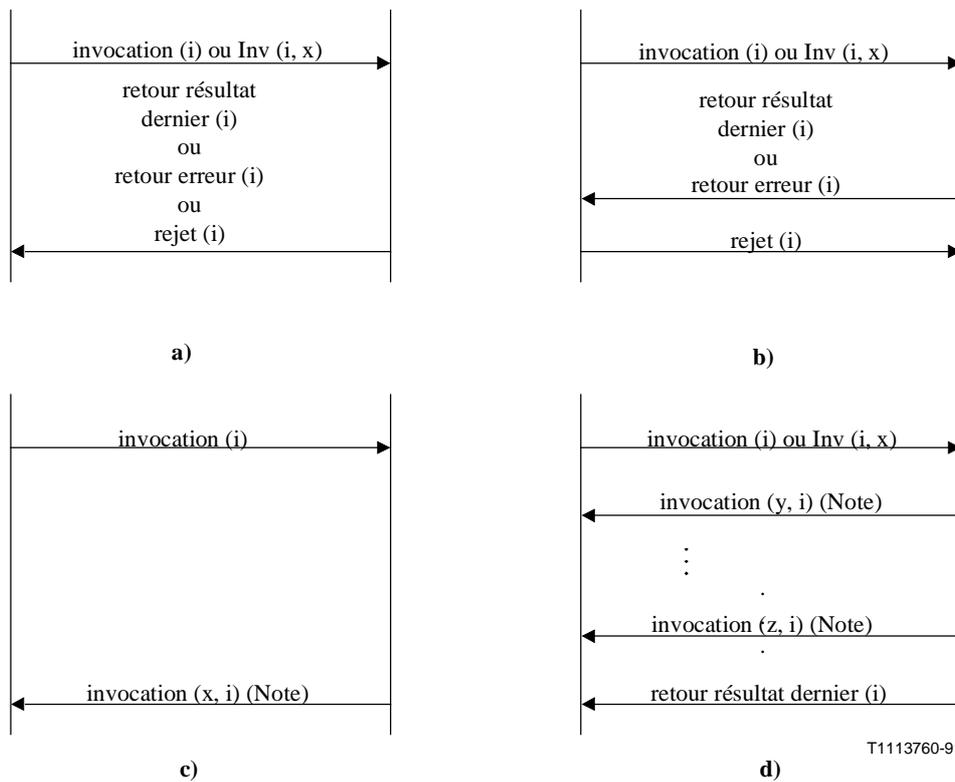
La Figure 6 qui développe les Recommandations X.219 et X.229, montre que le sous-système TCAP permet des retours résultats multipliés à la suite d'une même opération afin de segmenter un résultat dans le cadre d'un service de réseau en mode sans connexion.

#### 3.2.1.2 Commande de dialogue par l'intermédiaire des primitives de la sous-couche composant

Les primitives de demande TC-UNI, TC-BEGIN, TC-CONTINUE et TC-END sont utilisées par un utilisateur du TC pour commander le transfert de composants.

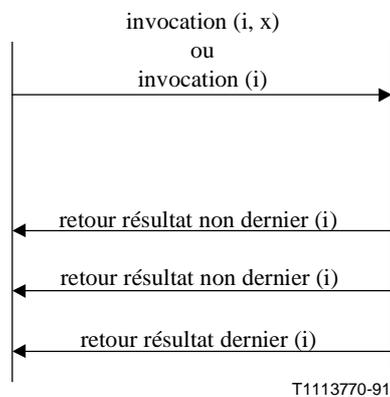
Certaines primitives de demande de commande de dialogue TC peuvent également entraîner la construction d'une unité APDU de commande de dialogue, si un paramètre de nom de contexte d'application figure dans la primitive de demande TC-BEGIN.

Le mappage entre les primitives de gestion du dialogue et les unités APDU de commande de dialogue est présentée sur le Tableau 3.



NOTE – Aucun changement de l'automate du composant de l'invocation initiale.

**Figure 5/Q.774 – Echanges de composants compatibles avec la Recommandation X.229**



**Figure 6/Q.774 – Résultat segmenté**

**Tableau 3/Q.774 – Mappage entre les primitives du service de gestion du dialogue TC et les unités APDU de commande de dialogue**

<b>Primitive TC (demande)</b>	<b>Unité APDU de commande de dialogue</b>
TC-UNI	dialogue unidirectionnel (AUDT)
TC-BEGIN	demande de dialogue (AARQ)
TC-CONTINUE	réponse de dialogue (AARE[accepté]) (Note 1)
TC-END	réponse de dialogue (AARE[accepté]) (Note 2)
TC-U-ABORT	dialogue abandonné (ABRT) Réponse de dialogue (AARE[rejeté]) (Note 3)
<p>NOTE 1 – Cela s'applique uniquement à la première primitive TC-CONTINUE vers l'arrière [c'est-à-dire lorsque le dialogue est dans l'état initialisation envoyée/initialisation reçue (IS/IR)].</p> <p>NOTE 2 – Cela s'applique seulement aux primitives de demande TC-END émises en réponse à une primitive d'indication TC-BEGIN (c'est-à-dire lorsque le dialogue est dans l'état IS/IR).</p> <p>NOTE 3 – Cela s'applique seulement lorsque le dialogue est établi (c'est-à-dire avant le premier message continue vers l'arrière) et lorsque le paramètre "raison de l'abandon" contenu dans la primitive de demande TC-U-ABORT indique "nom du contexte d'application non fourni" ou dialogue refusé"</p>	

Les unités PDU de commande de dialogue sont acheminées dans la partie dialogue du message TC. La partie dialogue, lorsqu'elle est présente, est concaténée avec la partie composant et transférée dans la sous-couche transaction en tant que données d'utilisateur de la primitive de service TR (couche de transaction) correspondante.

Les composants d'un message sont remis à l'utilisateur du TC distant dans l'ordre dans lequel ils ont été reçus par la sous-couche composant d'origine de l'utilisateur du TC local. Les primitives d'indication correspondantes sont utilisées par la sous-couche composant pour informer l'utilisateur du TC à l'extrémité réception, de l'état du dialogue.

Un utilisateur du TC utilise une primitive de demande de commande de dialogue (TC-UNI, TC-BEGIN, TC-CONTINUE et TC-END) pour déclencher la transmission de tous les composants précédemment passés avec le même identificateur de dialogue, à l'exception des primitives TC-ABORT, qui déclenchent l'élimination des composants en attente. La seule autre fonction d'une primitive de commande de dialogue de la sous-couche composant est de déclencher la demande de service correspondante vers la sous-couche transaction, c'est-à-dire la sous-couche où le service de commande de transaction est fourni. Le mappage entre les primitives de commande de dialogue et de commande de transaction est présentée dans le Tableau 4.

**Tableau 4/Q.774 – Mappage entre les primitives du service de traitement de dialogue du TC et les primitives TR**

Primitive TC	Primitive TR
TC-UNI	TR-UNI
TC-BEGIN	TR-BEGIN
TC-CONTINUE	TR-CONTINUE
TC-END	TR-END
TC-U-ABORT	TR-U-ABORT
TC-P-ABORT	TR-P-ABORT

### **Etablissement d'un dialogue**

Une primitive de demande TC-BEGIN se traduit par une primitive de demande TR-BEGIN, laquelle débute une transaction et transmet n'importe quel composant (0 ou plus) passé par l'interface avec le même identificateur de dialogue. Si le paramètre nom du contexte d'application figure dans la primitive de demande TC-BEGIN, une unité APDU de demande de dialogue (AARQ, *dialogue request*) est également envoyée, une fois concaténée dans la partie composant.

L'adresse de destination et l'adresse d'origine fournies dans la primitive de demande TC-BEGIN sont mises en mémoire par la sous-couche transaction avant l'envoi du message Begin.

A l'extrémité de destination, une primitive d'indication TR-BEGIN est reçue par la sous-couche composant. Cela provoque la remise d'une primitive d'indication TC-BEGIN, contenant le cas échéant des informations de dialogue, qui permettra l'établissement du dialogue avec l'utilisateur du TC, suivie par les primitives de gestion des composants associées à chacun des composants reçus (s'il y en a).

La sous-couche transaction de réception met en mémoire l'adresse d'origine reçue comme adresse de destination pour cette transaction et sa propre adresse comme adresse d'origine (extraite de la mémoire ou de l'adresse de destination de la primitive d'indication N-UNITDATA). L'utilisateur du TC reçoit une primitive d'indication TC-BEGIN contenant l'adresse de destination et l'adresse d'origine reçues.

### **Confirmation d'un dialogue**

Si l'utilisateur du TC a reçu un paramètre nom du contexte d'application dans la primitive d'indication TC-BEGIN et que ce contexte d'application soit acceptable, l'utilisateur du TC devra inclure la même valeur dans la première primitive de demande TC-CONTINUE vers l'arrière. Cela déclenche l'envoi d'une unité APDU de réponse de dialogue (AARE, *dialogue response*) concaténée à un composant quelconque contenu dans un message Continue.

Si le nom du contexte d'application proposé n'est pas acceptable, l'utilisateur du TC pourra souhaiter poursuivre le dialogue mais offrir un nom de contexte d'application différent dans la première primitive de demande TC-CONTINUE vers l'arrière. Cela provoque l'envoi d'une unité APDU de réponse de dialogue (AARE, *dialogue response*), concaténée à un composant quelconque contenu dans un message Continue.

Dans les deux cas, le champ "résultat" de l'unité APDU AARE est indiqué "accepté" alors que le champ "diagnostic de l'origine du résultat" est indiqué soit sur "utilisateur du service de dialogue (zéro)", soit "utilisateur du service de dialogue (aucune raison indiquée)". Le choix entre l'une ou l'autre des valeurs dépend des applications et elles sont, aux fins de la présente Recommandation, équivalentes du point de vue sémantique.

L'utilisateur du TC qui répond émet une primitive de demande TC-CONTINUE avec la possibilité d'ajouter facultativement un paramètre d'adresse d'origine qui n'est utilisé que si l'utilisateur du TC décide de modifier sa propre adresse (adresse d'origine du côté B).

La sous-couche transaction met en mémoire les nouvelles adresses d'origine et envoie un message de continuation vers l'utilisateur du TC de départ.

La sous-couche transaction à l'extrémité qui initialise la transaction reçoit le message de continuation et, comme c'est le premier message vers l'arrière, met en mémoire l'adresse d'origine dans la primitive d'indication N-UNITDATA comme l'adresse de destination pour la transaction. L'adresse d'origine mise en mémoire pour cette transaction reste inchangée. Ces adresses seront utilisées pour tous les messages ultérieurs concernant cette transaction et restent inchangées pendant la durée de la transaction.

### **Continuation d'un dialogue**

Une primitive de demande TC-CONTINUE se traduit en une primitive de demande TR-CONTINUE laquelle transmet les composants passés à l'interface avec le même identificateur de dialogue. Si des composants rejet (voir 3.2.2.2) ont été construits par la sous-couche composant pour ce dialogue, ils sont également transmis.

A l'extrémité de destination, une primitive d'indication TR-CONTINUE reçue par la sous-couche composant provoque l'envoi d'une primitive d'indication TC-CONTINUE à l'utilisateur du TC, suivie par les primitives de gestion des composants associées à chacun des composants reçus.

Aucune unité APDU de commande de dialogue n'est échangée à ce stade de la transaction. Si des unités APDU de commande de dialogue sont échangées pendant l'établissement du dialogue, le nom du contexte d'application envoyé dans l'unité APDU AARE est supposé correspondre au contexte d'application qui est mis en œuvre entre les utilisateurs du TC pendant la durée du dialogue. Le gestionnaire de transactions ne vérifie pas cela mais traite la présence d'une unité APDU de contrôle de dialogue à ce stade du dialogue comme un événement anormal. Pendant cette phase, une partie de dialogue associée à une syntaxe abstraite définie du point de vue de l'utilisateur pourrait être éventuellement présente.

### **Terminaison d'un dialogue**

Si l'utilisateur du TC envoie une primitive de demande TC-END pour une terminaison de base du dialogue de façon à réagir immédiatement à une primitive d'indication TC-BEGIN contenant un nom de contexte d'application, il provoque la formation d'une unité APDU de réponse de dialogue (AARE) dont le champ "résultat" indique "accepté" alors que le champ "diagnostic de l'origine du résultat" indique soit "utilisateur du service de dialogue (zéro)" soit "utilisateur du service de dialogue (aucune raison indiquée)". Le choix entre l'une ou l'autre des valeurs du champ "diagnostic de l'origine du résultat" dépend des implémentations et aux fins de la présente Recommandation ces valeurs sont équivalentes du point de vue sémantique. L'unité APDU AARE concaténée à un composant quelconque est passée afin d'être transmise dans une primitive de demande TR-END.

La primitive d'indication TC-END provoque la mise au repos de l'automate associé au dialogue.

Dans le cas d'une terminaison de base d'un dialogue, les composants passés à l'interface et les composants de rejet construits par la sous-couche composant pour ce dialogue sont passés pour transmission à la sous-couche transaction dans une primitive de demande TR-END, puis le dialogue est terminé.

A l'extrémité de destination, un dialogue est terminé lorsque chaque composant (s'il y en a) accompagnant la primitive d'indication TR-END a été remis à l'utilisateur du TC à l'aide de la primitive de gestion de composant appropriée qui suit l'indication TC-END.

La sous-couche composant ne vérifie pas, lorsqu'un utilisateur du TC demande la terminaison d'un dialogue, que tous les automates de la sous-couche composant associés à ce dialogue sont remis au repos. De la même façon, la sous-couche composant ne vérifie pas que tous les automates associés à ce dialogue sont remis au repos lorsque les composants accompagnant une primitive d'indication TR-END ont été remis. Dans une situation de terminaison, les automates n'étant pas au repos sont remis au repos lorsque la primitive de demande TR-END est passée à la sous-couche transaction (à l'extrémité origine),

ou lorsque tous les composants accompagnateurs ont été remis à l'utilisateur du TC (à l'extrémité destination); les composants en attente de transmission sont éliminés.

Lorsqu'un utilisateur du TC a reçu une primitive d'indication TC-BEGIN accompagnée d'un paramètre nom du contexte d'application qu'il estime inacceptable, et qu'il ne souhaite pas poursuivre le dialogue, il émet une primitive de demande TC-U-ABORT. Dans la primitive de demande TC-U-ABORT, le réglage du paramètre "raison de l'abandon" sur la valeur "nom du contexte d'application non fourni" provoque le formatage d'une unité APDU (AARE) de réponse de dialogue. Le réglage des valeurs des différents champs dans l'unité APDU AARE est le suivant: le champ de résultat est réglé sur "rejet permanent" et le "diagnostic de l'origine du résultat" est "utilisateur du service de dialogue (nom du contexte d'application non fourni)". Cette unité APDU est envoyée, concaténée à un composant quelconque dans le champ de données de l'utilisateur de la primitive de demande TR-U-ABORT.

Si le paramètre "raison de l'abandon" dans la primitive de demande TC-U-ABORT est absent ou a une valeur autre que "nom du contexte d'application non fourni" ou "dialogue refusé", cela indique une fin anormale de dialogue, ce cas est décrit en 3.2.2.

Lorsqu'un utilisateur du TC a reçu une primitive d'indication TC-BEGIN contenant une information d'utilisateur qu'il estime inacceptable, il peut émettre une primitive de demande TC-U-ABORT dont le paramètre "raison de l'abandon" est réglé sur la valeur "dialogue refusé". L'émission d'une telle primitive provoque le formatage d'une unité APDU de réponse de dialogue (AARE). Le réglage des valeurs des différents champs dans l'unité APDU AARE est le suivant: le champ "nom de contexte d'application" est identique au champ reçu dans le paramètre "nom de contexte d'application" de la primitive de demande TC-U-ABORT, le champ "résultat" est réglé sur "rejet permanent" et le champ "diagnostic de l'origine du résultat" indique soit "utilisateur du service de dialogue (zéro)" soit "utilisateur du service de dialogue (aucune raison indiquée)".

Une terminaison prédéterminée et un abandon d'un dialogue par l'utilisateur du TC ne déclenchent pas la transmission des composants en suspens. Tous les automates associés à ce dialogue sont remis au repos et les composants sont éliminés.

### **3.2.2 Procédures anormales**

#### **3.2.2.1 Commande de dialogue**

Toute situation anormale due à une composante énoncée ou reçue hors séquence, détectée par la sous-couche composant se traduit par le rejet d'un composant et une notification à l'utilisateur local du TC. L'abandon d'un dialogue est toujours le reflet d'une décision prise par:

- la sous-couche composant dans le cas où une partie de dialogue incorrecte est reçue, c'est-à-dire incorrecte du point de vue de la syntaxe ou incompatible avec l'état de la transaction sous-jacente. Ce dernier cas correspond à une situation dans laquelle une partie de dialogue manque alors que sa présence est obligatoire (par exemple une unité APDU AARQ a été envoyée dans un message Begin, mais aucune unité APDU AARE n'a été reçue dans le premier message Continue vers l'arrière) ou qu'une partie de dialogue est reçue au mauvais moment (par exemple, une unité APDU de dialogue est reçue pendant l'état actif d'une transaction). Du côté où le phénomène normal est décelé, une primitive d'indication TC-P-ABORT est envoyée à l'utilisateur local du TC avec le paramètre "abandon par le fournisseur" dans la primitive réglée sur "dialogue anormal". Par ailleurs, une primitive de demande TR-U-ABORT est envoyée en tant que données d'utilisateur à la sous-couche transaction, avec une donnée d'usager APDU ABRT. Le champ de la cause de l'abandon de l'unité APDU ABRT est indiqué sur "fournisseur du service de dialogue" et le champ d'information d'usager est absent. A l'extrémité réceptrice, la sous-couche composant envoie une indication TC-P-ABORT après réception d'une unité APDU ABRT en tant que donnée d'utilisateur dans la primitive d'indication TR-U-ABORT, le champ de la cause de l'abandon de l'unité APDU ABRT étant "fournisseur du service de dialogue".

Si des composants sont reçus dans le message avec une partie dialogue incorrecte, ils sont éliminés;

- la sous-couche transaction d'abandonner la transaction sous-jacente. La sous-couche composant libère les automates d'opération de ce dialogue, élimine les composants en suspens et passe aux utilisateurs du TC une primitive d'indication TC-P-ABORT;
- l'utilisateur du TC d'abandonner le dialogue. A l'extrémité origine, une demande TC-U-ABORT est reçue de l'utilisateur du TC: les automates des composants actifs pour ce dialogue sont au repos, et une demande TR-U-ABORT est passée à la sous-couche transaction. A l'extrémité destination une primitive d'indication TR-U-ABORT correspondante est reçue de la sous-couche transaction, les automates des composants actifs pour ce dialogue sont au repos, et une primitive d'indication TC-U-ABORT est passée à l'utilisateur du TC.

Si une primitive de demande TC-U-ABORT est émise pendant l'état actif d'un dialogue, et que le paramètre "raison de l'abandon" est absent ou réglé sur "défini par l'utilisateur", une unité APDU de fin de dialogue (ABRT, *dialogue abort*) est formatée seulement dans le cas où les unités APDU AARQ/AARE sont utilisées pendant la phase d'établissement du dialogue. Les données d'utilisateur fournies dans la primitive sont ensuite transférées dans le champ d'information de l'utilisateur de l'unité APDU ABRT.

Lorsqu'il a été reçu un message BEGIN contenant une unité APDU de demande de dialogue (AARQ) (c'est-à-dire lorsque le dialogue est dans l'état "initialisation reçue"), l'utilisateur du TC peut abandonner le dialogue pour toute raison de son choix. En pareille situation, l'utilisateur du TC émet une primitive de demande TC-U-ABORT dont le paramètre "raison de l'abandon" est absent ou réglé sur une valeur autre que "nom du contexte d'application non fourni" ou "dialogue refusé". En pareil cas, une unité APDU d'abandon de dialogue (ABRT) dont la valeur d'origine de l'abandon est codée sur "utilisateur du service de dialogue" est générée et fournie en tant que paramètre de données d'utilisateur de la primitive de demande TR-U-ABORT. L'information d'utilisateur fournie (le cas échéant) dans la primitive de demande TC-U-ABORT est codée dans le champ d'information d'utilisateur de l'unité APDU ABRT.

A la réception, pendant l'état "Initialisation envoyée", d'une indication TR-U-ABORT pour laquelle une réponse à une unité APDU de demande de dialogue (AARQ) est attendue, le champ données d'utilisateur de cette primitive doit contenir une unité APDU d'abandon de dialogue (ABRT) dont le champ origine de l'abandon est codé sur la valeur "Utilisateur du service de dialogue". La sous-couche composant CSL débloque toutes les ressources de dialogue et de gestion des composants, et émet une primitive d'indication TC-U-ABORT dont le paramètre raison de l'abandon indique "spécifique à l'utilisateur" et le paramètre Information d'utilisateur provenant (le cas échéant) du champ d'information d'utilisateur de l'unité APDU ABRT. En l'absence d'une unité APDU ABRT codée comme indiqué ci-dessus, la sous-couche CSL envoie à l'utilisateur du TC une indication TC-P-ABORT dont le paramètre de cause d'abandon par le fournisseur est codé sur la valeur "dialogue anormal", et débloque toutes les ressources de dialogue et de gestion des composants.

Lorsque le dialogue est dans l'état "initialisation envoyée", c'est-à-dire qu'un message Begin a été envoyé mais que le message vers l'arrière n'a pas été reçu pour cette transaction, le résultat de la primitive de demande TC/TR-U-ABORT est purement local. Tout message reçu par la suite et qui se rapporte à cette transaction doit être traité conformément aux actions indiquées dans le Tableau 7.

Dans les cas décrits précédemment, l'information accompagnant les primitives (cause de l'abandon par le fournisseur, ou information fournie par l'utilisateur) traverse la sous-couche composant de manière transparente.

La gestion de la notification de situations anormales qui ne peuvent être reliées à un dialogue particulier est pour étude ultérieure.

### 3.2.2.2 Procédures anormales relatives aux opérations

Les situations anormales suivantes sont considérées:

- aucune réaction à l'invocation d'opération de la classe 1 (voir 3.2.1.1.3);
- réception d'un composant mal formé: le type de composant et/ou l'identificateur d'invocation d'opération ne peuvent pas être reconnus (c'est-à-dire que l'automate ne peut être identifié);
- réception d'un composant bien formé en violation des transitions d'états autorisées.

