

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**ISO/CEI
IEC
9506-6**

Première édition
First edition
1994-06

**Système d'automatisation industrielle –
Spécification de messagerie industrielle –**

**Partie 6:
Norme d'accompagnement
pour la conduite de procédé**

**Industrial automation systems –
Manufacturing message specification –**

**Part 6:
Companion Standard for Process Control**



Numéro de référence
Reference number
ISO/IEC 9506-6: 1994 (E/F)

SOMMAIRE

	Pages
AVANT-PROPOS	10
INTRODUCTION	14
Articles	
1 Domaine d'application	18
2 Références normatives	20
3 Définitions	20
3.1 Définitions des modèles de référence	20
3.2 Définitions spécifiques à la présente partie de l'ISO/IEC 9506	22
4 Abréviations	24
5 Description de l'application	26
5.1 Modèles de Contrôle-Commande de procédé	26
5.1.1 Modèles de communication industrielle de procédé	26
5.2 Fonctions de Contrôle-Commande de procédé	30
5.2.1 Support de régulation	30
5.2.2 Gestion des événements et des alarmes	30
5.2.3 Besoins pour la fabrication par lots	34
5.2.4 Besoins pour le stockage et la récupération d'historiques	34
5.2.5 Besoins pour la configuration et la gestion du système de Contrôle-Commande de procédé	36
5.3 Modèles d'Application	36
5.3.1 Bloc	36
5.3.2 Algorithme	38
6 Mappage du contexte de Contrôle-Commande de procédé	38
6.1 Mappage du modèle de Contrôle-Commande de procédé par rapport au VMD	38
6.2 Définition des objets de Contrôle-Commande de procédé mappant des Domaines	38
6.2.1 L'objet Bloc	38
6.2.2 L'objet Algorithme	40
6.3 Définition des objets de Contrôle-Commande de procédé qui mappent des Invocations de Programme	40
6.3.1 L'objet Bloc	40
6.4 Prescriptions de Contrôle-Commande de procédé affectant d'autres objets MMS	40
6.4.1 Extensions à l'objet Condition Événementielle	40
6.4.2 Extensions aux objets Enveloppe Événementielle	44
6.5 Définitions des nouveaux objets abstraits MMS	44
6.5.1 Objet Liste de Condition Événementielle	44
6.5.2 Objet Contrôle-Commande d'Unité	48

© ISO/IEC 1994

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 131 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

CONTENTS

	Page
FORWORD	11
INTRODUCTION	15
 Clause	
1 Scope	19
2 Normative references	21
3 Definitions	21
3.1 Reference model definitions	21
3.2 Definitions unique to this part of ISO/IEC 9506	23
4 Abbreviations	25
5 Application description	27
5.1 Process control models	27
5.1.1 Process manufacturing communication models	27
5.2 Process control functions	31
5.2.1 Support for regulatory control	31
5.2.2 Management of events and alarms	31
5.2.3 Support for batch manufacturing	35
5.2.4 Support for creation and retrieval of historical records	35
5.2.5 Support for process control system configuration and management	37
5.3 Application Models	37
5.3.1 Block	37
5.3.2 Algorithm	39
6 Process control context mapping	39
6.1 Mapping the process control model to the VMD	39
6.2 Definition of process control objects that map to Domains	39
6.2.1 The Block object	39
6.2.2 The Algorithm object	41
6.3 Definition of process control objects that map to Program Invocations	41
6.3.1 The Block object	41
6.4 Process control requirements that affect other MMS objects	41
6.4.1 Extensions to the Event Condition object	41
6.4.2 Extensions to Event Enrollment objects	45
6.5 Definitions of new MMS abstract objects	45
6.5.1 Event Condition List object	45
6.5.2 Unit Control object	49