Les actions prises par la sous-couche composant pour signaler les erreurs de la partie composant sont décrites au Tableau 5. Les considérations suivantes ont guidé les choix indiqués dans ce tableau:

- lorsqu'une erreur de protocole a été détectée par l'utilisateur local du TC, cet utilisateur n'est pas avisé par la suite via une indication de rejet de composant (tel qu'indiqué au Tableau 5) puisqu'il a lui-même détecté l'erreur de protocole;
- dans les autres cas (rejet par la sous-couche composant), l'utilisateur local du TC est toujours avisé et peut émettre une primitive de commande de dialogue (voir le mécanisme de rejet décrit ci-dessous);
- lorsqu'un composant est rejeté, l'automate associé est mis au repos;
- le mécanisme de rejet s'applique aussi souvent que possible: même si l'identificateur d'invocation d'opération n'est pas attribué ou pas reconnu (c'est-à-dire qu'aucun automate ne peut être identifié) le mécanisme de rejet doit être déclenché. Le seul cas où le rejet est purement local se présente lorsque le composant à rejeter est lui-même un composant rejet.

Les erreurs de protocole dans la partie composant d'un message TCAP sont signalées en utilisant le composant rejet. Celui-ci est envoyé en réponse à un composant incorrect autre qu'un rejet. Si le composant incorrect est lui-même un composant rejet, ce composant est ignoré et l'utilisateur du TC local est avisé de l'erreur de syntaxe dans le composant rejet reçu.

Lorsqu'un identificateur d'invocation est disponible dans un composant à rejeter, cet identificateur est réutilisé dans le composant rejet.

Les abréviations de type composant sont identifiées dans le Tableau 1.

A l'intérieur d'un message comprenant plusieurs composants, les composants qui suivent un composant mal formé détecté par la sous-couche composant sont éliminés.

**Tableau 5/Q.774 – Actions prises sur les erreurs de protocole de la partie composant**

Local					Distant (Note)	
Type de composant reçu	Type d'erreur	Action locale	Automate de composant	Utilisateur local avisé	Automate de composant	Utilisateur distant avisé
invocation	erreur de syntaxe ou Id corrélé non valide (id d'invocation valide)	initier rejet	S.O	oui <sup>a)</sup>	retour au repos	oui
	erreur de syntaxe (id d'invocation non valide)	initier rejet	S.O	oui <sup>a)</sup>	aucune action	oui

**Tableau 5/Q.774 – Actions prises sur les erreurs de protocole de la partie composant (fin)**

Local					Distant (Note)	
Type de composant reçu	Type d'erreur	Action locale	Automate de composant	Utilisateur local avisé	Automate de composant	Utilisateur distant avisé
retour résultat (dernier/non dernier) ou retour erreur	erreur de syntaxe (id d'invocation non valide)	initier rejet	retour au repos	oui <sup>a)</sup>	S.O	oui
	erreur de syntaxe (id d'invocation non valide)	initier rejet	aucune action	oui <sup>a)</sup>	S.O	oui
retour résultat (dernier/non dernier)	opération classe 2/4	initier rejet	retour au repos	oui <sup>a)</sup>	S.O	oui
retour erreur	opération classe 3/4	initier rejet	retour au repos	oui <sup>a)</sup>	S.O	oui
rejet	erreur de syntaxe	initier rejet local	aucune action	oui <sup>a)</sup>	S.O	non
inconnu	erreur de syntaxe	initier rejet	aucune action/S.O <sup>b)</sup>	oui <sup>a)</sup>	S.O/aucune action <sup>b)</sup>	oui

S.O sans objet.

a) Pour informer l'utilisateur du TC qui peut ainsi émettre une primitive de commande de dialogue afin d'envoyer le composant rejet formulé par la sous-couche composant.

b) Il est impossible de décider s'il existe un ISM, en conséquence, aucune action ne peut être entreprise.

NOTE – Toute action à l'extrémité distante dépend de l'émission par l'utilisateur local d'une primitive de dialogue pour envoyer le composant rejet formulé par le CSL local. Certains utilisateurs peuvent choisir de ne pas émettre la primitive de dialogue dans certains ou dans tous les cas. En pareil cas, il n'y a aucune action à l'extrémité distante.

Le rejet de n'importe quelle partie d'un résultat segmenté est équivalent au rejet du résultat dans sa totalité. L'automate associé revient au repos. Les parties suivantes du même résultat fractionné sont également rejetées (en raison de l'inactivité de l'automate).

Le mécanisme de rejet lorsque la sous-couche composant détecte une situation où un rejet (non local) doit être initialisé (comme dans le Tableau 5), elle construit un composant rejet, le mémorise et informe l'utilisateur local du TC à l'aide de la primitive d'indication TC-L-REJECT. L'utilisateur du TC peut décider:

- a) de continuer le dialogue;
- b) de terminer le dialogue en utilisant le scénario de base;
- c) d'abandonner le dialogue.

Dans les cas a) et b), la première primitive de gestion de dialogue (primitive de demande TC-CONTINUE ou primitive de demande TC-END respectivement) émise par l'utilisateur du TC déclenche la transmission du ou des composants rejet mémorisés construits pour ce dialogue par la sous-couche composant. La sous-couche composant distante reçoit le ou les composants rejet construits

pour ce dialogue, libère, si possible, les automates des composants correspondants (Tableau 5) et informe l'utilisateur du TC du rejet (distant) par la ou les primitives d'information TC-R-REJECT.

Si le rejet émis par la sous-couche composant combiné avec les composants accumulés de l'utilisateur du TC dépasse la longueur de message autorisée, l'utilisateur du TC, informé du composant rejet, doit émettre deux primitives de gestion de dialogue. La sous-couche composant, elle aussi consciente du problème de longueur, enverra tous les composants, sauf le composant rejet, avec la première primitive. Le rejet sera envoyé avec la prochaine primitive de gestion de dialogue et avec les autres composants éventuellement fournis par l'utilisateur du TC.

### 3.2.3 Question de compatibilité

Etant donné que les parties dialogue d'un message TC, telles que définies dans les présentes spécifications, sont facultatives du point de vue de la syntaxe, une application de sous-système TCAP respectant les présentes Recommandations comprendra un message TC-BEGIN sans partie dialogue, c'est-à-dire une implémentation qui respecte les Recommandations relatives au gestionnaire de capacité de 1988, et pourra renvoyer des messages ultérieurs qui soient conformes aux Recommandations plus récentes.

Une implémentation TC se conformant aux Recommandations de 1988, qui reçoit un message BEGIN avec une partie de dialogue le traite comme un message non valable du point de vue syntaxique. L'utilisateur réagira en envoyant un message ABORT accompagné d'un élément d'information sur la cause de l'abandon par le fournisseur P-Abort indiquant "partie de transaction incorrecte". Il appartiendra à l'utilisateur du TC à l'extrémité expéditrice d'interpréter cet abandon. Une interprétation possible sera que l'abandon est dû à une communication dont la mise en œuvre respecte les dispositions du *Livre bleu*; ainsi, l'utilisateur du TC pourra émettre une nouvelle primitive de demande TC-BEGIN sans paramètre lié à la partie de dialogue. Toutefois, ces interprétations et les actions qui en découlent dépassent la portée de ces Recommandations.

Le champ du numéro de version dans l'unité APDU (AARQ) de demande de dialogue et l'unité APDU (AARE) de réponse de dialogue serviront à indiquer les versions de la partie de dialogue qui sont offertes par le sous-système TCAP au nœud qui émet le message. La version décrite dans ces Recommandations est la "version 1". Si la sous-couche composant d'un nœud se conformant à ces Recommandations (notamment la partie de traitement du dialogue facultative) reçoit une unité APDU de demande de dialogue (AARQ) avec un champ de version qui indique que la version 1 n'est pas sur la liste des versions offertes, la sous-couche composant crée une unité APDU de réponse de dialogue (AARE) dont le champ est le suivant:

- version du protocole = version 1;
- nom du contexte d'application = celui qui figure dans l'unité APDU AARQ;
- résultat = rejet (permanent);
- diagnostic de l'origine du résultat = fournisseur du service de dialogue (pas de version de dialogue commune);
- information d'utilisateur = absente.

La sous-couche composant émet une indication de demande TR-U-ABORT avec l'unité APDU AARE en tant qu'information d'abandon par l'utilisateur. Il n'est pas nécessaire que l'utilisateur du TC local soit informé et tout composant reçu est ignoré.

La sous-couche composant au nœud recevant une unité APDU AARE formatée en tant qu'information d'abandon par l'utilisateur dans une primitive d'indication TR-U-ABORT, informe son utilisateur du TC par l'intermédiaire d'une primitive d'indication TC-P-ABORT contenant le paramètre "abandon par le fournisseur" réglé sur "pas de partie de dialogue commune".

Le mécanisme de compatibilité vers l'avant susmentionné vise à garantir l'interfonctionnement avec les versions futures qui résulteront d'une évolution (éventuelle) de la partie dialogue. Dans les cas où seule

la version actuelle de la partie de dialogue (définie dans ces Recommandations) est utilisée, la réception d'une unité APDU AARE accompagnée du champ de version réglé sur n'importe quelle valeur autre que "version 1" sera considérée comme une erreur de syntaxe et les procédures décrites en 3.2.2.1 seront adoptées.

### 3.3 Procédures de la sous-couche transaction

#### 3.3.1 Généralités

Dans le cas d'un dialogue structuré, la sous-couche transaction fournit une connexion de bout en bout entre ses utilisateurs (utilisateurs de la sous-couche transaction). Cette connexion de bout en bout est appelée transaction.

Les procédures de la sous-couche transaction associent chaque message TCAP et, donc, tous les composants contenus ainsi que la partie du dialogue, si elle est présente comme à une transaction donnée.

La sous-couche transaction traite la partie transaction d'un message TCAP (type de message et identificateur de transaction). Les identificateurs de transaction identifient une transaction. Chaque extrémité attribue un identificateur local de transaction; ces identificateurs locaux de transaction sont échangés dans la partie transaction des messages comme indiqué dans la Recommandation Q.773.

La partie composant d'un message TCAP est passée entre la sous-couche composant et la sous-couche transaction en tant que données d'utilisateur dans les primitives de la sous-couche transaction.

Dans le cas d'un dialogue non structuré, aucun identificateur de transaction n'est affecté. La partie composant du message UNIDIRECTIONNEL est reçue sous forme de données d'utilisateur dans la primitive de demande TR-UNI. La partie transaction du message UNIDIRECTIONNEL est mise en forme et le message est transmis.

#### 3.3.2 Mappage entre les primitives de service de la sous-couche transaction et les types de message

La Recommandation Q.771 décrit les services fournis par la sous-couche transaction en définissant l'interface de service entre l'utilisateur de la sous-couche transaction et la sous-couche transaction, et entre la sous-couche transaction et le SCCP. Elle contient aussi des diagrammes de transitions d'états utilisant des primitives de service. Le présent sous-paragraphe offre une description du protocole basée sur les messages. Le mappage entre les primitives de la sous-couche transaction et les unités de données de protocole de la sous-couche transaction est présentée dans le Tableau 6.

**Tableau 6/Q.774 – Mappage entre les primitives de service de transaction et les types de message**

Primitive de service	Type de message
TR-UNI	UNIDIRECTIONNEL
TR-P-ABORT	ABANDON (Note 1)
TR-BEGIN	DÉBUT
TR-CONTINUE	CONTINUATION
TR-U-ABORT	ABANDON (Note 2)
TR-END	TERMINAISON
NOTE 1 – Avec l'élément d'information cause de l'abandon du dialogue par le fournisseur.	
NOTE 2 – Vide, ou avec un élément d'information abandon du dialogue par l'utilisateur.	

### **3.3.3 Procédures normales**

#### **3.3.3.1 Transfert de message sans établissement de transaction**

##### **3.3.3.1.1 Actions à l'extrémité initiatrice**

La primitive de demande TR-UNI est utilisée quand un utilisateur de la sous-couche transaction envoie un message à un autre utilisateur de la sous-couche transaction mais n'a pas besoin d'entrer dans une transaction. Un message unidirectionnel qui n'a pas d'identificateur de transaction est utilisé en pareil cas.

##### **3.3.3.1.2 Actions à l'extrémité destinatrice**

La réception d'un message unidirectionnel provoque l'envoi d'une primitive d'indication TR-UNI à l'utilisateur de la sous-couche transaction. Aucune autre disposition n'est prise par la sous-couche transaction.

#### **3.3.3.2 Transfert de messages à l'intérieur d'une transaction**

##### **3.3.3.2.1 Etablissement de transaction**

Dans la suite, le nœud émetteur du premier message TCAP est appelé "A", et le nœud récepteur est appelé "B".

##### **3.3.3.2.1.1 Actions à l'extrémité initiatrice**

L'utilisateur de la sous-couche transaction au nœud "A" établit une transaction en utilisant une primitive de demande TR-BEGIN, ce qui entraîne l'envoi d'un message Begin du nœud "A" vers le nœud "B".

Le message Begin contient un identificateur de transaction de l'origine. La valeur de l'identificateur de transaction, quand elle est incluse dans n'importe quel message ultérieur partant du nœud "A" en tant qu'identificateur de transaction de l'origine ou dans un message vers le nœud "A" en tant qu'identificateur de transaction de la destination, identifie la transaction au nœud "A".

Une fois que la sous-couche transaction au nœud "A" a envoyé un message Begin, elle ne peut pas envoyer d'autre message à la sous-couche transaction au nœud "B" pour la même transaction jusqu'à ce qu'elle reçoive un message Continue du nœud "B" pour cette transaction.

##### **3.3.3.2.1.2 Actions à l'extrémité destinatrice**

La réception d'un message Begin provoque le passage d'une primitive d'indication TR-BEGIN à l'utilisateur de la sous-couche transaction au nœud B. En réponse à une primitive d'indication TR-BEGIN, l'utilisateur de la sous-couche transaction au nœud "B" décide d'établir, ou non, une transaction. Si l'utilisateur de la sous-couche transaction au nœud "B" veut établir une transaction, il passe une primitive de demande TR-CONTINUE à la sous-couche transaction; autrement, il met fin à la transaction (voir 3.3.3.2.3). Les conditions sont définies par l'utilisateur de la sous-couche transaction.

Le message Begin contient seulement un identificateur de transaction de l'origine. Si, après avoir reçu un message Begin contenant un identificateur de transaction de l'origine donné, la sous-couche transaction reçoit un autre message Begin avec le même identificateur de transaction d'origine, elle ne doit pas considérer cela comme une situation anormale; une seconde transaction est initialisée au nœud "B".

##### **3.3.3.2.2 Continuation de transaction**

Un message Continue est envoyé d'un nœud vers un autre lorsque la primitive de demande TR-CONTINUE est passée de l'utilisateur de la sous-couche transaction à la sous-couche transaction au nœud d'origine.

Un message Continue contient un identificateur de transaction de destination identique – c'est-à-dire de la même longueur d'octet et de la même valeur – à l'identificateur de transaction de l'origine contenu dans le premier message venant du nœud homologue. Chaque nœud attribue son propre identificateur de

transaction de l'origine au moment de l'établissement de la transaction. Les identificateurs de transaction restent constants pendant la durée de la transaction.

Un message Continue contient un identificateur de transaction de l'origine et un identificateur de transaction de la destination. L'identificateur de transaction d'origine dans les messages successifs de Continuation n'est pas examiné.

Lorsqu'un message Continue est reçu, une primitive d'indication TR-CONTINUE est passée à l'utilisateur de la sous-couche transaction de destination.

Une fois que l'utilisateur au nœud "B" a répondu avec une primitive de demande TR-CONTINUE pour établir une transaction, toutes les interactions futures aux deux extrémités entre l'utilisateur de la sous-couche transaction et la sous-couche transaction se font à l'aide de primitives TR-CONTINUE jusqu'à ce que la transaction soit terminée. En termes de message, une fois qu'un message Continue a été envoyé au nœud "B", tous les messages suivants seront des messages Continue jusqu'à ce que la transaction soit terminée.

#### **3.3.3.2.3 Terminaison de transaction**

- méthode de base: un utilisateur de la sous-couche transaction à n'importe quelle extrémité, peut terminer une transaction en passant une primitive de demande TR-END (indiquant une terminaison de base) à la sous-couche transaction. Un message End est envoyé à l'entité homologue qui, à son tour, passe une primitive d'indication TR-END à son utilisateur TR. Le message End contient un identificateur de transaction de destination qui est identique – c'est-à-dire de la même longueur d'octet et de la même valeur – à l'identificateur de transaction reçu dans le premier message reçu du nœud homologue.
- méthode prédéterminée: cette méthode implique que les entités homologues sachent, *a priori* (à un point donné dans le protocole spécifique d'application), que la transaction va être libérée. Donc, dans ce cas, chaque utilisateur TR passe une primitive de demande TR-END (indiquant une terminaison prédéterminée) à sa sous-couche transaction, et aucun message End n'est envoyé.

#### **3.3.3.2.4 Abandon par l'utilisateur de la sous-couche transaction**

Lorsqu'un utilisateur de la sous-couche transaction veut abandonner une transaction, il passe une primitive de demande TR-U-ABORT à la sous-couche transaction, laquelle envoie un message Abort contenant une information fournie par l'utilisateur (cause et diagnostic).

A l'extrémité réception, la sous-couche transaction qui reçoit un message Abort contenant l'information fournie par l'utilisateur, passe cette information, sans l'analyser, à l'utilisateur de la sous-couche transaction dans une primitive d'indication TR-U-ABORT.

#### **3.3.3.2.5 Exemple d'échange de messages**

La Figure 7 montre un exemple d'échange de messages TCAP entre deux utilisateurs de la sous-couche de transaction.

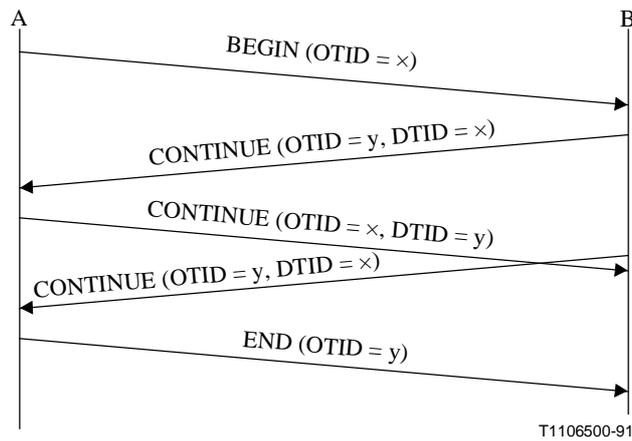


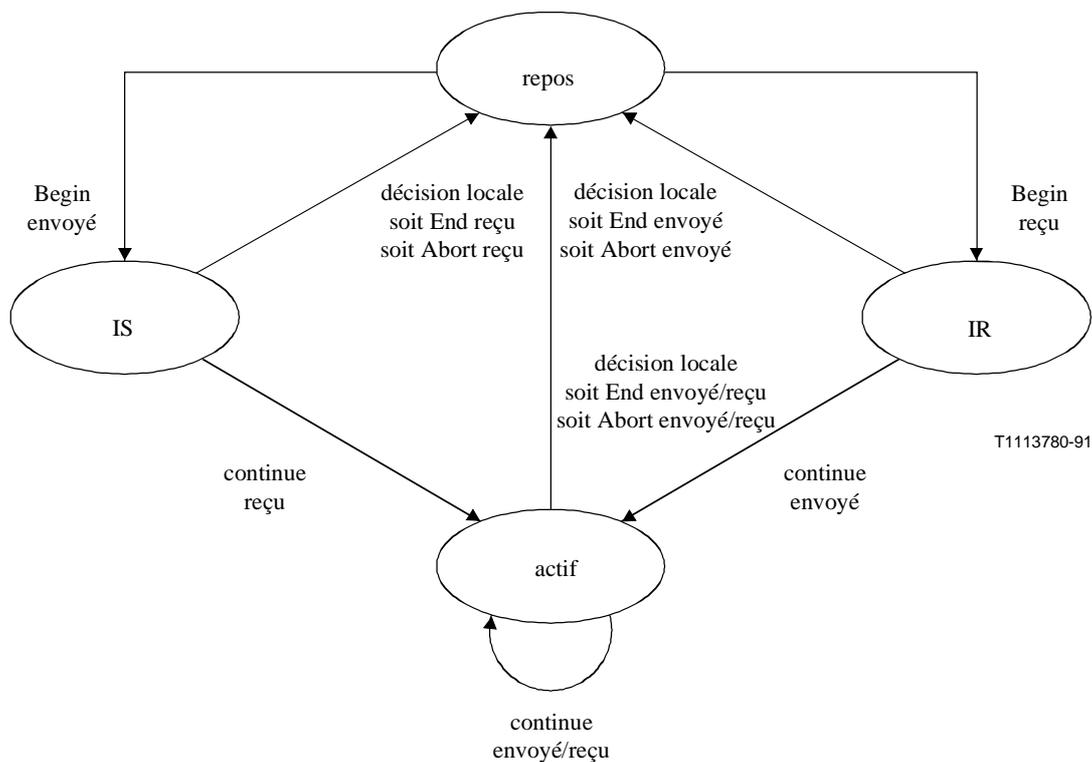
Figure 7/Q.774 – Exemple simple d'échange de messages TCAP

### 3.3.3.2.6 Diagrammes de transitions d'états de la sous-couche transaction

Un automate est associé à une transaction à chaque extrémité de cette transaction. Quatre états de transaction sont utilisés:

- *repos*: aucun automate n'existe.
- *initialisation envoyée (IS, init sent)*: un message Begin a été envoyé: une indication est attendue de l'entité homologue indiquant si la transaction a été établie ou non.
- *initialisation reçue (IR, init received)*: un message Begin a été reçu: une demande est attendue de l'utilisateur de transaction, soit pour continuer la transaction, soit pour la terminer.
- *actif*: la transaction est établie: les messages Continue peuvent être échangés, simultanément, dans les deux directions.

La Figure 8 montre le diagramme de transitions d'états.



NOTE – Décision locale:  
 1) terminaison prédéterminée;  
 2) voir 3.3.4.

**Figure 8/Q.774 – Diagramme de transition d'états de la sous-couche transaction**

### 3.3.4 Procédures anormales relatives à la commande de transaction

Les situations anormales suivantes sont prises en compte par la sous-couche transaction:

- 1) pas de réaction à une transaction (initialisée ou établie);
- 2) réception d'une indication de situation anormale de la couche sous-jacente;
- 3) réception d'un message comportant un identificateur de transaction de destination non attribué ou non déductible, de sorte que le message ne peut être associé à aucune transaction (non déductible signifie que l'information ne peut être ni trouvée, ni reconnue; non attribué signifie que l'identificateur peut être obtenu mais n'a pas été affecté à une transaction);
- 4) réception d'un message comportant un identificateur de transaction de destination correct (le message peut être associé à une transaction) mais dont le type n'est pas compatible avec l'état de la transaction.

Le cas 1 se rapporte aux situations dans lesquelles un nœud est à l'état repos et l'autre ne l'est pas, en raison par exemple, d'une perte de message.

De telles situations sont traitées par un mécanisme local, dépendant de l'implémentation, qui entraîne, localement, l'abandon de la transaction, comme cela est décrit ci-dessous.

Le cas 2 est pour étude ultérieure.

Lorsqu'une erreur est détectée dans la partie transaction d'un message (cas 3 et 4 ci-dessus), la sous-couche transaction doit entreprendre les actions décrites ci-dessous.

L'état de l'identificateur de transaction d'origine doit être vérifié. Les actions sont les suivantes:

- 1) si l'identificateur de transaction d'origine n'est pas déductible, l'extrémité locale, qui a reçu le message, l'ignore et n'entreprend aucune autre action, c'est-à-dire pas d'envoi de message Abort, pas de terminaison de transaction;
- 2) si l'ID de transaction d'origine peut être déduit, les actions sont les suivantes:
  - i) la sous-couche transaction doit construire un message Abort, contenant un élément d'information sur la cause de l'abandon par le fournisseur approprié et l'envoyer à l'extrémité d'origine. L'extrémité d'origine entreprendra alors les actions appropriées pour terminer la transaction si l'identificateur de transaction d'origine est attribué;
  - ii) si l'identificateur de transaction de destination est non déductible ou déductible mais non attribué, la sous-couche transaction n'entreprendra aucune action pour terminer la transaction à son extrémité;
  - iii) si l'identificateur de transaction de destination est déductible et attribué:
    - a) la sous-couche transaction termine la transaction à son extrémité, c'est-à-dire revient au repos;
    - b) la sous-couche transaction informe la sous-couche composant de l'abandon de la transaction via l'abandon de la sous-couche transaction;
    - c) la sous-couche composant doit:
      - libérer tous les identificateurs d'invocation associés à cette transaction,
      - ignorer tous les composants en suspens pour cette transaction,
      - informer l'utilisateur du TC de l'abandon de la transaction.

Enfin, quel que soit l'état des identificateurs de transaction, la totalité du message TCAP erroné doit être ignorée.

Sur réception d'un message Abort, la sous-couche transaction de destination entreprend les actions suivantes:

- si le message Abort contient l'information fournie par l'utilisateur (ou pas d'information), elle informe l'utilisateur de la sous-couche transaction à l'aide de la primitive d'indication TR-U-ABORT;
- si le message Abort contient un élément d'information sur la cause d'abandon par le fournisseur, elle informe l'utilisateur de la sous-couche transaction à l'aide de la primitive d'indication TR-P-ABORT. La question de la notification à l'entité de gestion doit faire l'objet d'un complément d'étude;
- dans les deux cas, tous les messages en suspens pour cette transaction sont éliminés et l'automate de la sous-couche transaction est remis au repos.

**Tableau 7/Q.774 – Actions entreprises sur réception d'un message ayant une partie transaction anormale**

Extrémité locale (qui détecte l'erreur de protocole)						Extrémité distante	
Type de message reçu	ID de transaction d'origine d)	ID de transaction de destination d)	Action	Automate transaction	Utilisateur local avisé	Automate transaction	Utilisateur avisé
UNIDIRECTIONNEL	–	–	ignoré	– <sup>c)</sup>	non	– <sup>c)</sup>	non
DÉBUT	non déductible	–	ignoré	S.O <sup>b)</sup>	non	S.O <sup>b)</sup>	non
	déductible	–	abandon	S.O <sup>b)</sup>	non	retour au repos <sup>a)</sup>	oui <sup>a)</sup>
CONTINUATION	non déductible	–	ignoré	S.O <sup>b)</sup>	non	S.O <sup>b)</sup>	non
	déductible	non déductible non attribué	abandon	S.O <sup>b)</sup>	non	retour au repos <sup>a)</sup>	oui <sup>a)</sup>
	déductible	attribué	abandon	retour au repos	oui	retour au repos <sup>a)</sup>	oui <sup>a)</sup>
TERMINAISON/ ABANDON	–	non déductible non attribué	ignoré	S.O <sup>b)</sup>	non	S.O <sup>b)</sup>	non
	–	attribué	ignoré	retour au repos	oui	S.O <sup>b)</sup>	non
INCONNU	non déductible	–	ignoré	S.O <sup>b)</sup>	non	S.O <sup>b)</sup>	non
	déductible	non déductible non attribué	abandon	S.O <sup>b)</sup>	non	retour au repos <sup>a)</sup>	oui <sup>a)</sup>
	déductible	attribué	abandon	retour au repos	oui	retour au repos <sup>a)</sup>	oui <sup>a)</sup>

S.O Le passage à l'état repos est sans objet [voir b) ci-après].  
Deductible Le fait qu'une anomalie rende un identificateur de transaction non déductible dépend des implémentations.  
Attribué Déductible et attribué  
Non attribué Déductible et non attribué  
Abandon Envoi d'un message Abort

<sup>a)</sup> Si l'identificateur de transaction est attribué à cette extrémité, sinon la transition d'état n'est pas applicable, et l'utilisateur n'est pas informé.  
<sup>b)</sup> L'expression S.O est utilisée dans les cas où la procédure normale de retour au repos aux deux extrémités, suite à l'apparition d'une situation anormale, n'est pas applicable parce qu'il n'est pas possible de déterminer le ou les identificateurs de transaction, et donc, d'établir une relation entre le message endommagé et une transaction particulière, à l'une ou l'autre des extrémités (locale et distante).  
<sup>c)</sup> Le message unidirectionnel ne concerne pas une transaction explicite. Il est donc sans effet sur l'automate de la sous-couche transaction.  
<sup>d)</sup> La possibilité d'obtenir les identificateurs de transaction dépend de l'implémentation.

## ANNEXE A

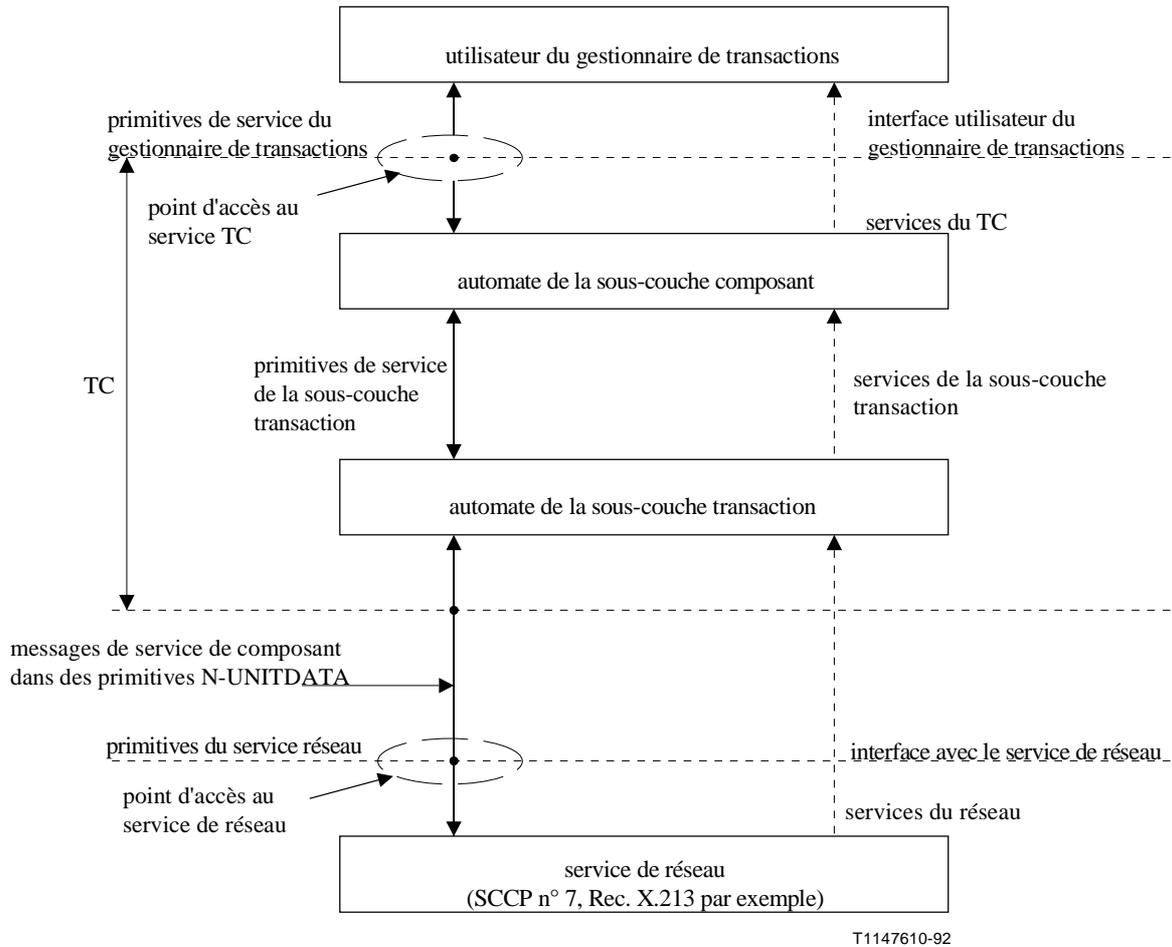
### Diagrammes SDL du gestionnaire de transactions

#### A.1 Généralités

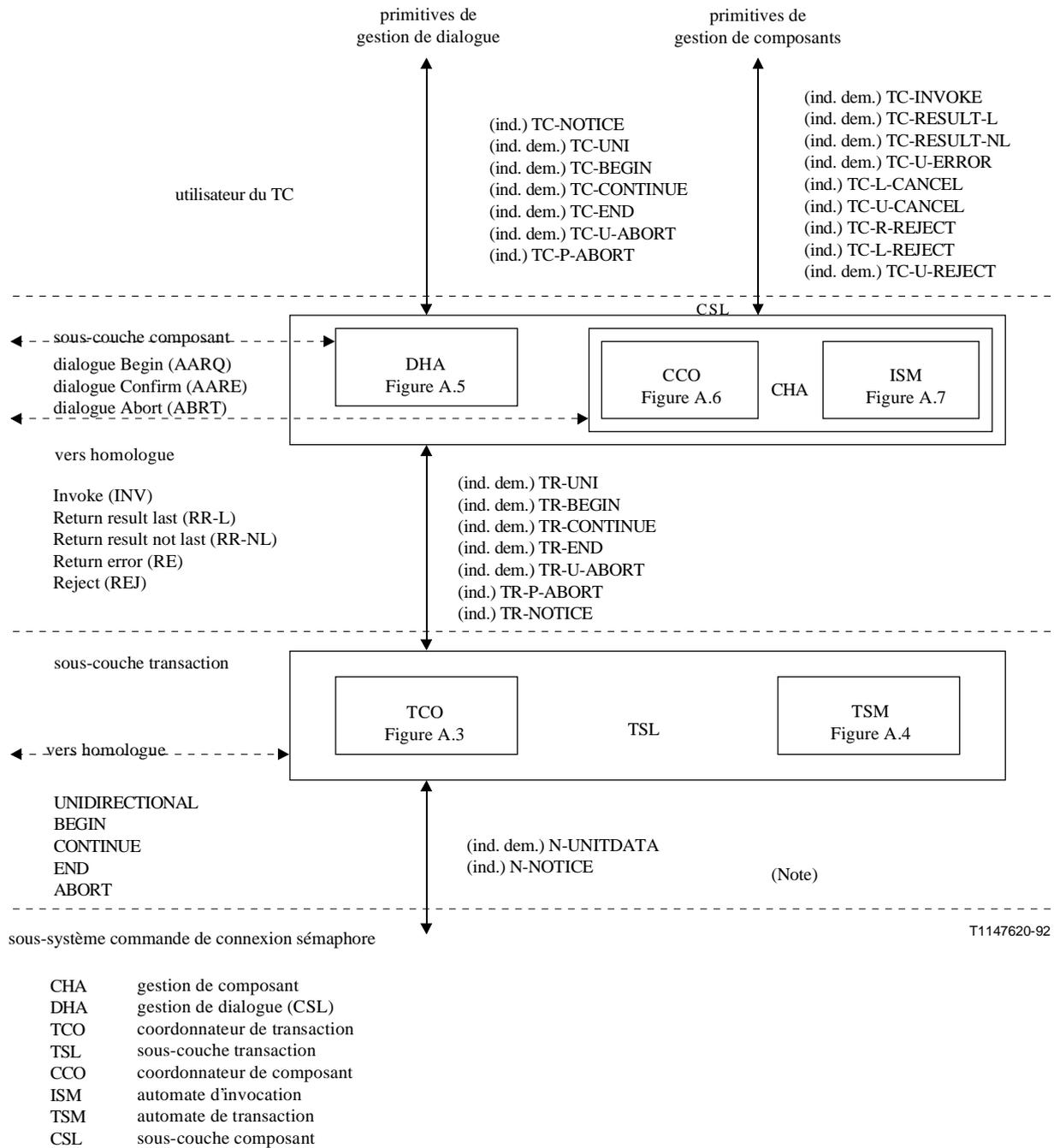
La présente annexe contient la description des procédures du gestionnaire de transactions, décrites dans la présente Recommandation à l'aide de diagrammes SDL conformément aux spécifications du CCITT et au langage de description SDL. En vue de faciliter la description fonctionnelle, ainsi que la

compréhension du comportement du système de signalisation, le sous-système application pour la gestion des transactions (TCAP, *transaction capabilities application*) est divisé en une sous-couche composant et une sous-couche transaction (Figures A.1). La sous-couche transaction est divisée en deux blocs fonctionnels: un coordonnateur de transaction (TCO, *transaction coordinator*) et un ou des automates de transaction (TSM, *transaction state machine*). La sous-couche composant est elle-même divisée en un bloc de gestion des composants (CHA, *component handling*) et un bloc de gestion du dialogue (DHA, *dialogue handling*) (Figure A.2).

Les SDL sont présentés selon cette découpe fonctionnelle qui est utilisée seulement pour faciliter la compréhension et ne doit pas être nécessairement adoptée dans une application pratique du TCAP. Les blocs fonctionnels et les primitives de services qui leur sont associées sont représentés à la Figure A.2.

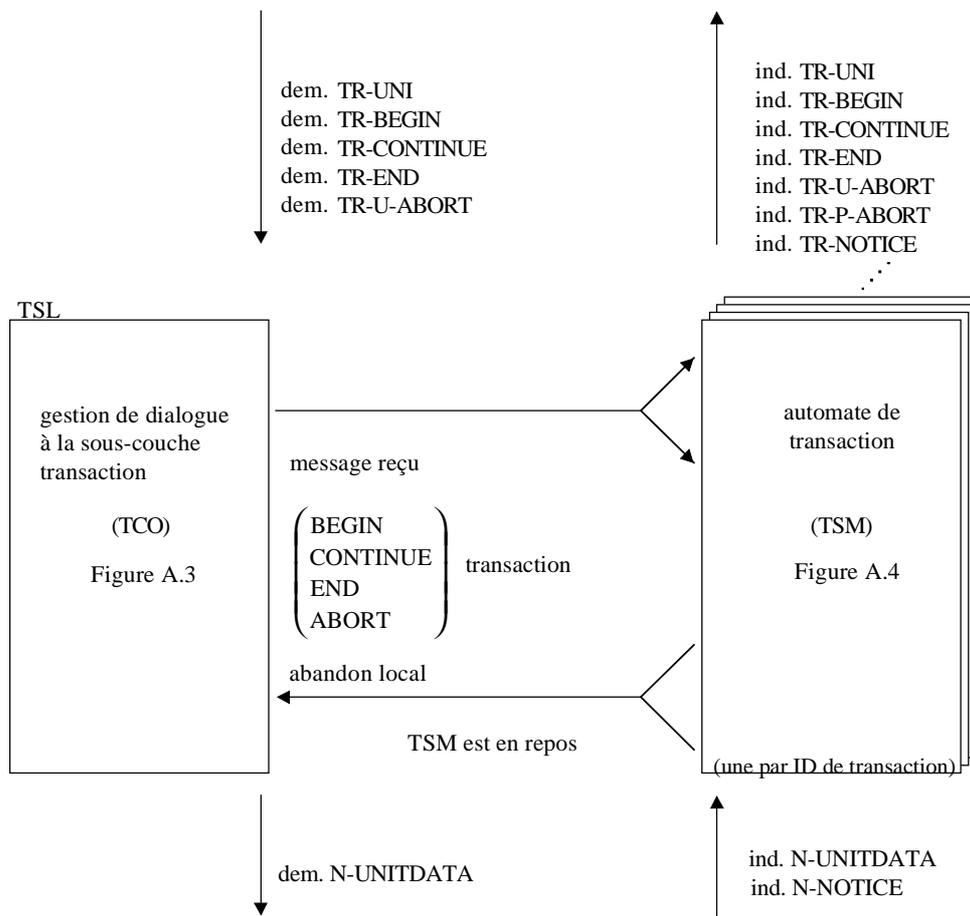


**Figure A.1/Q.774 – Interfaces des sous-couches transaction et composant (automates) et primitives de services**



NOTE – Les autres primitives du service de réseau sont pour étude ultérieure (voir les Recommandations de la série Q.700).

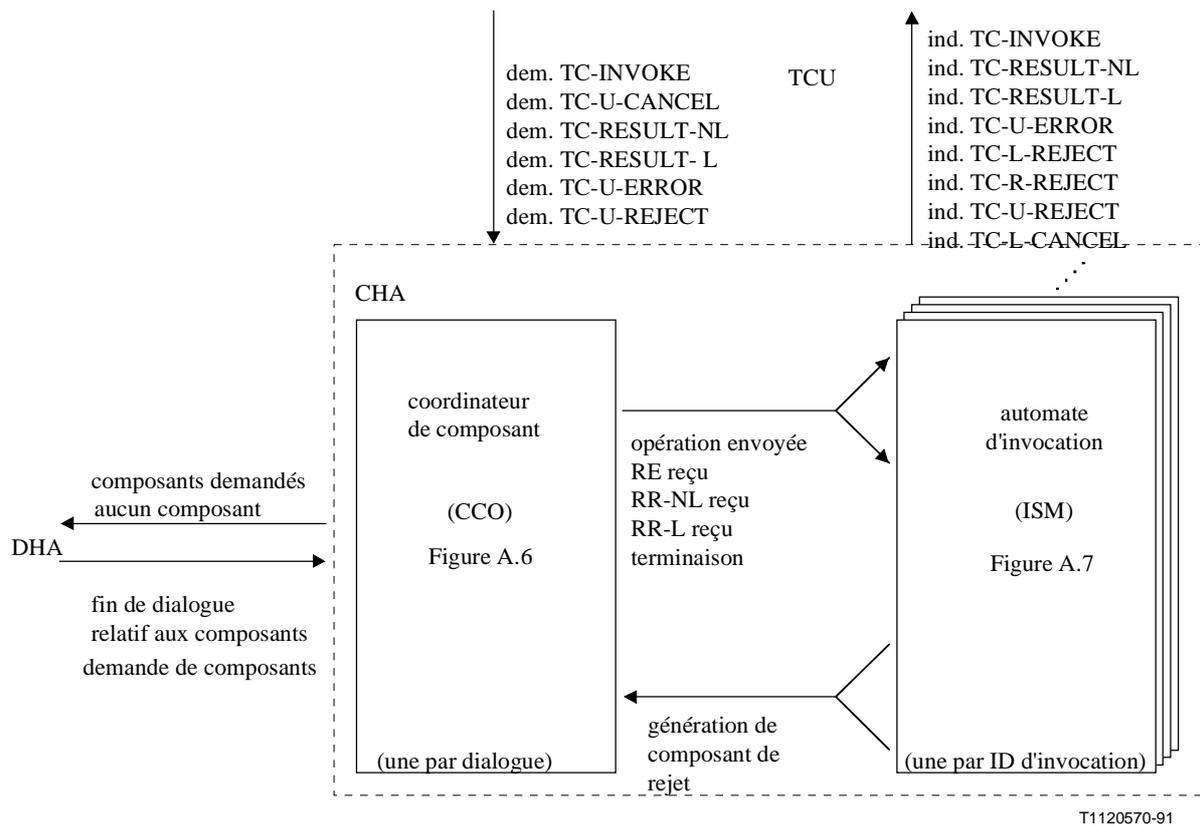
**Figure A.2a/Q.774 – Diagramme des blocs fonctionnels du TC**



T1132080-91

NOTE – Les fonctions de la sous-couche transaction sont divisées en deux blocs fonctionnels: le coordonnateur de transaction (TCO) et l'automate de transaction (TSM). Le TCO gère le contrôle de syntaxe des messages TC entrants et les interactions avec les différents TSM. Pour chaque transaction, il existe un TSM qui donne l'information d'état, envoie des primitives à l'utilisateur du TR, assemble et envoie les messages TC. Les messages UNIDIRECTIONNEL et ABANDON qui ne sont corrélés à aucun TSM local sont assemblés et envoyés par le TCO.