Articles	Pages
6.6 Conformité des paramètres	50
6.6.1 DES	50
6.6.2 DEI	50
6.6.3 RECL	50
7 Services	50
7.1 Utilisation des services ACSE	50
7.2 Utilisation des services MMS	52
7.2.1 Nom de contexte d'application de procédé	52
7.2.2 Définition de la syntaxe abstraite de Contrôle-Commande de procédé	52
7.2.3 Définition du module ASN.1 spécifique au Contrôle-Commande de procédé	52
7.2.4 Services relatifs au VMD	54
7.2.5 Services de gestion d'événements	56
7.2.6 Autres paramètres	78
7.3 Définition et utilisation des services spécifiques de Contrôle-Commande de procédé	84
7.3.1 Control Element (Elément de Contrôle-Commande)	84
7.3.2 Service InitiateUnitControlLoad (Déclencher Chargement de Contrôle-Commande d'Unité)	90
7.3.3 Service UnitControlLoadSegment (Segment de Chargement de Contrôle-Commande d'Unité)	94
7.3.4 Service UnitControlUpload (Télésauvegarde de Contrôle-Commande d'Unité)	100
7.3.5 Service StartUnitControl (Démarrage Contrôle-Commande d'Unité)	108
7.3.6 Service StopUnitControl (Arrêt Contrôle-Commande d'Unité)	112
7.3.7 Service CreateUnitControl (Créer Contrôle-Commande d'Unité)	118
7.3.8 Service AddToUnitControl (Ajouter à Contrôle-Commande d'Unité)	120
7.3.9 Service RemoveFromUnitControl (Retirer de Contrôle-Commande d'Unité)	124
7.3.10 Service GetUnitControlAttributes (Obtenir Attributs Contrôle-Commande d'Unité)	128
7.3.11 Service LoadUnitControlFromFile (Charger Contrôle-Commande d'Unité du Fichier)	132
7.3.12 Service StoreUnitControlToFile (Archiver Contrôle-Commande d'Unité dans Fichier)	138
7.3.13 Service DeleteUnitControl (Détruire Contrôle-Commande d'Unité)	142
7.3.14 Service DefineEventConditionList (Définir Liste de Conditions Événementielles)	146
7.3.15 Service DeleteEventConditionList (Détruire Liste de Conditions Événementielles)	154
7.3.16 Service AddEventConditionListReference (Ajouter Référence de Liste de Conditions Événementielles)	158
7.3.17 Service RemoveEventConditionListReference (Retirer Référence de Liste de Conditions Événementielles)	164
7.3.18 Service GetEventConditionListAttributes (Obtenir Attributs de Liste de Conditions Événementielles)	170
7.3.19 Service ReportEventConditionListStatus (Rendre Compte de l'Etat de Liste de Conditions Événementielles)	174

Clause	Page
6.6 Parameter conformance	51
6.6.1 DES	51
6.6.2 DEI	51
6.6.3 RECL	51
7 Services	51
7.1 Use of ACSE services	51
7.2 Use of MMS services	53
7.2.1 Process application context name	53
7.2.2 Process control abstract syntax definition	53
7.2.3 Process control specific ASN.1 module definition	53
7.2.4 VMD support services	55
7.2.5 Event management services	57
7.2.6 Other productions	79
7.3 Definition and use of process control specific services	85
7.3.1 Control Element	85
7.3.2 InitiateUnitControlLoad service	91
7.3.3 UnitControlLoadSegment service	95
7.3.4 UnitControlUpload service	101
7.3.5 StartUnitControl service	109
7.3.6 StopUnitControl service	113
7.3.7 CreateUnitControl service	119
7.3.8 AddToUnitControl service	121
7.3.9 RemoveFromUnitControl service	125
7.3.10 GetUnitControlAttributes service	129
7.3.11 LoadUnitControlFromFile service	133
7.3.12 StoreUnitControlToFile service	139
7.3.13 DeleteUnitControl service	143
7.3.14 DefineEventConditionList service	147
7.3.15 DeleteEventConditionList service	155
7.3.16 AddEventConditionListReference service	159
7.3.17 RemoveEventConditionListReference service	165
7.3.18 GetEventConditionListAttributes service	171
7.3.19 ReportEventConditionListStatus service	175

Articles	Pages
7.3.20 Service AlterEventConditionListMonitoring (Modifier Surveillance de Liste de Conditions Événementielles)	182
7.4 Le Service et le Protocole Déclencher	186
7.4.1 Paramètre Supplémentaire de Demande de Déclenchement	186
7.4.2 Paramètre Supplémentaire de Réponse d'Initialisation	190
7.4.3 Protocole Initiate (Déclenchement)	194
7.5 Extensions de protocole généralisées	196
7.5.1 Extensions ConfirmedServiceRequest (Demande de Service Confirmée)	196
7.5.2 Extensions ConfirmedServiceResponse (Réponse de Service Confirmée)	198
7.5.3 Extensions erreur spécifique au service	198
7.6 Fin de Module	200
8 Objets normalisés	200
9 Conformité	200
9.1 Classes de conformité	200
9.1.1 Définition des classes de conformité	200
9.1.2 Services exigés pour les classes de conformité	202
9.1.3 CBB de paramètre exigés pour les classes de conformité	208
9.2 Partie Un de la PICS: Information de Mise en oeuvre	210
9.3 Partie Deux de la PICS: CBB de Service	210
9.4 Partie Trois de la PICS: CBB de paramètre	210
9.5 Partie Quatre de la PICS: Valeurs de mise en oeuvre locales	212
Figures	
1 Classes de communication	16
2 Interaction dans un environnement d'homologues	26
3 Exemple d'une configuration matérielle	28
Tableaux	
1 Extensions du paramètre DefineEventCondition	56
2 Extensions du paramètre GetEventConditionAttributes	60
3 Extensions du paramètre AlterEventConditionMonitoring	64
4 Extensions du paramètre DefineEventEnrollment	68
5 Extensions du paramètre GetEventEnrollmentAttributes	70
6 Extensions du paramètre AlterEventEnrollment	72
7 Extensions du paramètre EventNotification	76
8 Paramètre d'élément de Contrôle-Commande	84
9 Interaction des primitives Contrôle-Commande d'Unité	90
10 Service InitiateUnitControlLoad	90
11 Service UnitControlLoadSegment	96
12 Service UnitControlUpload	100
13 Service StartUnitControl	108
14 Service StopUnitControl	114
15 Service CreateUnitControl	118
16 Service AddToUnitControl	122
17 Service RemoveFromUnitControl	126
18 Service GetUnitControlAttributes	130
19 Service LoadUnitControlFromFile	134
20 Service StoreUnitControlToFile	138
21 Service DeleteUnitControl	142

Clause		Page
7.3.20	AlterEventConditionListMonitoring service	183
7.4	The Initiate Service and Protocol	187
7.4.1	Init Request Detail parameter	187
7.4.2	Init Response Detail parameter	191
7.4.3	Initiate protocol	195
7.5	Generalized protocol extensions	197
7.5.1	ConfirmedServiceRequest extensions	197
7.5.2	ConfirmedServiceResponse extensions	199
7.5.3	Service specific error extensions	199
7.6	End of module	201
8	Standardized objects	201
9	Conformance	201
9.1	Conformance classes	201
9.1.1	Definition of conformance classes	201
9.1.2	Services required for conformance classes	203
9.1.3	Parameter CBBs required for conformance classes	209
9.2	PICS Part One: Implementation information	211
9.3	PICS Part Two: Service CBBs	211
9.4	PICS Part Three: Parameter CBBs	211
9.5	PICS Part Four: Local implementation values	213
Figures		
1	Classes of communication	17
2	Interaction in a peer environment	27
3	An example hardware configuration	29
Tables		
1	DefineEventCondition extensions	57
2	GetEventConditionAttributes extensions	61
3	AlterEventConditionMonitoring extensions	65
4	DefineEventEnrollment extensions	69
5	GetEventEnrollmentAttributes extensions	71
6	AlterEventEnrollment extensions	73
7	EventNotification extensions	77
8	Control Element Parameter	85
9	Interaction of Unit Control primitives	91
10	InitiateUnitControlLoad service	91
11	UnitControlLoadSegment service	97
12	UnitControlUpload service	101
13	StartUnitControl service	109
14	StopUnitControl service	115
15	CreateUnitControl service	119
16	AddToUnitControl service	123
17	RemoveFromUnitControl service	127
18	GetUnitControlAttributes service	131
19	LoadUnitControlFromFile service	135
20	StoreUnitControlToFile service	139
21	DeleteUnitControl service	143

22	Service DefineEventConditionList	148
23	Service DeleteEventConditionList	154
24	Service AddEventConditionListReference	158
25	Service RemoveEventConditionListReference	166
26	Service GetEventConditionListAttributes	172
27	Service ReportEventConditionListStatus	176
28	Service AlterEventConditionListMonitoring	182
29	Paramètre Supplémentaire Demande de Déclenchement	186
30	Paramètre Supplémentaire de Réponse de Déclenchement	190
31	Classes de conformité	200
32	Exigences de service pour les classes de conformité	204
33	Exigences de paramètre pour les classes de conformité	208
34	CBB de services supplémentaires	210
35	CBB de paramètres supplémentaires	212
Annexes		
A	Modèle d'Association d'Application	214
B	Concept de bloc	218
C	Utilisation de la présente partie de l'ISO/IEC 9506 pour le traitement par lots	230
D	Définitions des symboles de bloc	234
INDEX		262

22	DefineEventConditionList service	149
23	DeleteEventConditionList service	155
24	AddEventConditionListReference service	159
25	RemoveEventConditionListReference service	166
26	GetEventConditionListAttributes service	172
27	ReportEventConditionListStatus service	177
28	AlterEventConditionListMonitoring service	183
29	Init Request Detail parameter	187
30	Init Response Detail parameter	191
31	Conformance classes	201
32	Service requirements for conformance classes	205
33	Parameter requirements for conformance classes	209
34	Additional service CBBs	211
35	Additional parameter CBBs	213
 Annexes		
A	Application Association model	215
B	Block concepts	219
C	Use of this part of ISO/IEC 9506 for batch processing	231
D	Block symbol definitions	235
INDEX		265

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

SYSTÈME D'AUTOMATISATION INDUSTRIELLE – SPÉCIFICATION DE MESSAGERIE INDUSTRIELLE –

Partie 6: Norme d'accompagnement pour la conduite de procédé

AVANT-PROPOS

1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.

2) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par les comités d'études où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.

3) Ces décisions constituent des recommandations internationales publiées sous forme de normes, de rapports techniques ou de guides et agréées comme telles par les Comités nationaux.

4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure possible, les Normes internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la norme de la CEI et la norme nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.

La Norme internationale ISO/IEC 9506-6 a été établie par le sous-comité 65C: Communications numériques, du comité d'études 65 de la CEI: Mesure et commande dans les processus industriels.

Le texte de cette norme est issu des documents suivants:

DIS	Rapport de vote
65C(BC)35	65C(BC)38

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

L'ISO/IEC 9506 comprend les parties suivantes, présentées sous le titre général *Système d'automatisation industrielle – Spécification de messagerie industrielle*:

- Partie 1: 1990, *Définition de service* (publiée actuellement en anglais seulement)
- Partie 2: 1990, *Spécification de protocole* (publiée actuellement en anglais seulement)
- Partie 3: 1991, *Norme d'accompagnement pour la robotique* (publiée actuellement en anglais seulement)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL AUTOMATION SYSTEMS –
MANUFACTURING MESSAGE SPECIFICATION –

Part 6: Companion Standard for Process Control

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international cooperation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.

3) They have the form of recommendations for international use published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.

4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.

International Standard ISO/IEC 9506-6 has been prepared by IEC by sub-committee 65C: Digital communications, or IEC technical committee 65: Industrial-process measurement and control.

The text of this standard is based on the following documents:

DIS	Report on voting
65C(CO)35	65C(CO)38

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

ISO/IEC 9506 consists of the following parts, under the general title *Industrial automation systems – Manufacturing message specification*:

- Part 1: 1990, *Service definition*
- Part 2: 1990, *Protocol specification*
- Part 3: 1991, *Companion standard for robotics*

- Partie 4: 1992, *Norme d'accompagnement pour la commande numérique* (publiée en anglais seulement)
- Partie 5: 199X (à l'étude)
- Partie 6: 1994, *Norme d'accompagnement pour la conduite de procédé*.

Il est recommandé d'utiliser cette partie de la norme lorsque les systèmes de commande de procédé sont reliés à un réseau employant les services et le protocole de MMS. En liaison avec MMS et ses autres normes d'accompagnement, la présente partie de cette norme permettra l'interconnexion en réseau de différentes classes de dispositifs programmables sur le site de l'usine.

L'annexe A fait partie intégrante de cette norme.

Les annexes B, C et D sont données uniquement à titre d'information.

Withdrawing
IECNORM.COM : Click to view the full PDF of ISO/IEC 9506-6:1994

- Part 4: 1992, *Companion standard for numerical control*
- Part 5: 199X (under consideration)
- Part 6: 1994, *Companion standard for process control*.

It is recommended to use this part of the standard when process control systems are linked to a network using the services and protocol. Together with MMS and its other companion standards, this part of this Standard will enable the networking of different classes of programmable devices on the factory floor.

Annex A forms an integral part of this standard.

Annexes B, C and D are for information only.

Withdrawing
IECNORM.COM : Click to view the full PDF of ISO/IEC 9506-6:1994

INTRODUCTION

Généralités

Le présent document fait partie d'une norme destinée à faciliter l'interconnexion de systèmes informatisés. Elle s'inscrit, au sein de la couche d'application de l'Environnement d'Interconnexion de Systèmes Ouverts, comme un élément de service d'application (ASE) complémentaires aux autres normes apparentées, dans le cadre du modèle de référence de base pour l'interconnexion de systèmes ouverts (ISO 7498).

Le but de l'interconnexion de systèmes ouverts est de permettre, par le biais d'un minimum d'accords techniques en dehors des normes d'interconnexion, l'interconnexion de systèmes informatisés:

- a) de fabricants différents;
- b) utilisant des gestions systèmes différents;
- c) de niveaux de complexité différents;
- d) dont la progressivité de mise en oeuvre est différente.

Cette partie de l'ISO/IEC 9506 traite de la communication et de l'interconnexion d'équipements programmables de systèmes de Contrôle-Commande de processus industriels utilisés dans l'industrie de procédé.

Objet

L'objet de cette partie de l'ISO/IEC 9506 est de favoriser l'utilisation de la Spécification de Messagerie Industrielle selon l'ISO/IEC 9506-1 et l'ISO/IEC 9506-2, pour les applications de Contrôle-Commande de procédé.