**Figure A.2b/Q.774 – Diagramme des blocs fonctionnels de la sous-couche transaction**

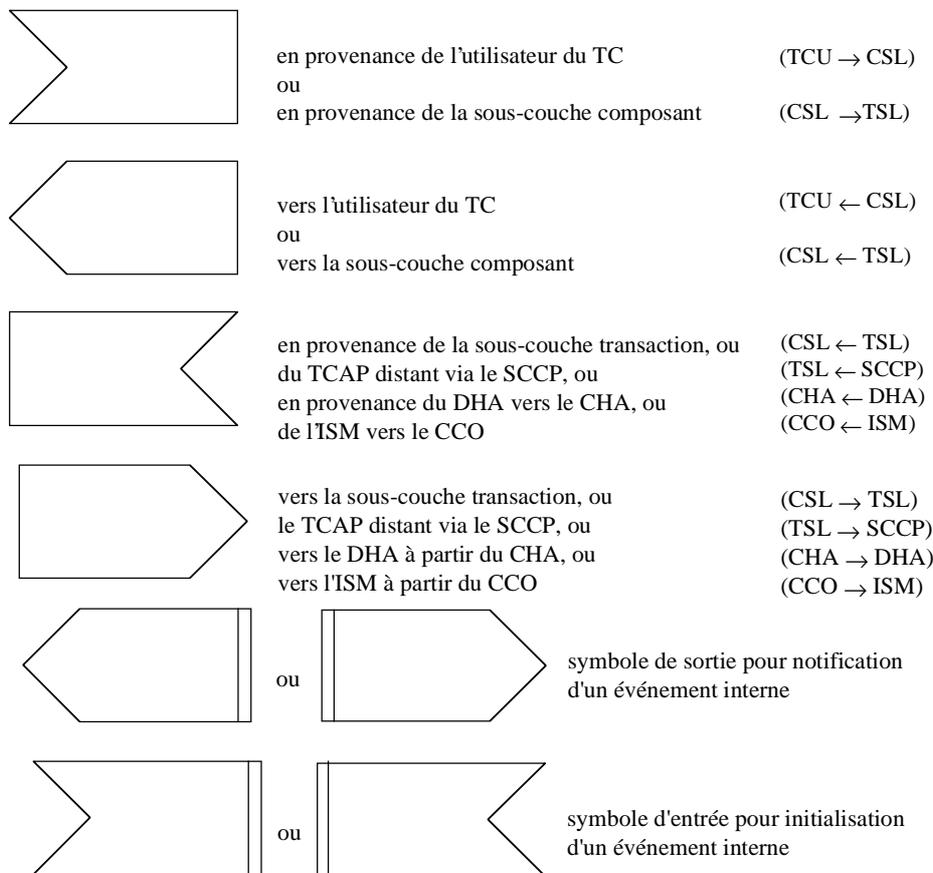


T1120570-91

**Figure A.2c/Q.774 – Diagramme des blocs fonctionnels de la gestion de transactions**

## A.2 Conventions utilisées

Afin d'indiquer la direction de chaque interaction, les symboles suivants sont utilisés:



T1147600-92

## A.3 Création dynamique de processus

Les ensembles communs coordonnateur de transactions (TCO, *transaction coordinator*) et identificateur sont les seuls processus créés à la mise en service du système. Les processus automate de transaction (TSM, *transaction state machine*), gestion de dialogue (DHA, *dialogue handling*) et coordonnateur de composants (CCO, *component coordinator*) doivent être créés dynamiquement et cesser d'exister lorsque le dialogue ou la transaction auquel ou à laquelle ils sont associés prend fin. L'automate d'invocation (ISM, *invocation state machine*) doit cesser d'exister dès que les ressources associées (temporisateurs, identificateurs d'invocation) sont mises à disposition.

NOTE – La version de 1993 de la Recommandation Q.774 indiquait que ces processus reviennent toujours à l'état de repos après avoir exécuté toutes les tâches associées à un dialogue ou une opération. Le retour systématique à l'état de repos est toutefois impossible car il signifierait qu'il existe un processus unique de chaque type qui gère séquentiellement tous les dialogues (ou toutes les opérations à distance). Dans la présente révision de la Recommandation Q.774, l'état de repos a été remplacé partout où il y avait lieu par le symbole d'arrêt. En outre, dans certaines des situations présentées dans les diagrammes SDL de 1993, le coordonnateur de composants (CCO) ne pouvait pas savoir que le dialogue auquel il était associé avait pris fin (ou que la procédure d'établissement du dialogue n'avait pas abouti). Dans la présente révision, on a modifié la gestion de dialogue (DHA) de manière qu'un signal de "fin de dialogue" soit systématiquement envoyé au bloc de gestion de composant (CHA) (CCO) avant l'appel de la procédure d'"Identification de libre dialogue".

Pour chaque type de processus dans un système TC, le Tableau A.1 indique quel type de processus crée des instances de ce processus et dans quelles conditions.

**Tableau A.1/Q.774 – Aperçu général de la création de processus**

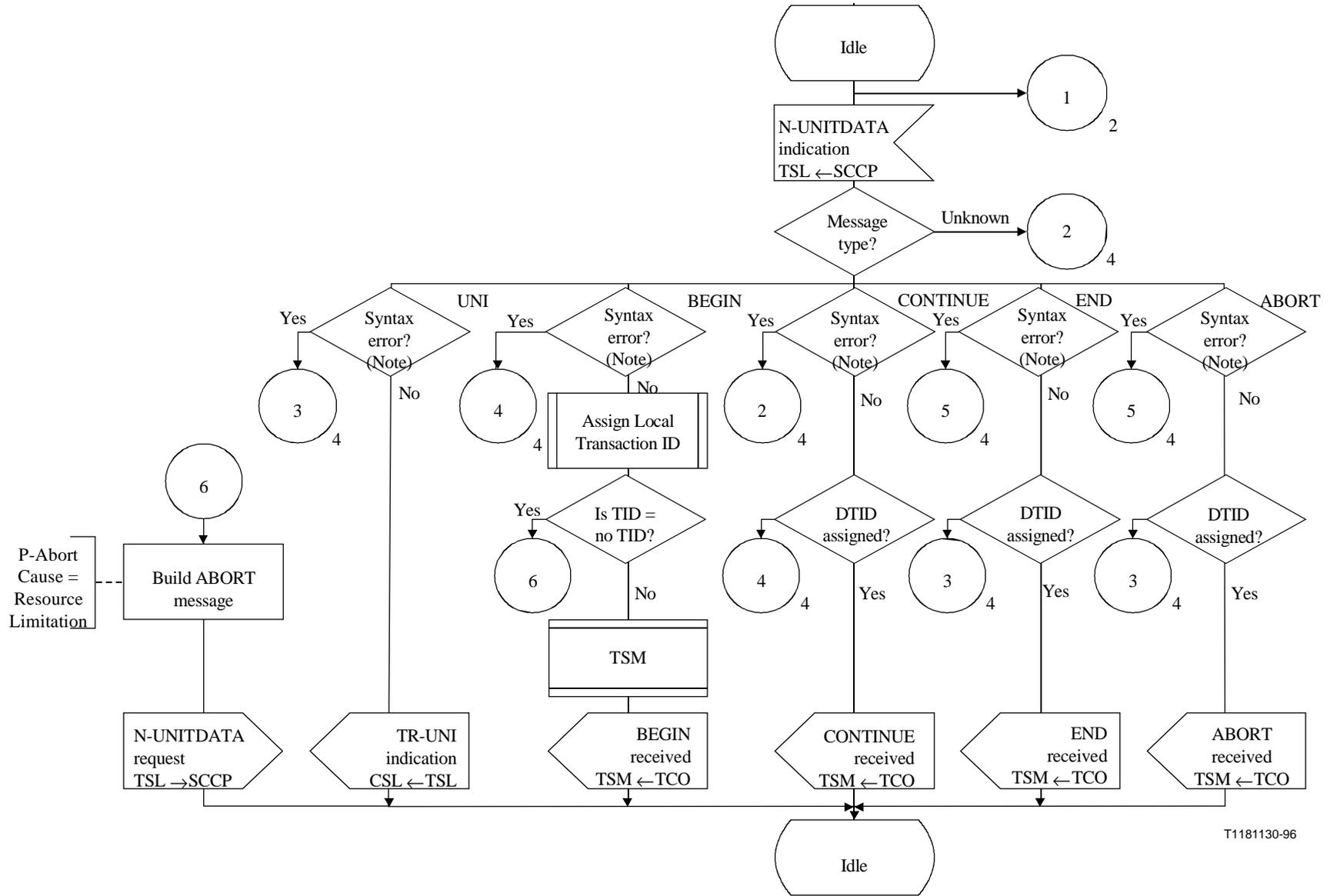
<b>Processus</b>	<b>Créé par</b>	<b>Evénements déclencheurs</b>
TCO	gestion du système	mise en service du système
TSM	TCO	chaque fois que le TCO reçoit un message BEGIN valable du SCCP ou une demande TR-BEGIN
DHA	TSM	lorsque le TSM reçoit un signal de "réception du message Begin" en provenance du TCO
	utilisateur du TC	chaque fois qu'un utilisateur du TC a besoin d'initialiser un nouveau dialogue
	TCO	chaque fois que le TCO reçoit un message UNI valable en provenance du SCCP
CCO	DHA	la création du bloc DHA crée à son tour un CCO
ISM	CCO	lorsque le bloc DHA indique qu'une opération a été lancée

Un processus n'est pas explicitement arrêté par un autre. Il s'arrête à la réception d'un signal ou après avoir exécuté certaines actions spécifiques. Pour chaque type de processus contenu dans un système TC, le Tableau A.2 indique les conditions qui conduisent une instance à s'arrêter.

**Tableau A.2/Q.774 – Aperçu général des conditions d'arrêt des processus**

<b>Processus</b>	<b>Evénements déclencheurs</b>
TCO	mise hors service du système
TSM	réception d'un signal (END, ABORT, ...) en provenance du TCO, indiquant qu'il faut mettre fin à la transaction
DHA	après émission de l'identificateur de dialogue à la clôture d'une session de dialogue
CCO	réception du signal de "Fin de dialogue" envoyé par le bloc DHA avant que celui-ci ne s'arrête
ISM	à la fin de l'opération ou à la réception d'un signal de "Fin" en provenance du CCO

On part du principe qu'une fonction locale "utilisateur du TC" crée une instance DHA avant le lancement d'un dialogue.



T1181130-96

NOTE – Checks for correctly formatted TR-portion information elements for this message type.

Figure A.3/Q.774 (feuille 1 de 4) – Coordonnateur de transactions

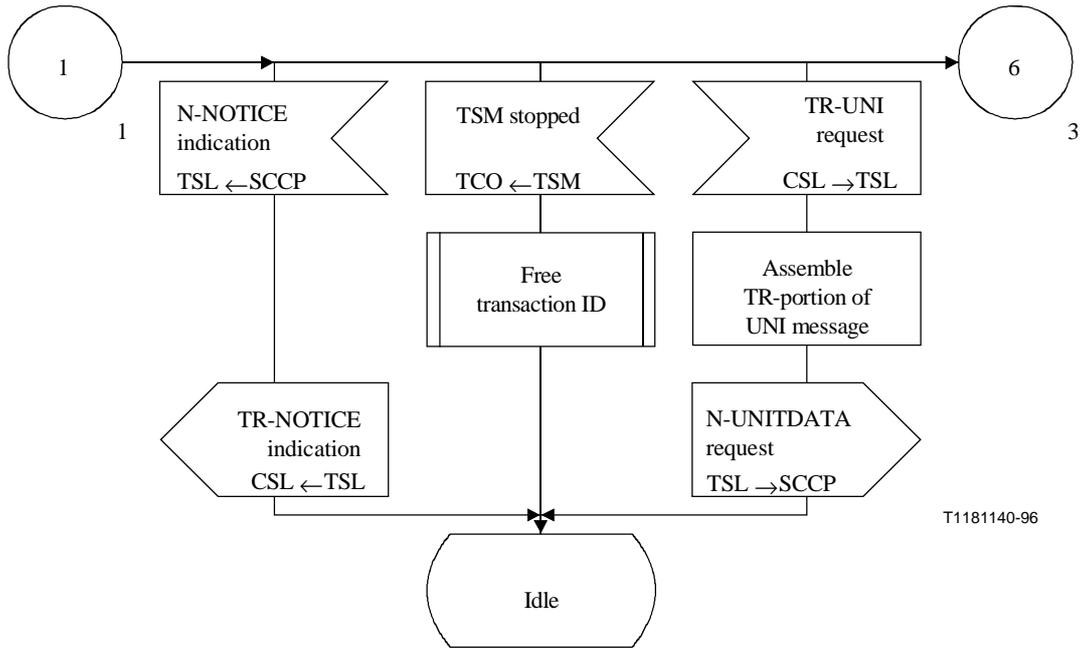


Figure A.3/Q.774 (feuillet 2 de 4) – Coordonnateur de transactions

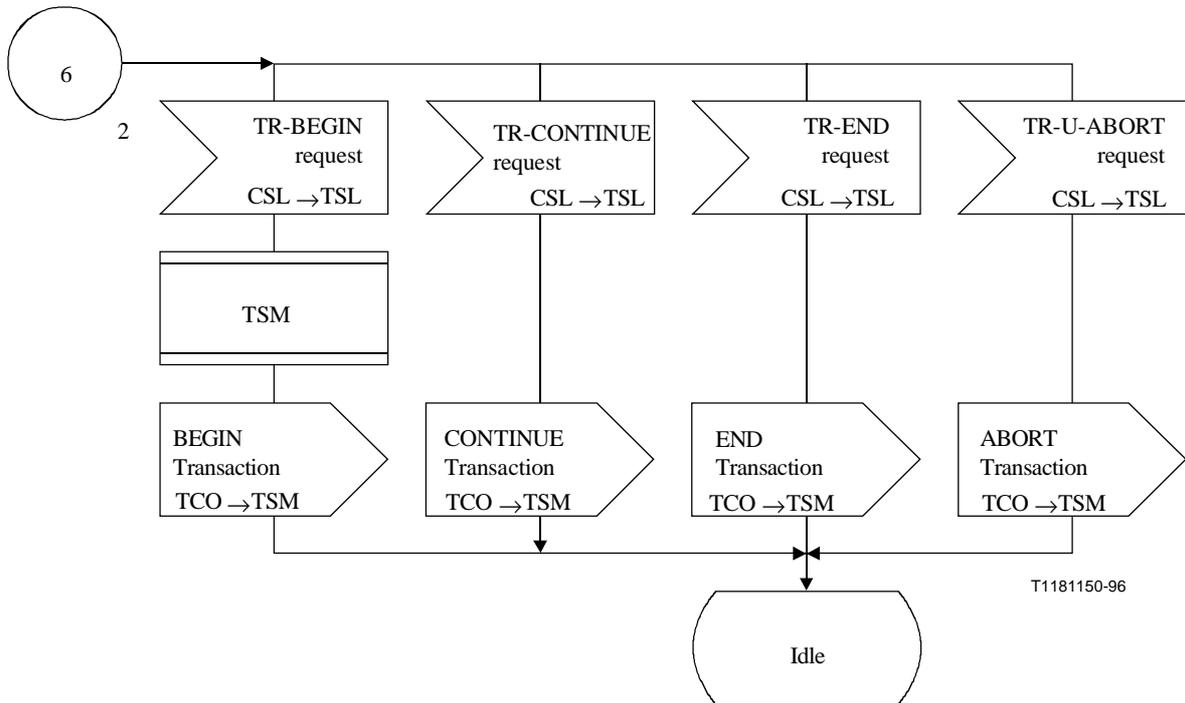
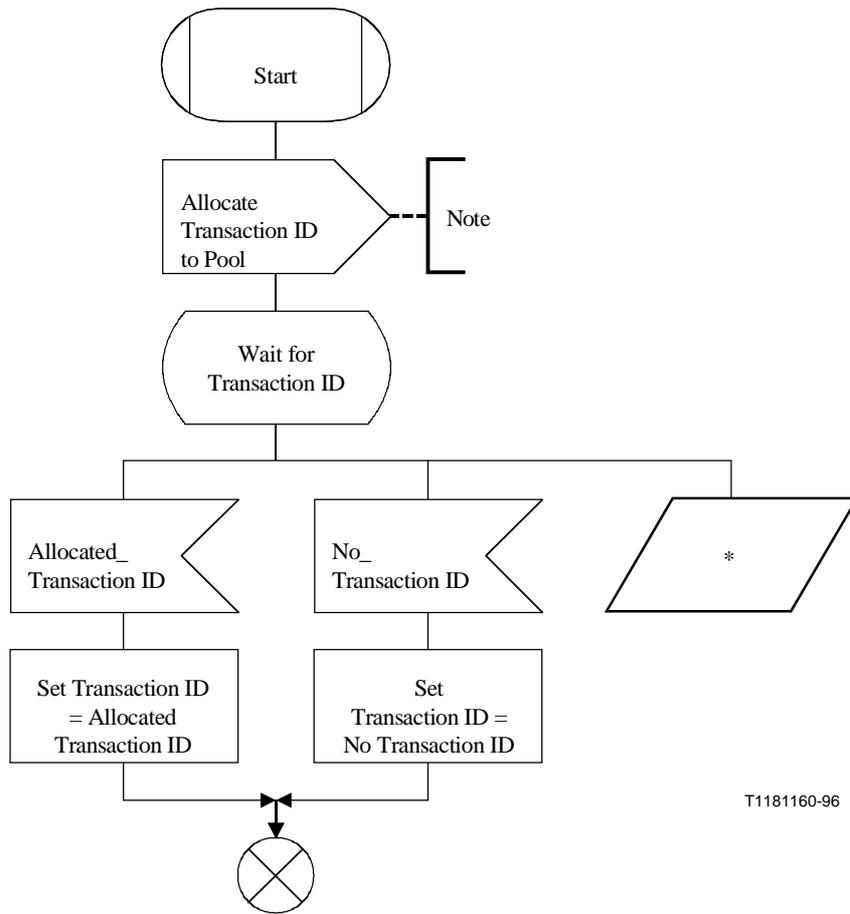


Figure A.3/Q.774 (feuillet 3 de 4) – Coordonnateur de transactions

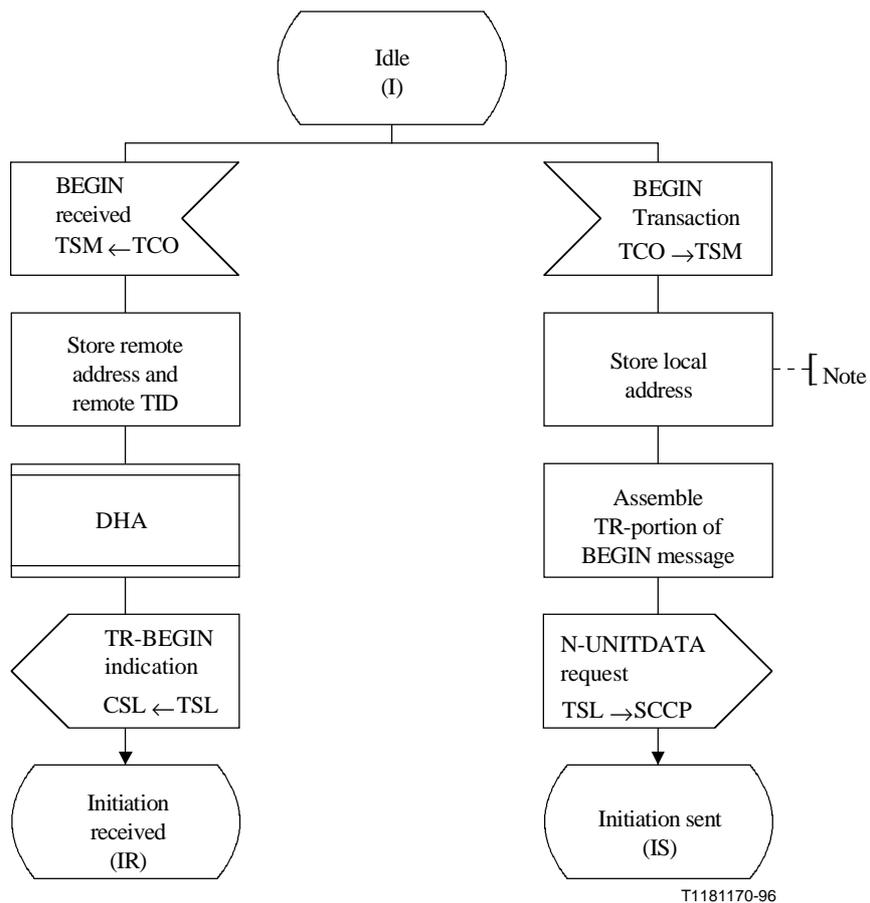




T1181160-96

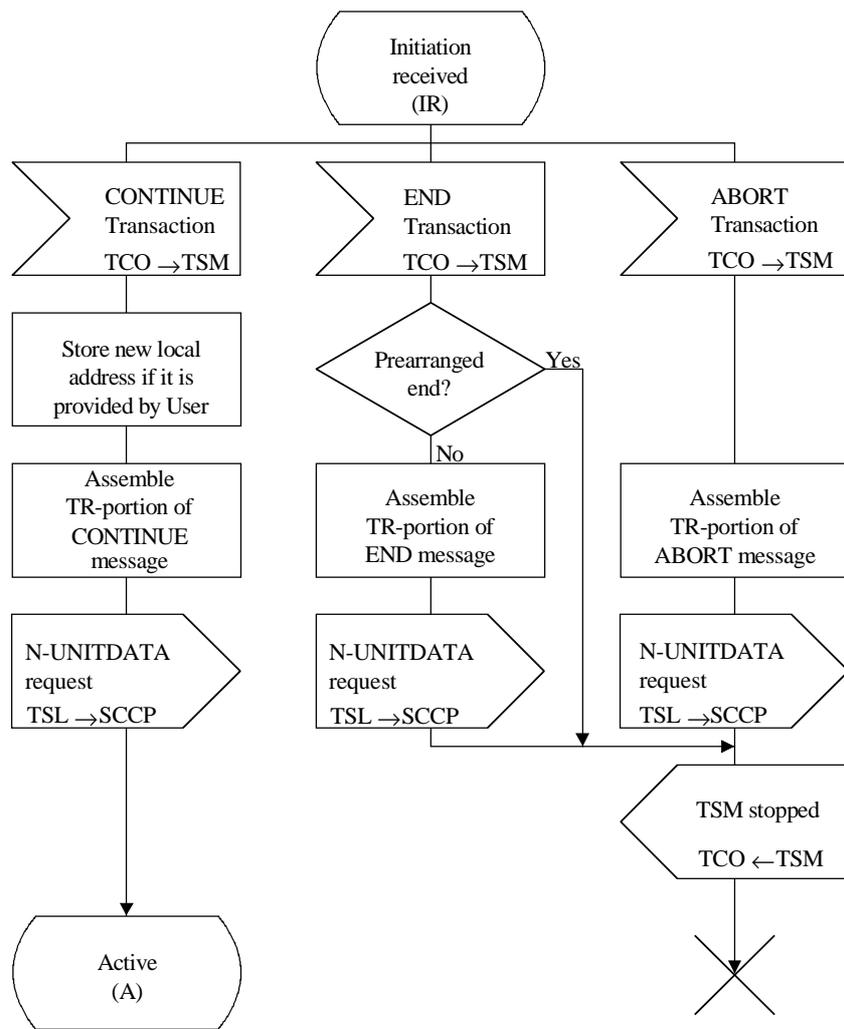
NOTE – The pool realization is implementation-dependent.

**Figure A.3 bis/Q.774 – Procédure ASSIGN TRANSACTION ID**



NOTE – This may be provided by TC-user or be implicitly associated with the access point at which the N-UNITDATA primitive is issued.

**Figure A.4/Q.774 (feuillet 1 de 5) – Automate de transaction**



T1181180-96

**Figure A.4/Q.774 (feuille 2 de 5) – Automate de transaction**

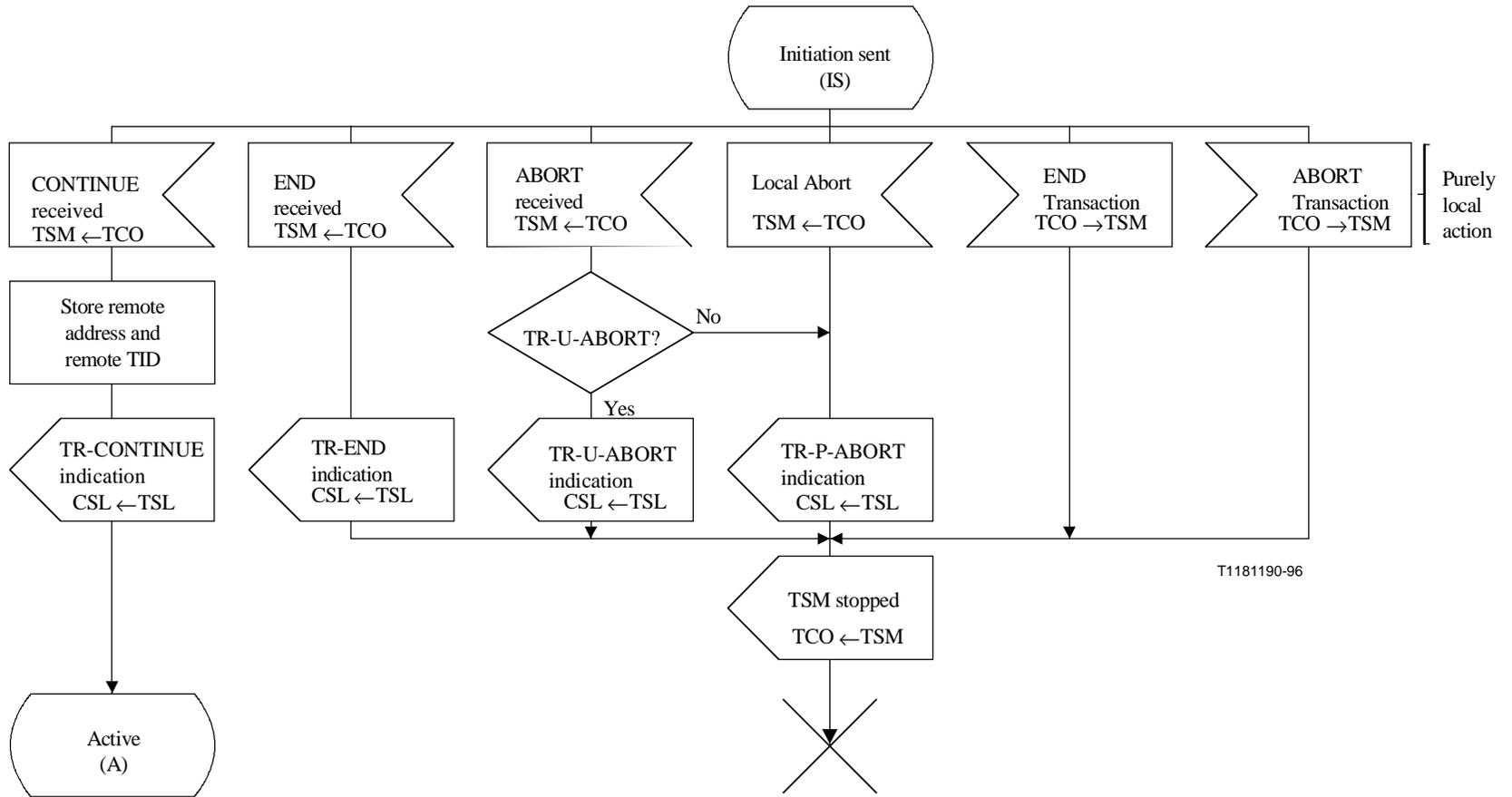


Figure A.4/Q.774 (feuillet 3 de 5) – Automate de transaction

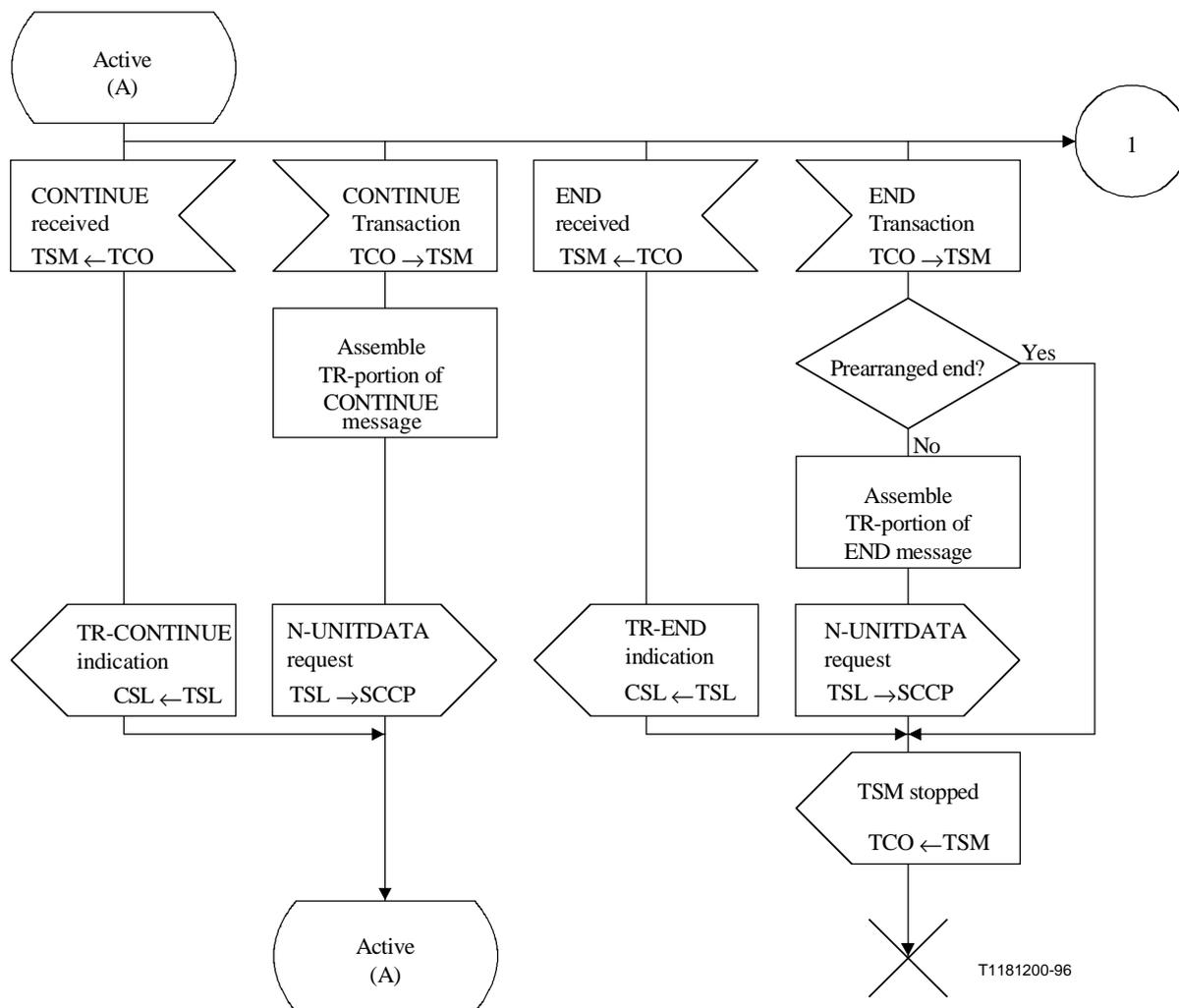
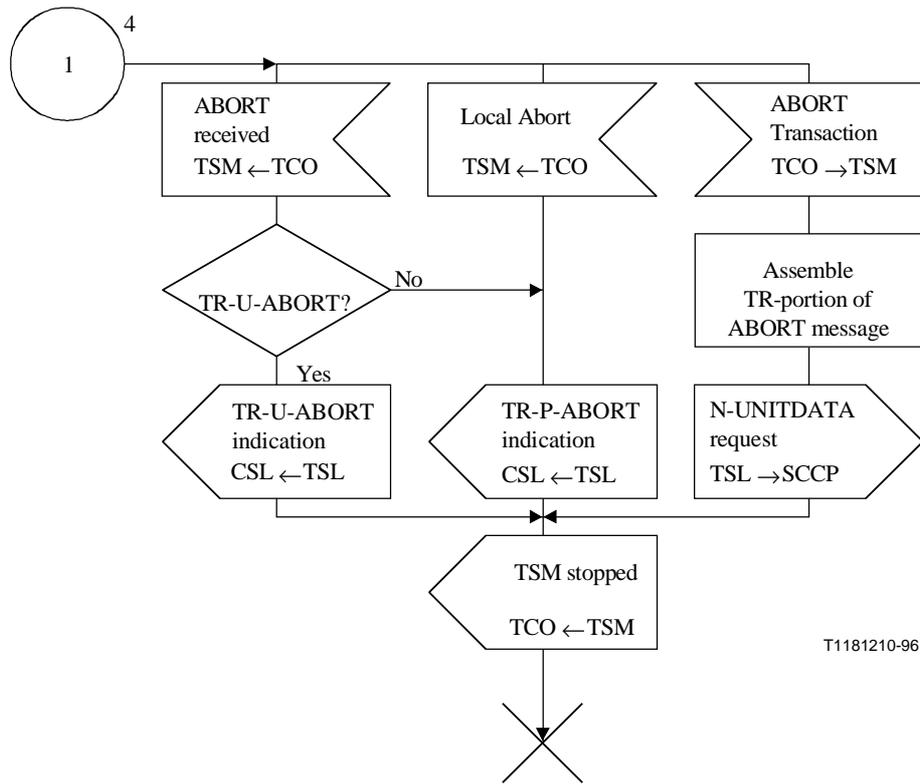


Figure A.4/Q.774 (feuille 4 de 5) – Automate de transaction



T1181210-96

Figure A.4/Q.774 (feuillet 5 de 5) – Automate de transaction

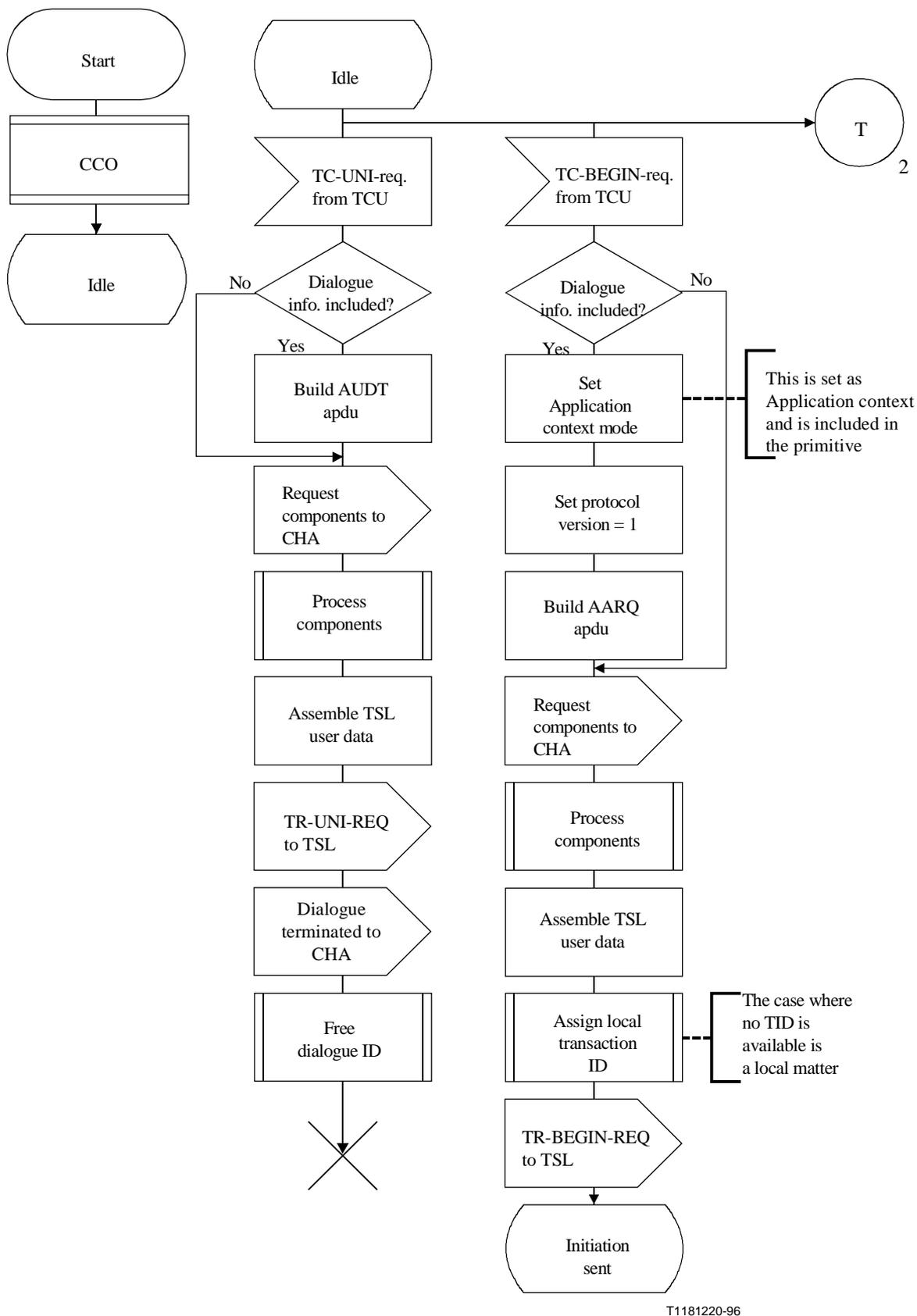


Figure A.5/Q.774 (feuille 1 de 11) – Gestion du dialogue dans la sous-couche composant

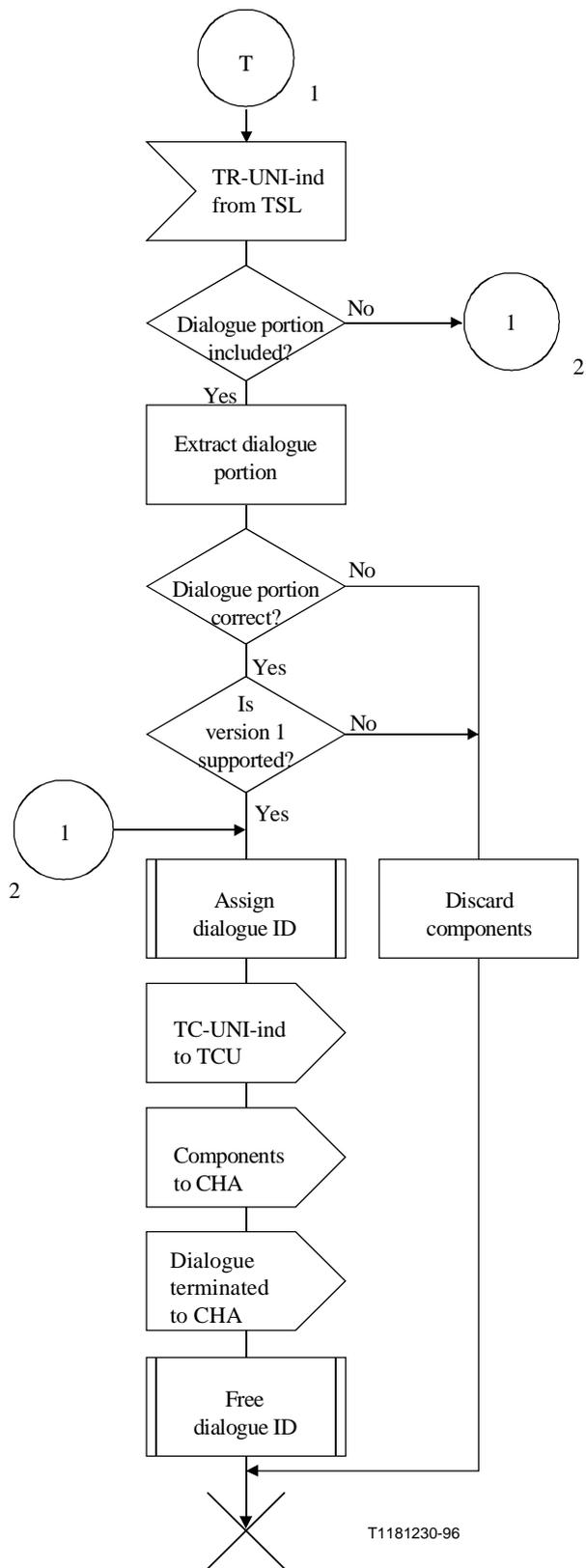


Figure A.5/Q.774 (feuillet 2 de 11) – Gestion du dialogue dans la sous-couche composant

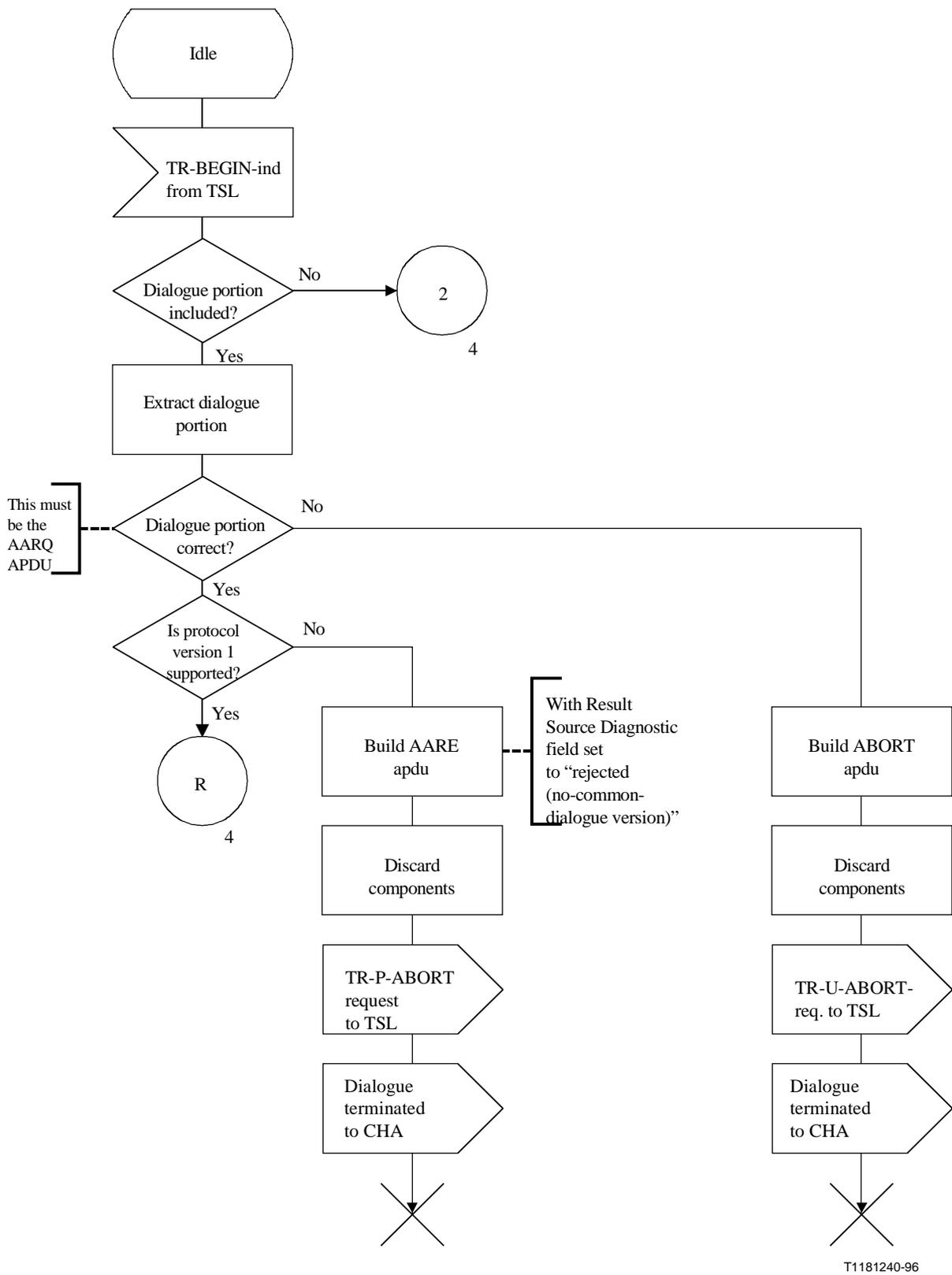
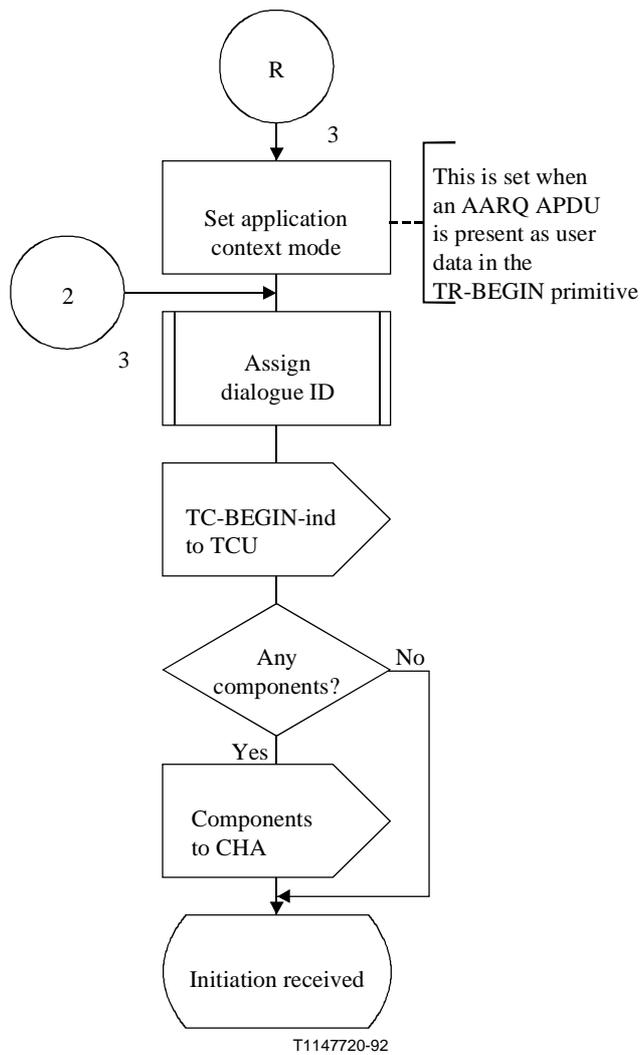