Cette partie de l'ISO/IEC 9506 est une norme d'accompagnement de la Spécification de Messagerie Industrielle (MMS). Elle utilise également et fait référence à la norme de Définition des Elément de Service de Contrôle d'Association (ISO 8649) dont elle intègre les dispositions afin de réaliser les objectifs de la Spécification de Messagerie Industrielle. Dans l'Environnement de Contrôle-Commande de Procédé, trois formes de communications sont reconnues; elles sont groupées par classes dans la figure 1 et décrites ci-dessous:

Classe A - Communication entre un calculateur de supervision et un système de Contrôle-Commande de procédé (SCC) ou communication entre un calculateur et un système réparti de Contrôle-Commande de procédé (SNCC). Le calculateur exécute des fonctions de haut niveau qui ne font pas partie des fonctionnalités du SCC ou du SNCC. Il est admis que ces fonctions comprennent la supervision, le suivi et la gestion de production, le télédiagnostic, les conseils de système expert ou toute combinaison de ces fonctions ou d'autres fonctions non contenues dans le SCC ou SNCC spécifique utilisé pour servir l'application. La communication dans un SNCC peut être propre au fabricant du SNCC, auquel cas une passerelle est nécessaire au niveau du SNCC pour assurer des services et des protocoles de communication conformes à la présente partie de l'ISO/IEC 9506. Il n'est pas exclu que les fonctions de passerelle résident dans un équipement spécial au niveau du SNCC ou partagent un équipement SNCC avec d'autres fonctionnalités. La communication au moyen d'une passerelle peut être utilisée pour relier des systèmes SCC, des systèmes SNCC ou les deux à la fois, si nécessaire.

INTRODUCTION

General

This document is a part of a standard developed to facilitate the interconnection of information processing systems. It is positioned within the application layer of the Open Systems Interconnection environment as an application service element (ASE) with respect to other related standards by the basic reference model for open systems interconnection (ISO 7498).

The aim of open systems interconnection is to allow, with a minimum of technical agreement outside the interconnection standards, the interconnection of information processing systems:

- a) from different manufacturers;
- b) under different managements;
- c) of different levels of complexity;
- d) of different evolutionary implementations.

This part of ISO/IEC 9506 is concerned with the communications and interworking of programmable devices of industrial process control systems utilized in the process industries.

Purpose

The purpose of this part of ISO/IEC 9506 is to augment the use of the Manufacturing Message Specification, ISO/IEC 9506-1 and ISO/IEC 9506-2, for process control applications.

This part of ISO/IEC 9506 is a companion standard to the Manufacturing Message Specification (MMS). It also uses and references the Association Control Service Element Definition (ISO 8649) whose provisions it assumes in order to accomplish the aims of the Manufacturing Message Specification. In the Process Control Environment, three forms of communication are recognized; these classes are depicted in figure 1 and described as follows:

Class A - Communications between a computer and a process control system (PCS), or communications between a computer and a distributed process control system (DCS). The computer performs higher level functions which are not part of the PCS or DCS functionality. These functions may include supervisory control, production control and management, remote diagnosis, expert system advice, or any combination of these or other functions not contained within the specific PCS or DCS employed to serve the application. Communications within a DCS may be proprietary to the manufacturer of the DCS, in which case a gateway is required at the DCS to provide communications services and protocols in conformance with this part of ISO/IEC 9506. The gateway functions may reside within a special device at the DCS or may share a DCS device with other functionality. Communications with a gateway may be used to connect PCS systems, or DCS systems, or both where necessary.

Classe B - Communication entre les équipements constitutifs du SNCC. Le SNCC est constitué d'équipements de divers fabricants et la communication entre ces équipements utilise les services et le protocole spécifiés dans la présente partie de l'ISO/IEC 9506.

Classe C - Communication entre un SCC ou un SNCC et des équipements utilisateur tels que des capteurs, des actionneurs et des multiplexeurs de terrain, ainsi que la communication entre équipements de terrain.

La description de l'application, dans l'article 5 de la présente partie de l'ISO/IEC 9506, traite principalement de la communication de classe A. Cette description peut en partie s'appliquer à la communication de classe B. L'interconnexion de systèmes satisfaisant à la présente partie de l'ISO/IEC 9506 et de systèmes prenant en charge la communication de classe C peut faire l'objet d'autres efforts de normalisation, par exemple, dans le cadre des travaux relatifs au Bus de terrain.