Figure A.5/Q.774 (feuille 3 de 11) – Gestion du dialogue dans la sous-couche composant



**Figure A.5/Q.774 (feuille 4 de 11) – Gestion du dialogue dans la sous-couche composant**

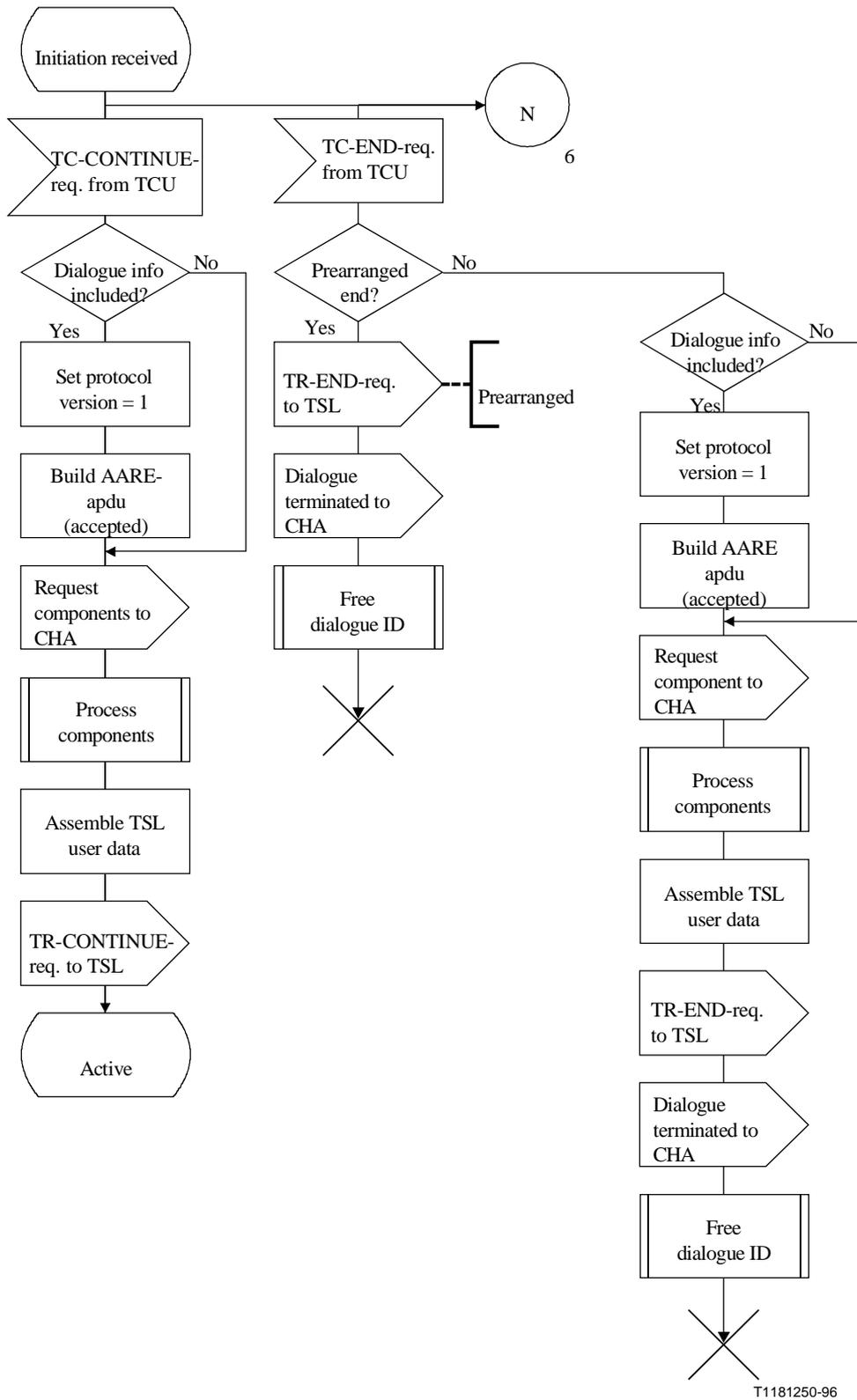
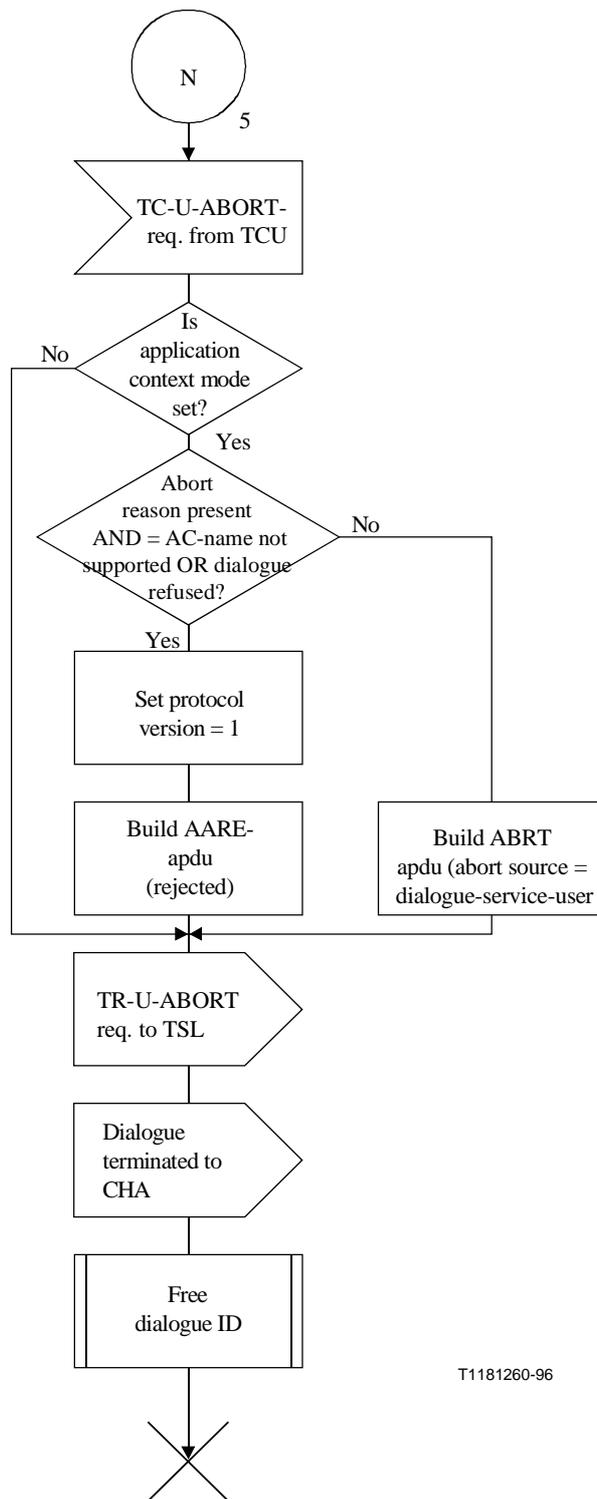
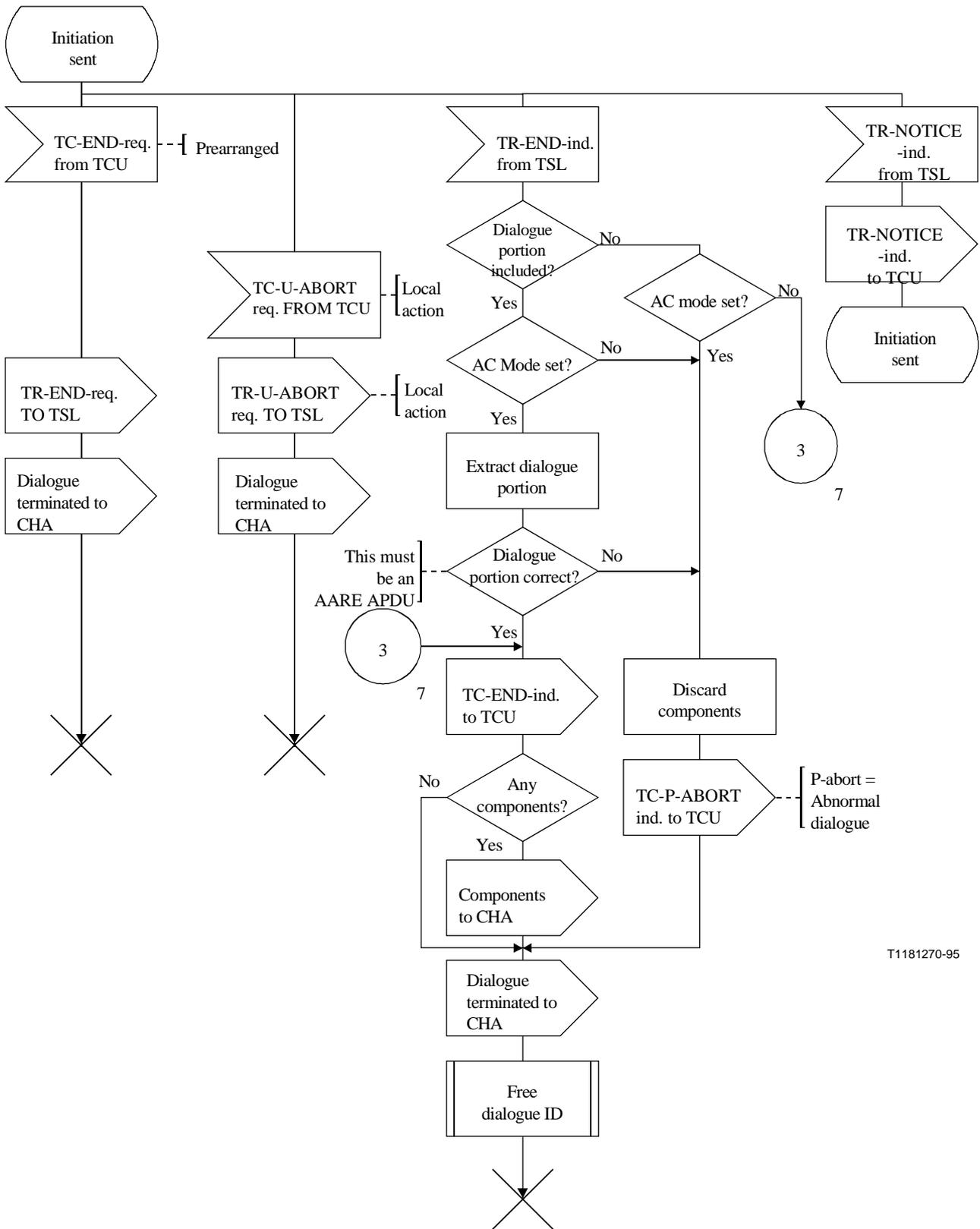


Figure A.5/Q.774 (feuille 5 de 11) – Gestion du dialogue dans la sous-couche composant



T1181260-96

Figure A.5/Q.774 (feuillet 6 de 11) – Gestion du dialogue dans la sous-couche composant



T1181270-95

Figure A.5/Q.774 (feuille 7 de 11) – Gestion du dialogue dans la sous-couche composant

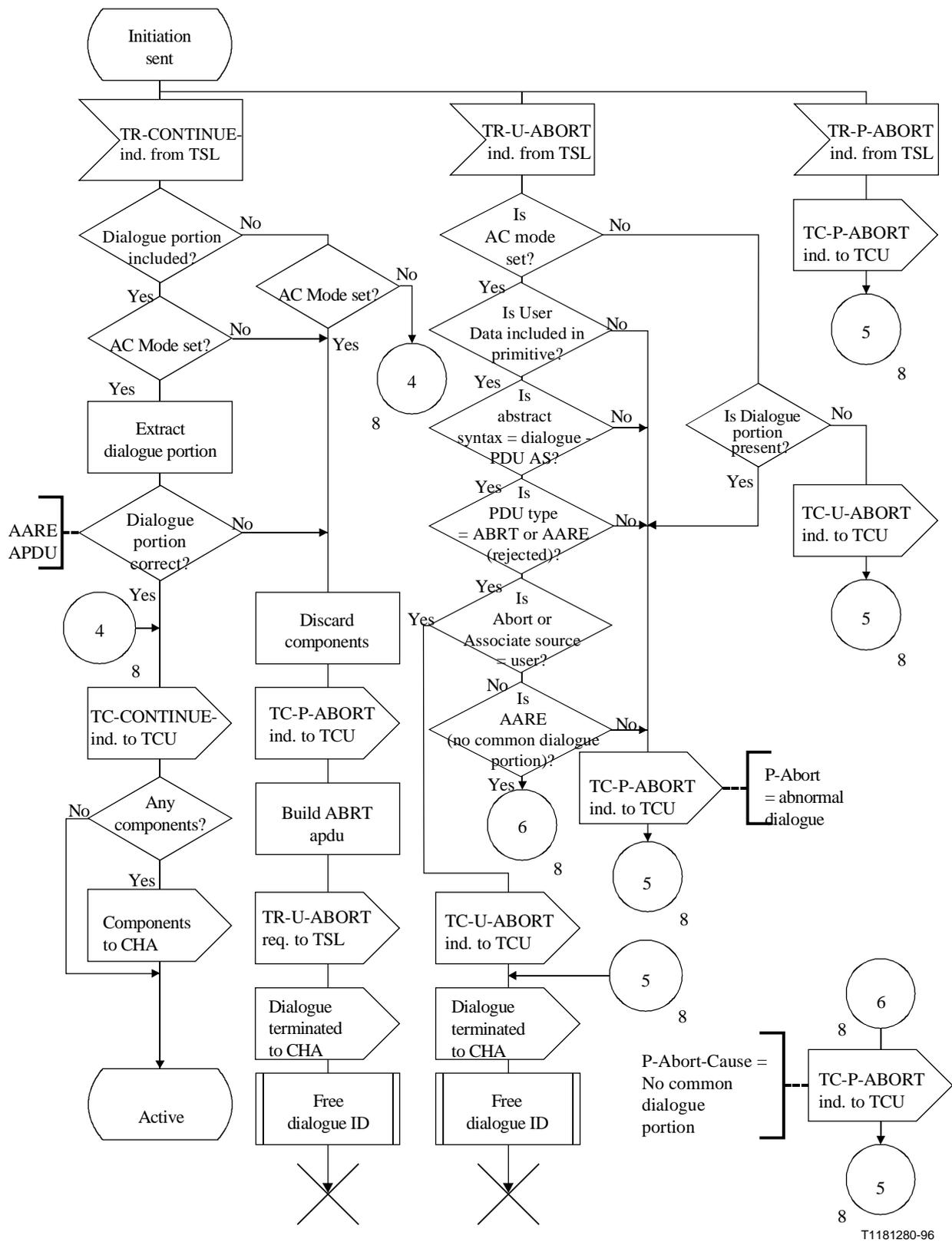


Figure A.5/Q.774 (feuille 8 de 11) – Gestion du dialogue dans la sous-couche composant

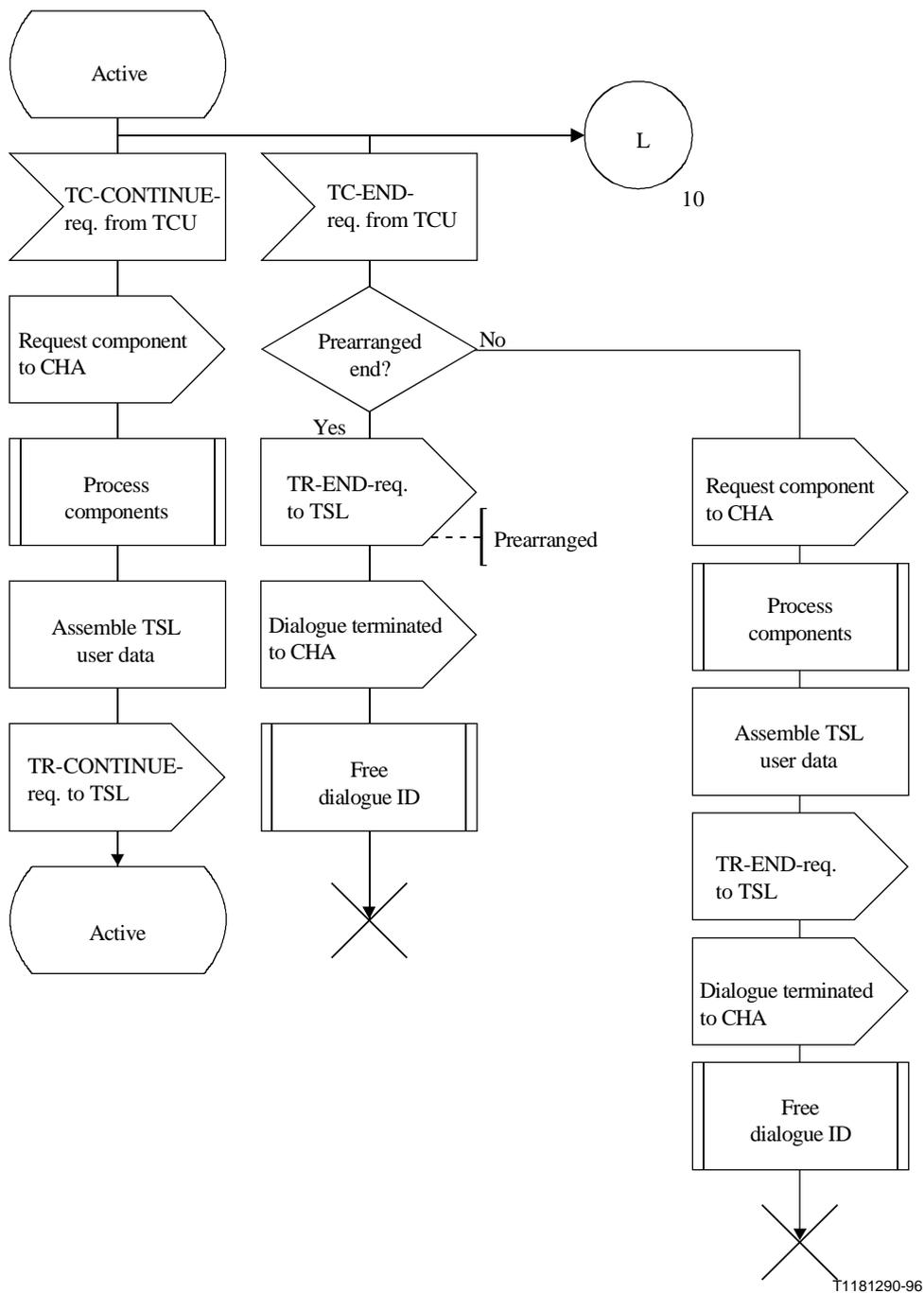
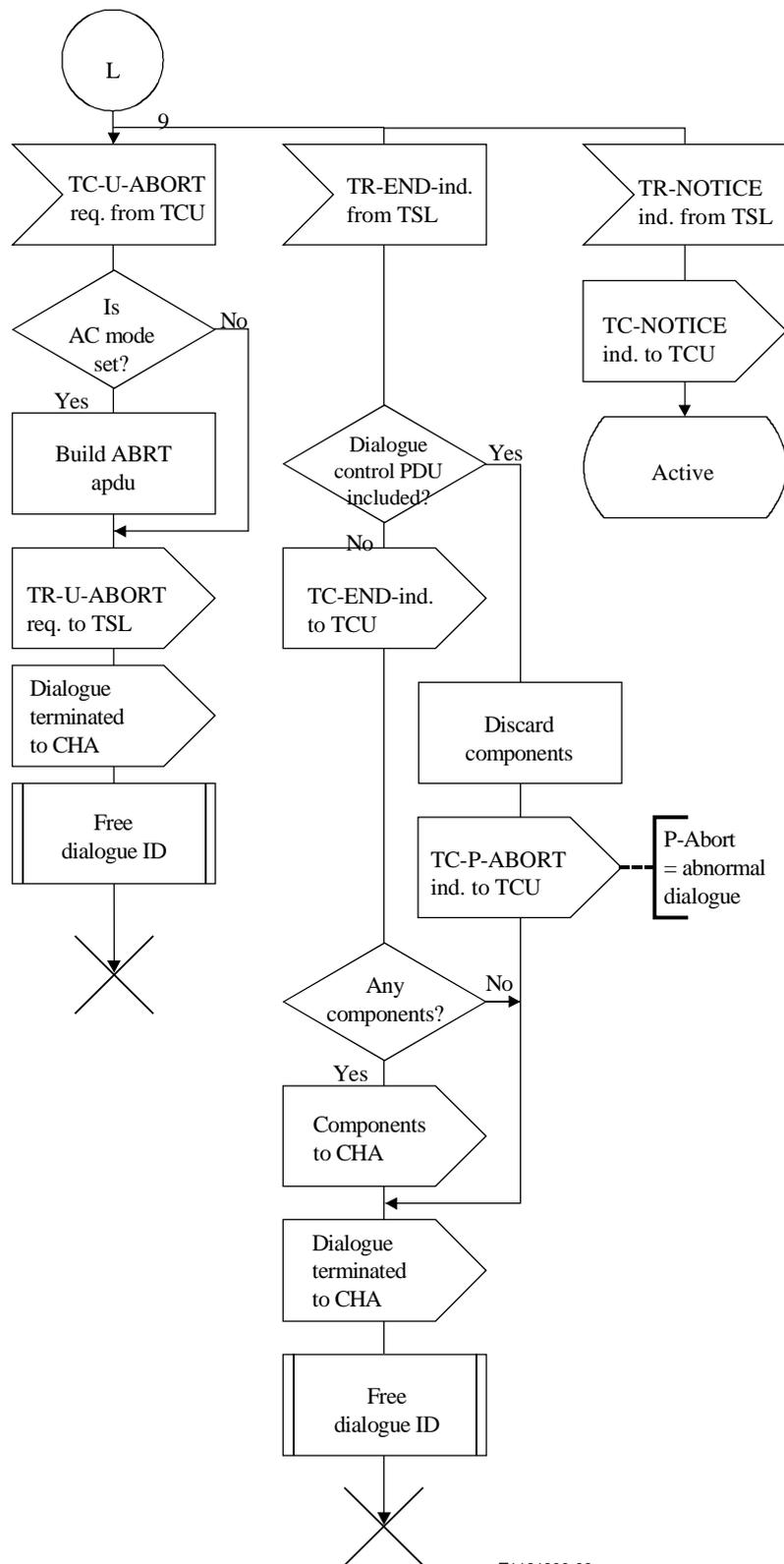
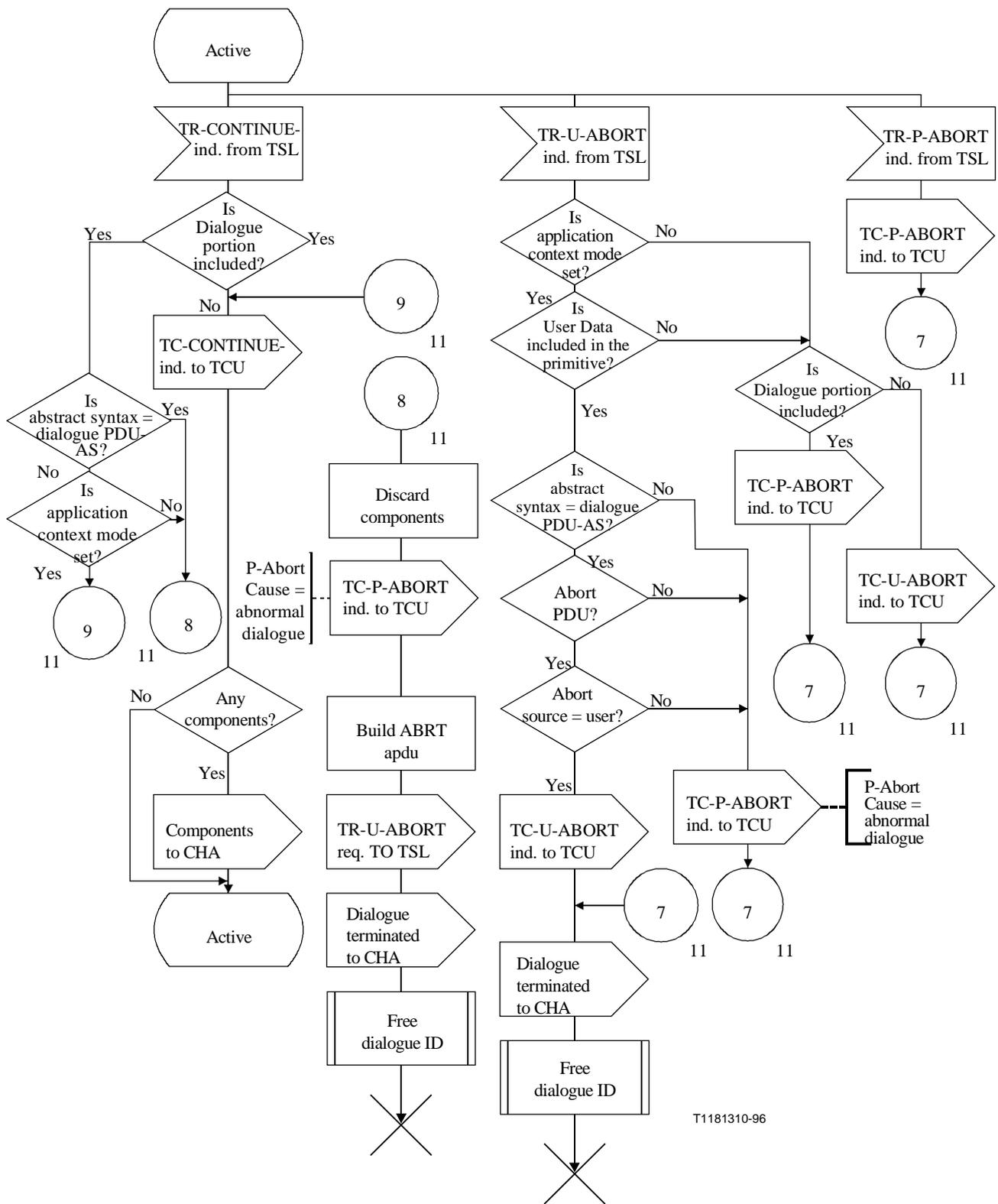


Figure A.5/Q.774 (feuille 9 de 11) – Gestion du dialogue dans la sous-couche composant



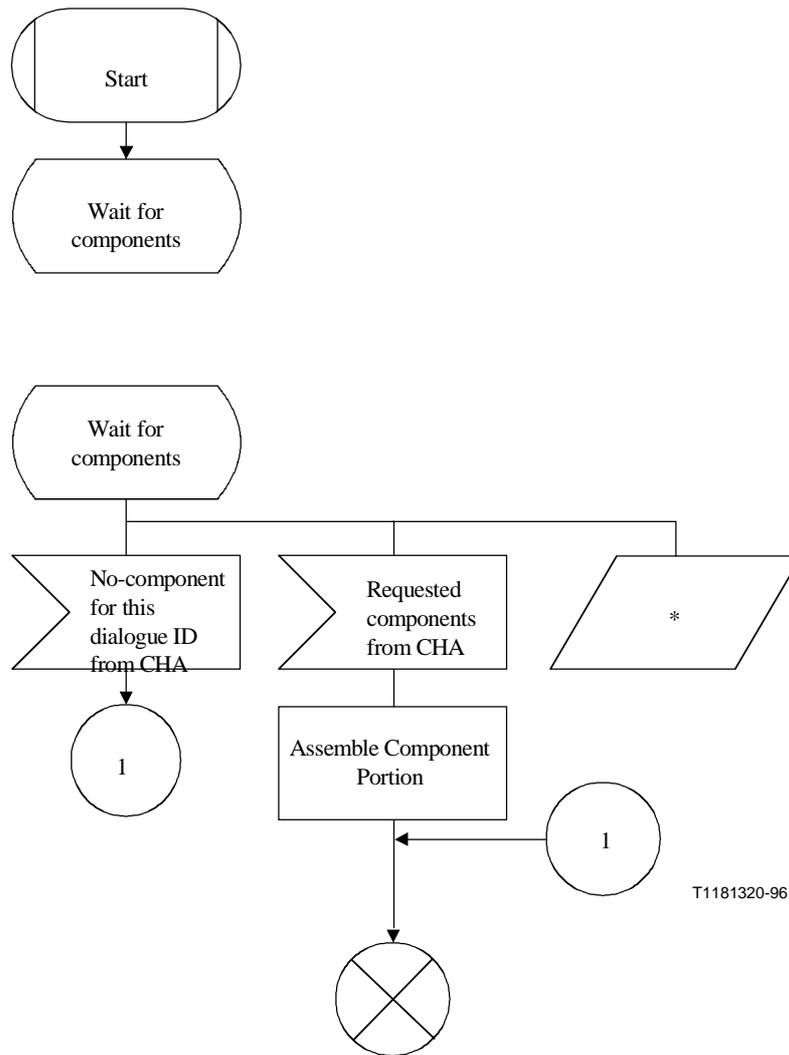
T1181300-96

Figure A.5/Q.774 (feuille 10 de 11) – Gestion du dialogue dans la sous-couche composant

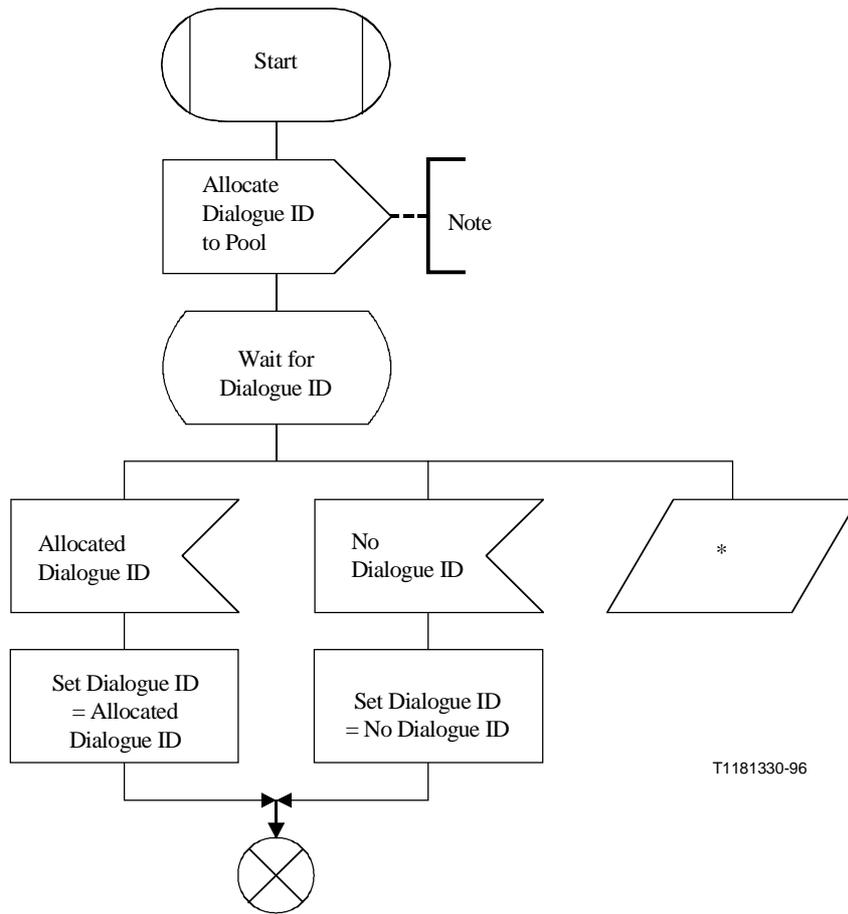


T1181310-96

Figure A.5/Q.774 (feuillet 11 de 11) – Gestion du dialogue dans la sous-couche composant

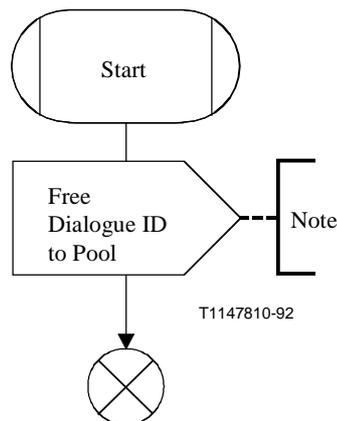


**Figure A.5 bis/Q.774 – Procédure de traitement des composantes**



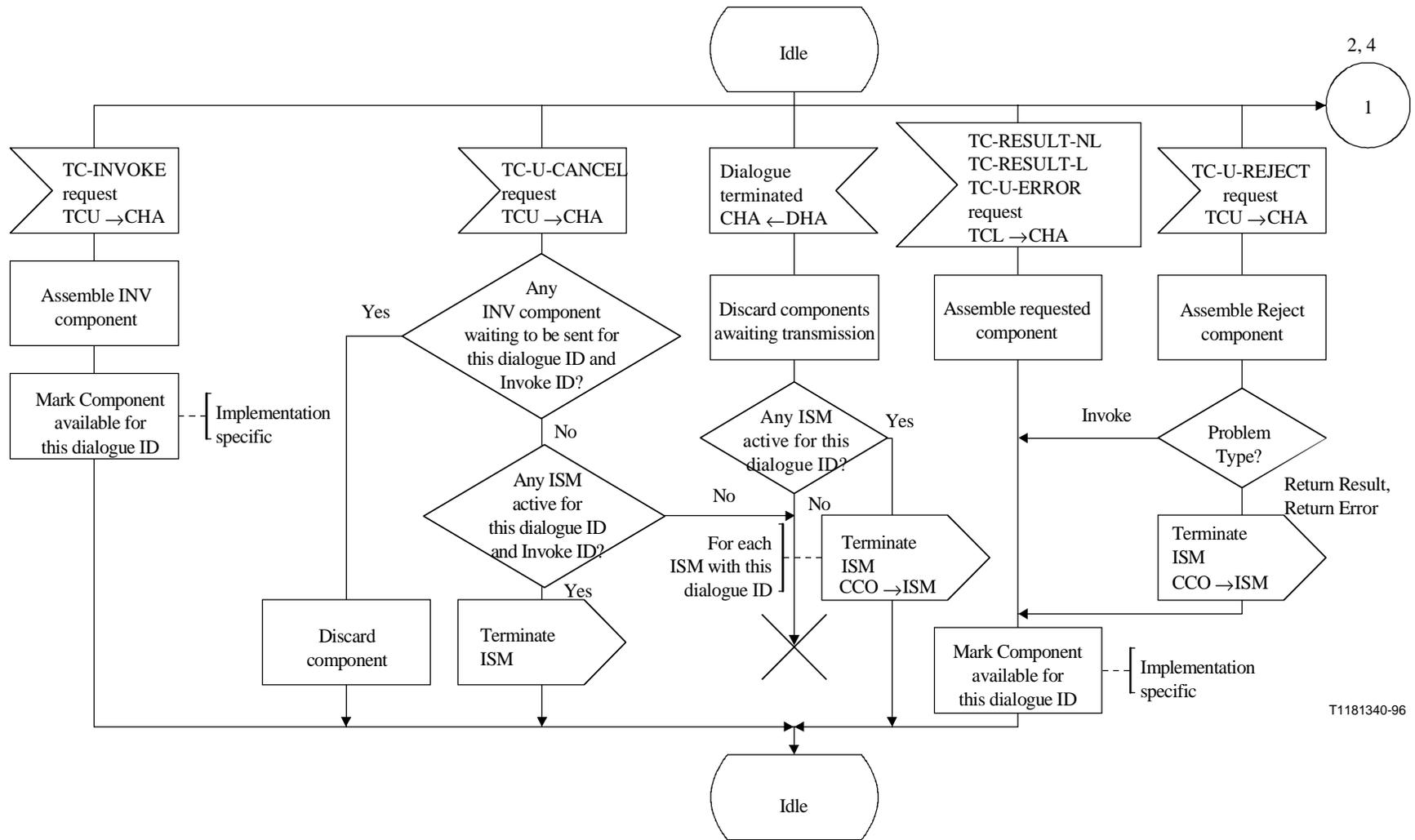
NOTE – The pool realization is implementation-dependent. The case of "No Dialogue ID" is not described in the calling process as the handling of this situation is an internal matter.

**Figure A.5 *ter*/Q.774 – Procédure ASSIGN DIALOGUE ID**



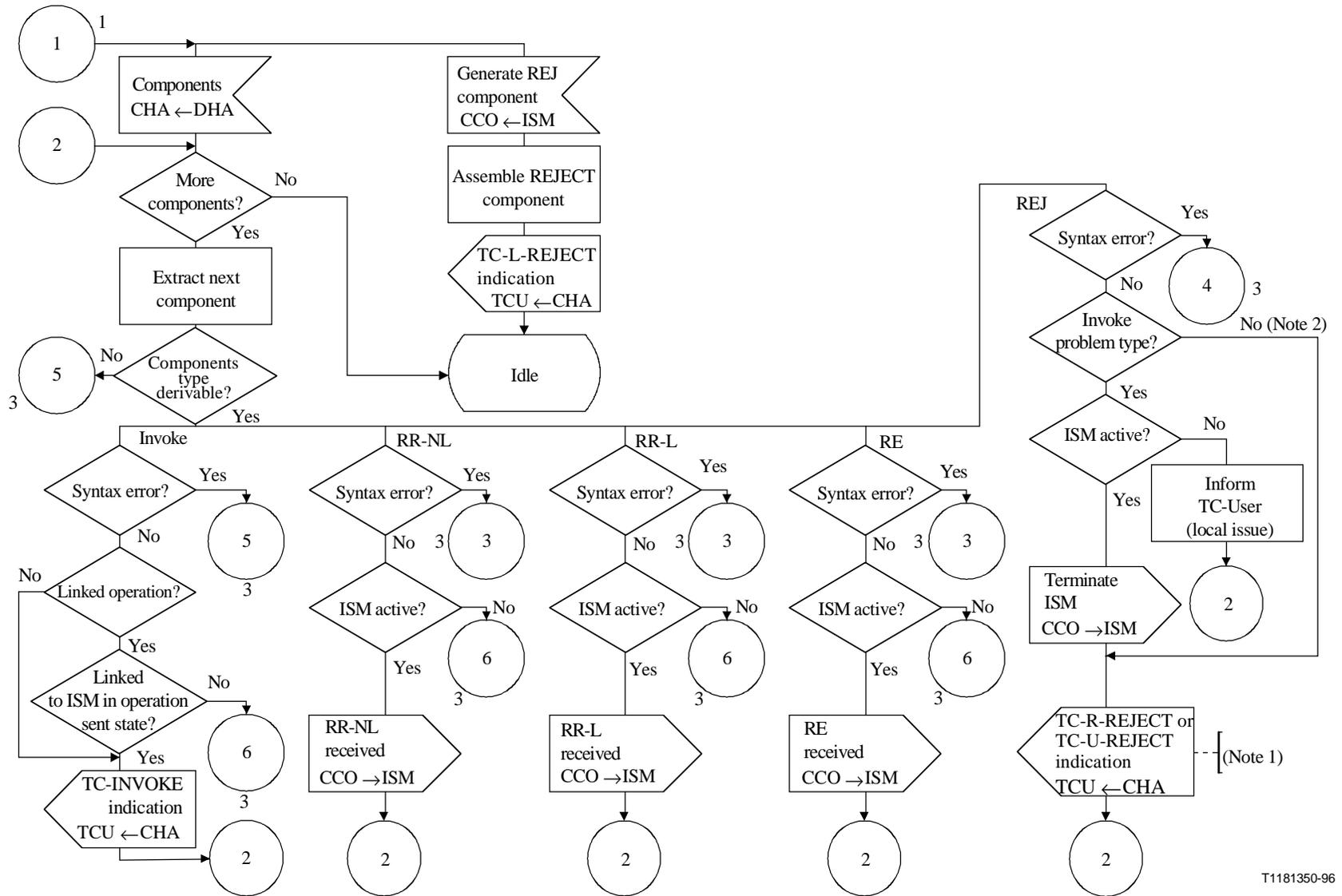
NOTE – The pool realization is implementation-dependent.

**Figure A.5 *quater*/Q.774 – Procédure FREE DIALOGUE ID**



T1181340-96

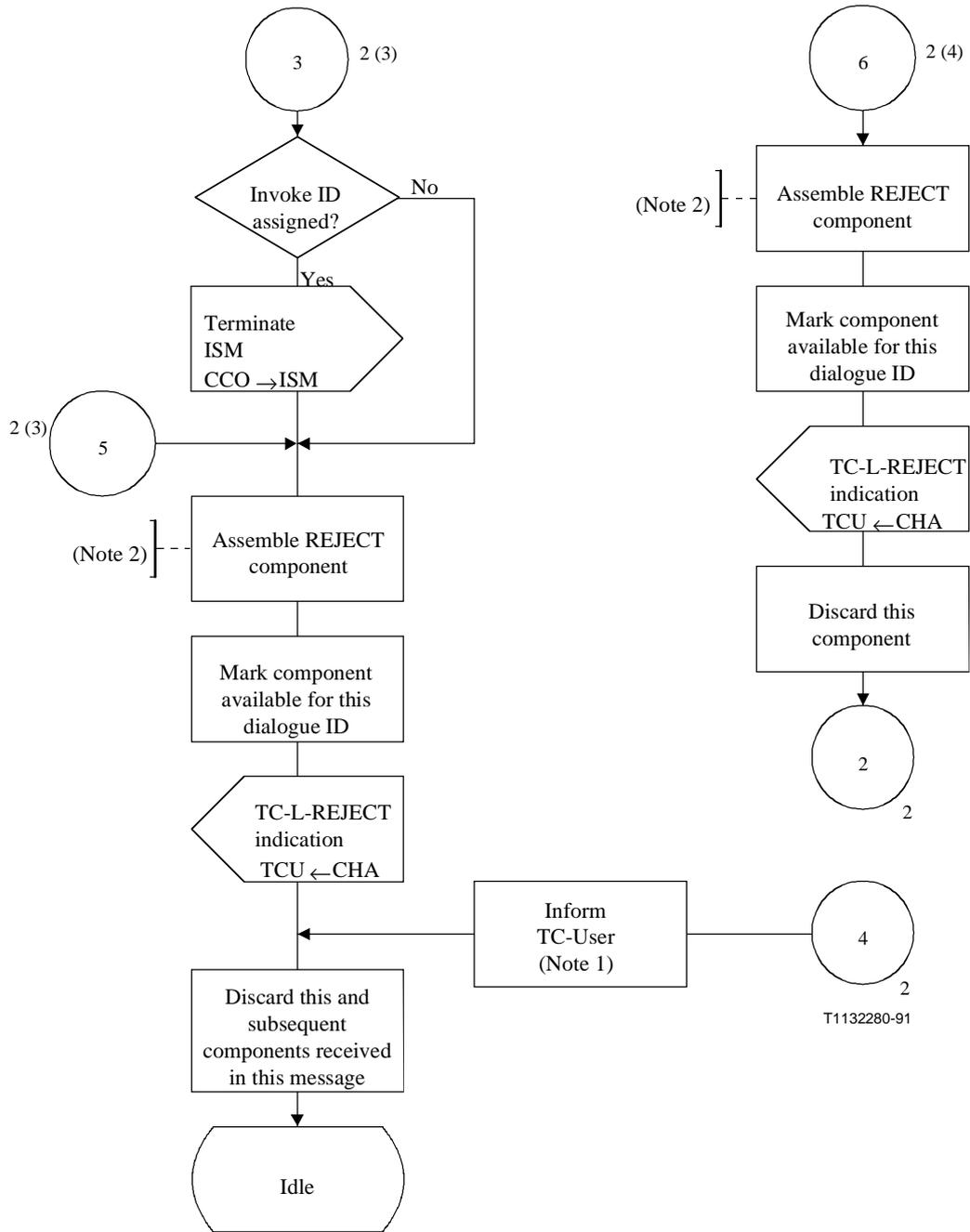
Figure A.6/Q.774 (feuille 1 de 4) – Coordonnateur de composant



NOTE 1 – Choice of TC-U-REJ or TC-R-REJ indication is determined from the Problem code value.

NOTE 2 – If "general" problem received, ID may refer to an ISM run by the other end. Therefore, a local ISM is not terminated.