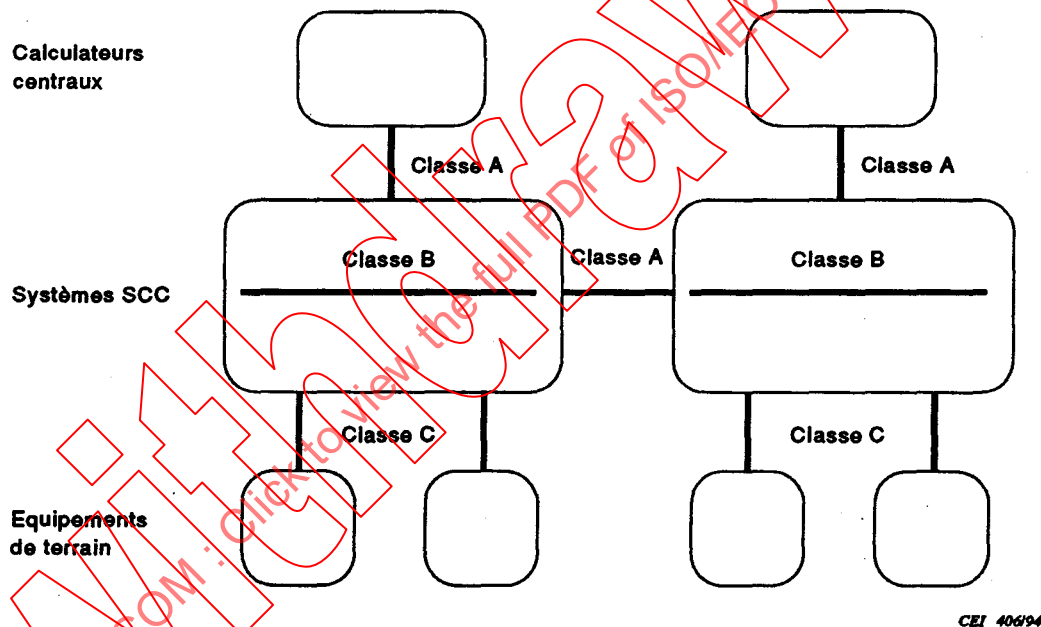


Figure 1 – Classes de communication

La présente partie de l'ISO/IEC 9506 met particulièrement l'accent sur la communication nécessaire à la réalisation des fonctions de supervision, de surveillance et de Contrôle-Commande, appelée communication de classe A. Les caractéristiques spécifiques de ce type de communication comprennent, de manière non limitative:

- les communications homologue entre calculateurs de mises en oeuvre diverses, utilisés pour le Contrôle-Commande et la gestion de la production et divers systèmes de Contrôle-Commande de procédé;
- l'accès aux Blocs fonctionnels de Contrôle-Commande et de supervision ainsi qu'à leurs attributs pour obtenir un meilleur Contrôle-Commande;
- les communications d'homologue à homologue avec d'autres équipements utilisés dans des applications de Contrôle-Commande de procédé, telles que les contrôleurs programmables.

Class B - Communications between the component devices of the DCS. The DCS is composed of devices from multiple manufacturers, and communications between devices take place using the services and protocol specified by this part of ISO/IEC 9506.

Class C - Communications between a PCS or a DCS and field devices including sensors, actuators and field multiplexers, as well as communication between field devices.

The application description in clause 5 of this part of ISO/IEC 9506 focuses primarily on class A communications. The description may be partially applicable to class B communications. Interworking of systems conforming to this part of ISO/IEC 9506 and systems supporting Class C communications may be the subject of other standardization efforts, e.g. the International Fieldbus work.

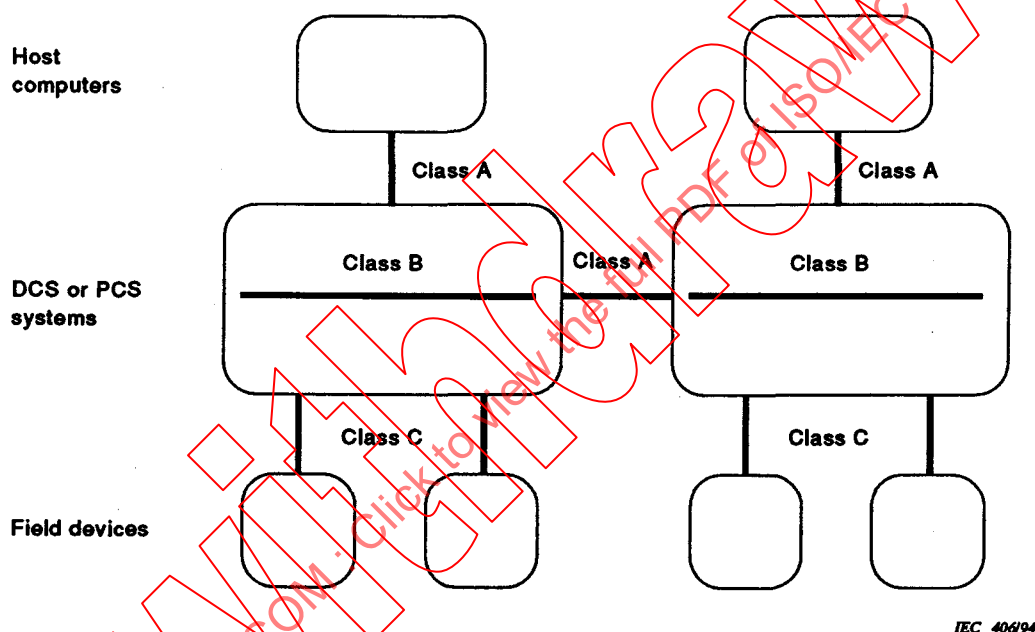


Figure 1 – Classes of communications

This part of ISO/IEC 9506 emphasizes communications in support of supervisory monitoring and control, or class A communications. Specific features of this communication include, but are not limited to:

- peer to peer communications between computers of various implementations, used for production control and management, and various process control systems;
- access to control and monitoring Blocks and their attributes for the purpose of achieving improved control;
- peer to peer communication with other equipment used in the process control application, such as programmable controllers.

SYSTÈMES D'AUTOMATISATION INDUSTRIELLE – SPÉCIFICATION DE MESSAGERIE INDUSTRIELLE –

Partie 6: Norme d'accompagnement pour la conduite de procédé

1 Domaine d'application

La présente partie de l'ISO/IEC 9506 décrit l'utilisation de la Spécification de Messagerie Industrielle dans l'Environnement de Contrôle-Commande de Procédé, en termes:

- a) d'exigences des interactions d'applications de contrôle commande de procédé;
- b) d'ensemble de modèles abstraits définissant l'interaction entre applications de contrôle commande de procédé;
- c) de fonctionnalité, visible de l'extérieur, des mises en oeuvre conformément à la présente partie de l'ISO/IEC 9506 sous la forme d'exigences de procédures associées à l'exécution des demandes de service;
- d) d'objets qui doivent être visibles de l'extérieur lors de réalisations conformément à la présente partie de l'ISO/IEC 9506 afin de permettre les interactions d'applications de contrôle commande de procédé;
- e) d'exigences de conformité à la présente partie de l'ISO/IEC 9506 qui contient des prescriptions minimales pour des applications simples telles que décrites dans l'article 5.

La présente partie de l'ISO/IEC 9506 spécifie les prescriptions applicables au rôle de serveur, sauf indication contraire.

La présente partie de l'ISO/IEC 9506 concerne l'utilisation entre un calculateur de supervision et tout type d'équipement programmable utilisé en Contrôle-Commande de procédé, y compris les contrôleurs programmables. En outre, elle peut également être utilisée entre deux équipements programmables.

La présente partie de l'ISO/IEC 9506 est une norme d'accompagnement de la Spécification de Messagerie Industrielle, selon l'ISO/IEC 9506-1 et l'ISO/IEC 9506-2, conçue pour permettre la communication de messages vers et à partir des contrôleurs programmables dans un environnement d'applications de Contrôle-Commandes de procédé intégrées. Dans la présente partie de l'ISO/IEC 9506, il est fait référence à cet environnement par les termes Environnement de Contrôle-Commande de Procédé.

La présente partie de l'ISO/IEC 9506 précise une mise en correspondance particulière de l'Environnement de Contrôle-Commande de Procédé sur le modèle d'Équipement Virtuel de Production (VMD) et décrit l'utilisation des services de Messagerie Industrielle dans cet environnement. La présente partie de l'ISO/IEC 9506 prescrit des suppléments aux services de la Spécification de Messagerie Industrielle permettant de répondre aux exigences de l'Environnement de Contrôle-Commande de Procédé et spécifie un protocole supplémentaire pour la prise en charge de ces services.

INDUSTRIAL AUTOMATION SYSTEMS – MANUFACTURING MESSAGE SPECIFICATION –

Part 6: Companion Standard for Process Control

1 Scope

This part of ISO/IEC 9506 describes the use of the Manufacturing Message Specification in the Process Control Environment in terms of:

- a) the interaction requirements of process control applications;
- b) a set of abstract models defining the interaction between process control applications;
- c) the externally visible functionality of implementations conforming to this part of ISO/IEC 9506 in the form of procedural requirements associated with the execution of service requests;
- d) objects required to be externally visible at implementations conforming to this part of ISO/IEC 9506, in order to support the interactions of process control applications;
- e) conformance requirements to this part of ISO/IEC 9506, which include minimum requirements for simple applications as described in clause 5.

This part of ISO/IEC 9506 specifies requirements for the server role, except where specifically noted otherwise.

This part of ISO/IEC 9506 is for use between a supervisory computer and any type of programmable equipment used in process control, including programmable controllers. In addition, it may be used between two instances of programmable equipment.

This part of ISO/IEC 9506 is a companion standard to the Manufacturing Message Specification, ISO/IEC 9506-1 and ISO/IEC 9506-2, designed to support messaging communications to and from programmable devices in an integrated process control application environment. This environment is referred to in this part of ISO/IEC 9506 as the Process Control Environment.