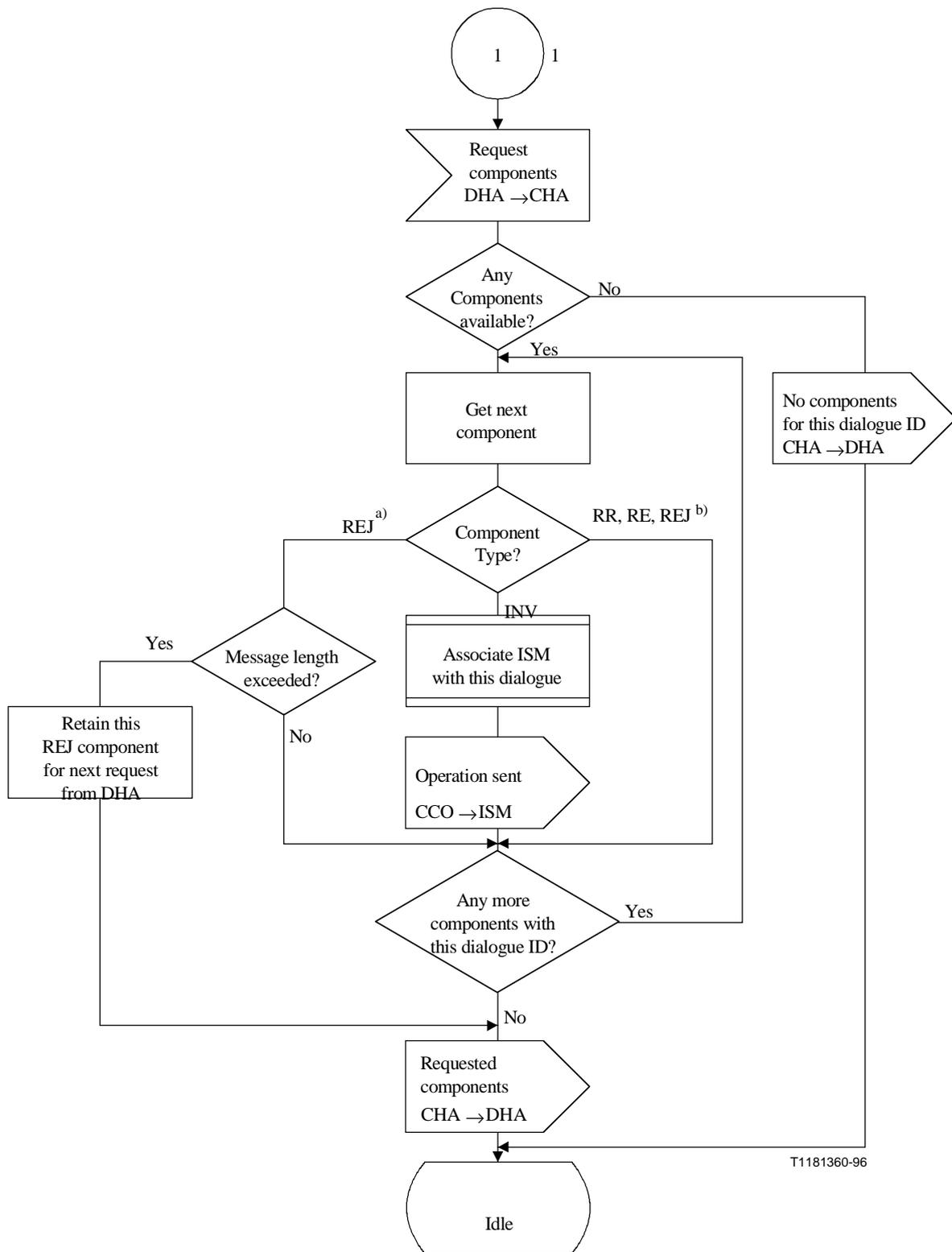
Figure A.6/Q.774 (feuille 2 de 4) – Coordonnateur de composant



NOTE 1 – Inform (local implementation) TC-user of syntax error in received Reject component.

NOTE 2 – Choose an appropriate problem code from values defined in Recommendation Q.772.

**Figure A.6/Q.774 (feuille 3 de 4) – Coodonnateur de composant**

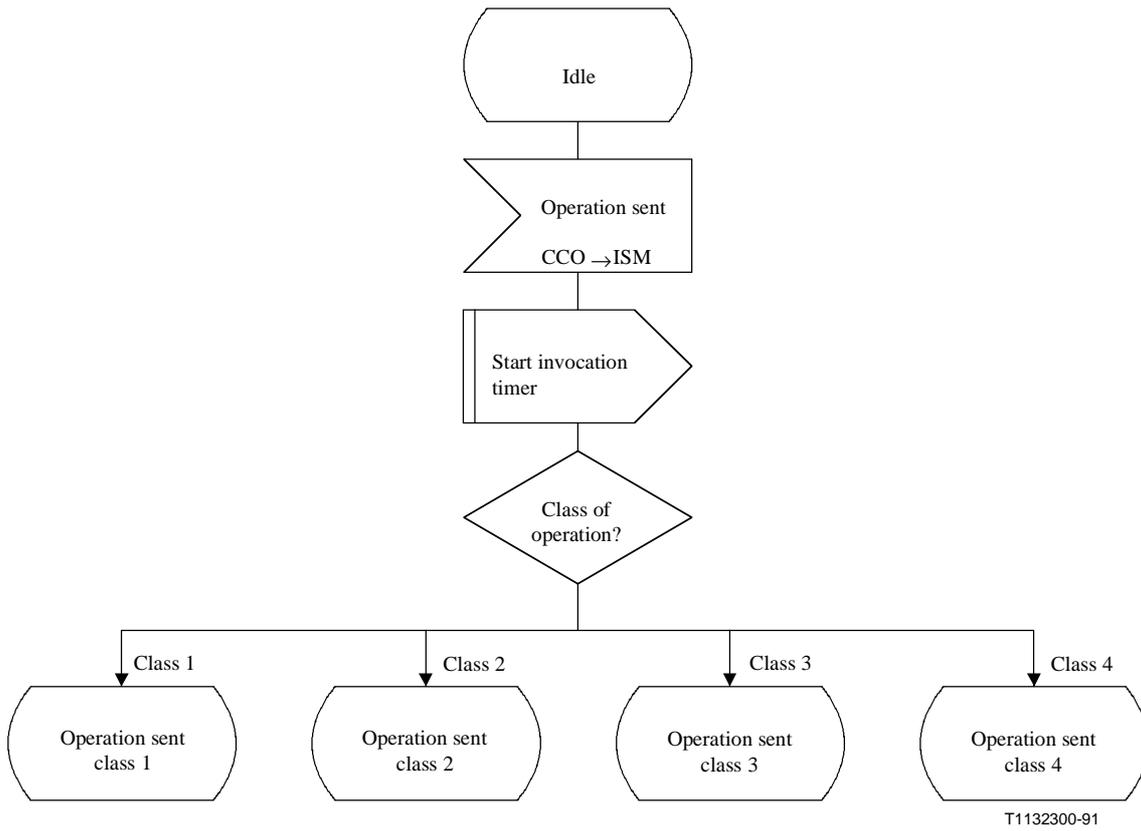


T1181360-96

a) Component sub-layer generated Reject.

b) TC-user generated Reject.

**Figure A.6/Q.774 (feuille 4 de 4) – Coodonnateur de composant**



**Figure A.7/Q.774 (feuillet 1 de 6) – Automate d'invocation**

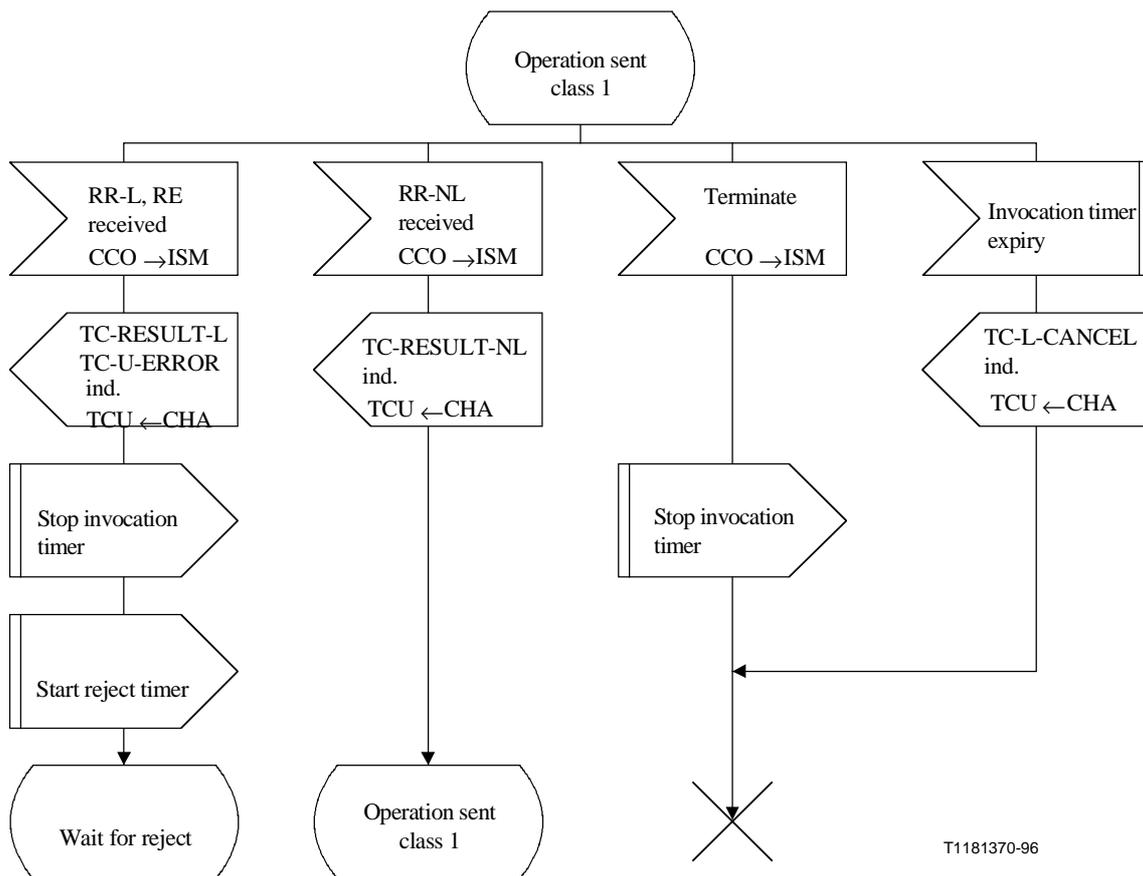


Figure A.7/Q.774 (feuillet 2 de 6) – Automate d'invocation

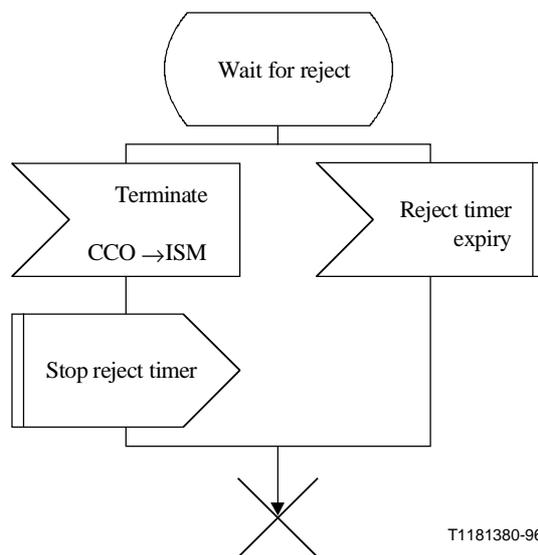
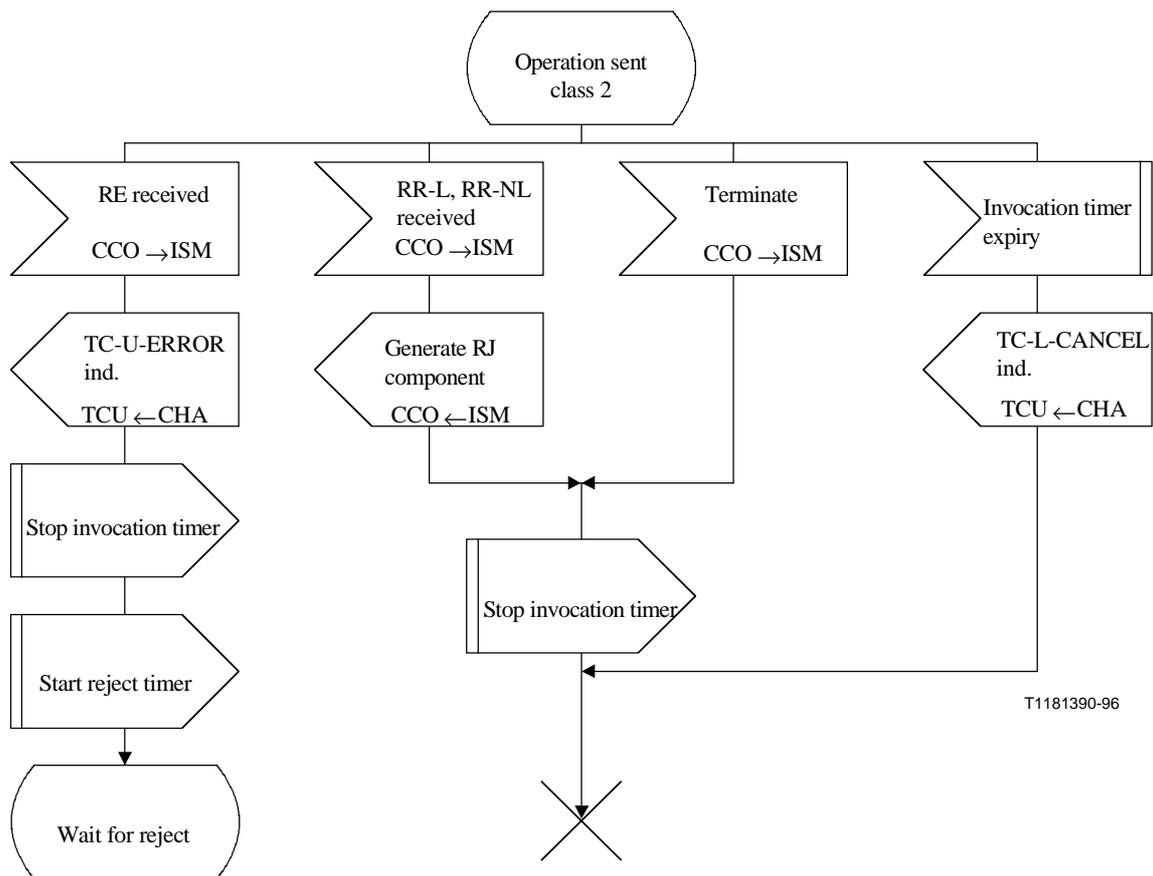


Figure A.7/Q.774 (feuillet 3 de 6) – Automate d'invocation



**Figure A.7/Q.774 (feuille 4 de 6) – Automate d'invocation**

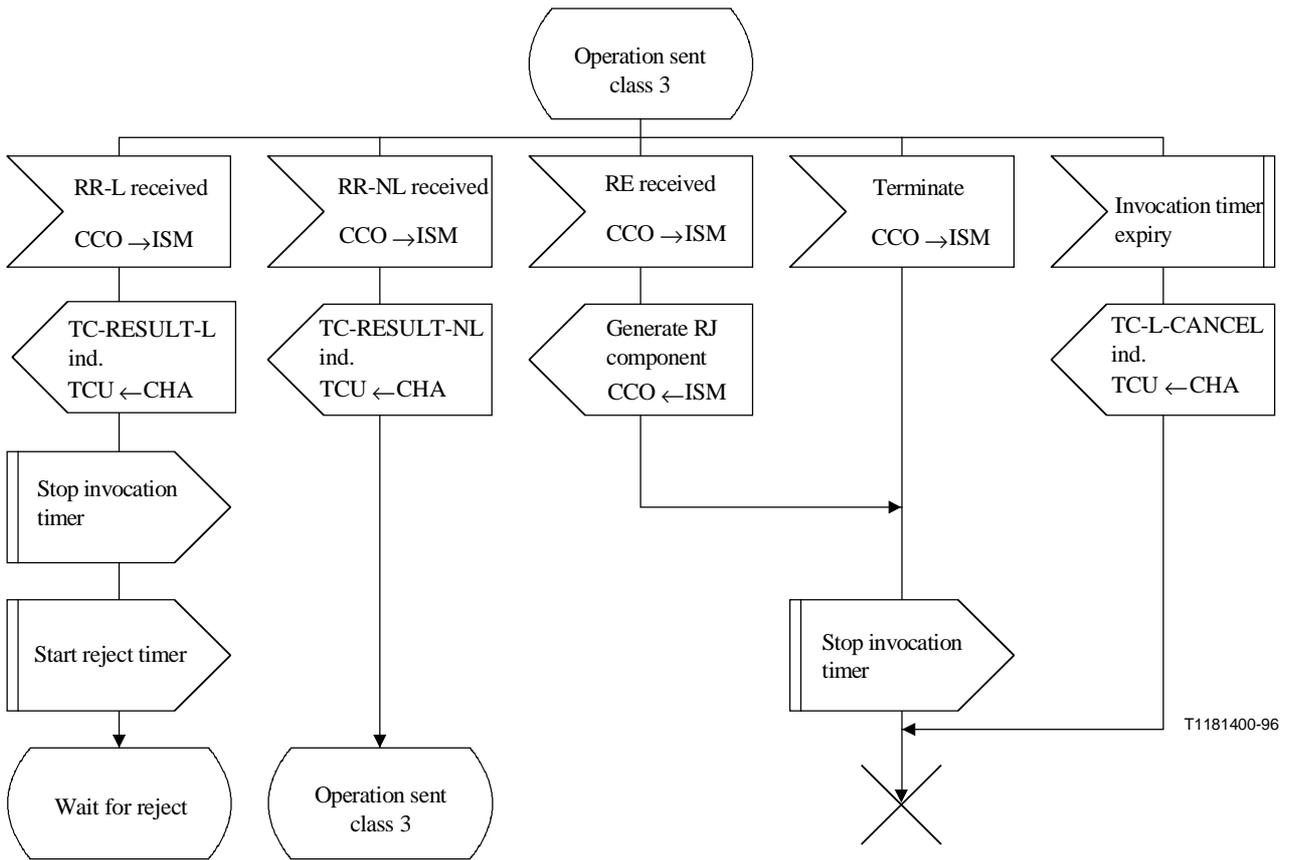


Figure A.7/Q.774 (feuillet 5 de 6) – Automate d'invocation

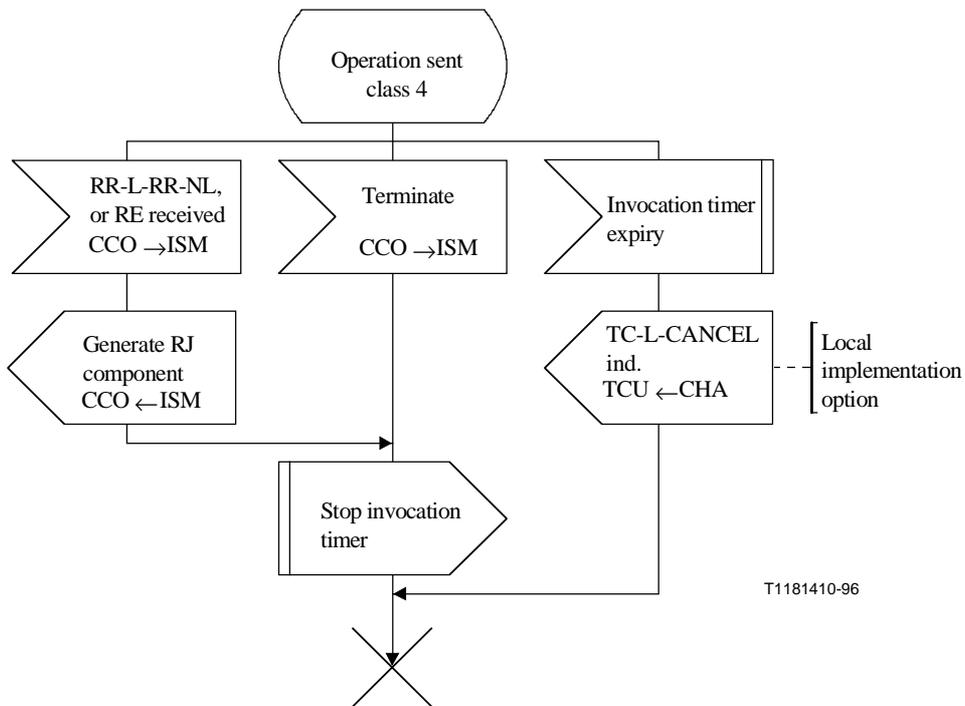


Figure A.7/Q.774 (feuillet 6 de 6) – Automate d'invocation

#### **A.4 Abréviations utilisées dans les diagrammes SDL**

CCO	coordonnateur de composant
CHA	gestion de composant ( <i>component handling</i> )
CSL	sous-couche composant ( <i>component sub-layer</i> )
DHA	gestion de dialogue (sous-couche composant) [ <i>dialogue handling (component sub-layer)</i> ]
INV	invocation
IR	état initialisation reçue ( <i>initiation received state</i> )
IS	état initialisation envoyée ( <i>initiation sent state</i> )
ISM	automate d'invocation ( <i>invocation state machine</i> )
L	composant dernier ( <i>last component</i> )
NL	composant non dernier ( <i>not last component</i> )
RE	retour erreur
REJ	rejet
RR	retour résultat
RR-L	retour résultat dernier ( <i>return result last</i> )
RR-NL	retour résultat non dernier ( <i>return result not last</i> )
SCCP	sous-système commande des connexions sémaphores ( <i>signalling connection control part</i> )
TC	gestionnaire de transactions ( <i>transaction capabilities</i> )
TCAP	sous-système application pour la gestion des transactions ( <i>transaction capabilities application part</i> )
TCO	coordonnateur de transaction ( <i>transaction coordinator</i> )
TCU	utilisateur du TC ( <i>TC-user</i> )
TSL	sous-couche transaction ( <i>transaction sub-layer</i> )
TSM	automate de transaction ( <i>transaction state machine</i> )
UNI	unidirectionnel



## SERIES 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 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, de télégraphie, de télécopie, circuits téléphoniques 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 pour données et communication entre systèmes ouverts
- Série Z Langages de programmation