This part of ISO/IEC 9506 specifies a particular mapping of the Process Control Environment onto the Virtual Manufacturing Device (VMD) model and describes the use of the Manufacturing Message Specification services within this environment. This part of ISO/IEC 9506 specifies additions to the services of the Manufacturing Message Specification to support the requirements of the Process Control Environment and specifies additional protocol to support these services.

2 Références normatives

Les documents normatifs suivants contiennent des dispositions qui, par suite de la référence qui y est faite, constituent des dispositions valables pour la présente partie de l'ISO/IEC 9506. Au moment de la publication, les éditions indiquées étaient en vigueur. Tout document normatif est sujet à révision et les parties prenantes aux accords fondés sur la présente partie de l'ISO/IEC 9506 sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des documents normatifs indiqués ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur.

ISO/IEC 646: 1991, *Technologies de l'Information - Jeu ISO de caractères codés à 7 éléments pour l'échange d'Information* (Publiée actuellement en anglais seulement)

ISO/IEC DIS 7498-1: 1992, *Technologies de l'information - Modèle de référence pour l'interconnexion de systèmes ouverts (OSI) - Partie 1: Modèle de référence de base*

ISO TR 8509: 1987, *Systèmes de traitement de l'information - Interconnexion de systèmes ouverts - Conventions de service*

ISO 8649: 1988, *Systèmes de traitement de l'information - Interconnexion de systèmes ouverts - Définition du service pour l'élément de service de contrôle d'association*
Amendement 1: *Authentification pendant l'établissement d'association* (1990)

ISO 8650: 1988, *Systèmes de traitement de l'information - Interconnexion de systèmes ouverts - Spécification du protocole pour l'élément de service de contrôle d'association*
Amendement 1: *Authentification pendant l'établissement d'association* (1990)

ISO/IEC 8824: 1990, *Technologies de l'information - Interconnexion de systèmes ouverts - Spécification de la notation de syntaxe abstraite numéro 1 (ASN.1)* (Publiée actuellement en anglais seulement)

ISO/IEC 9506-1: 1990, *Systèmes d'automatisation industrielle - Spécification de messagerie industrielle - Partie 1: Définition de service* (Publiée actuellement en anglais seulement)

ISO/IEC 9506-2: 1990, *Systèmes d'automatisation industrielle - Spécification de messagerie industrielle - Partie 2: Spécification de protocole* (Publiée actuellement en anglais seulement)

3 Définitions

Pour les besoins de la présente partie de l'ISO/IEC 9506, les définitions suivantes s'appliquent.

3.1 Définitions des modèles de référence

Les articles 3 de l'ISO/IEC 9506-1 et ISO/IEC 9506-2 énumèrent un certain nombre de termes définis dans l'ISO 7498-1, dans l'ISO TR 8509 et dans l'ISO 8824 ainsi que leurs définitions propres. Ces définitions sont incluses par référence dans la présente partie de l'ISO/IEC 9506.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9506. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9506 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 646: 1991, *Information processing – ISO 7-bit coded character set for information interchange*

ISO/IEC DIS 7498-1: 1992, *Information technology – Open Systems Interconnection Reference Model – Part 1: Basic Reference Model*

ISO TR 8509: 1987, *Information processing systems – Open Systems Interconnection – Service conventions*

ISO 8649: 1988, *Information processing systems – Open Systems Interconnection – Service definition for the Association Control Service Element*
Amendment 1: *Authentication during association establishment* (1990)

ISO 8650: 1988, *Information processing systems – Open Systems Interconnection – Protocol Specification for the Association Control Service Element*
Amendment 1: *Authentication during association establishment* (1990)

ISO/IEC 8824: 1990, *Information processing systems – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*

ISO/IEC 9506-1: 1990, *Industrial automation systems – Manufacturing Message Specification – Part 1: Service definition*

ISO/IEC 9506-2: 1990, *Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*

3 Definitions

For the purpose of this part of ISO/IEC 9506, the following definitions apply.

3.1 Reference model definitions

Clause 3 of ISO/IEC 9506-1 and ISO/IEC 9506-2 lists a number of terms defined in ISO 7498-1, in ISO TR 8509, and in ISO 8824 as well as its own definitions. These definitions are included in this part of ISO/IEC 9506 by reference.

3.2 Définitions spécifiques à la présente partie de l'ISO/IEC 9506

Pour les besoins de la présente partie de l'ISO/IEC 9506, les définitions suivantes s'appliquent également.

3.2.1 contrôleur de traitement par lots: Contrôleur utilisé pour réaliser la surveillance et le pilotage de l'élaboration d'un produit qui est fabriqué par lots.

3.2.2 structure de contrôle en cascade: Structure de régulation dans laquelle la variable de sortie d'une boucle de contrôle donnée constitue la variable de référence d'une ou de plusieurs boucles de régulation secondaires.

3.2.3 configuration: Résultat de la personnalisation d'un système de Contrôle-Commande pour l'adapter à un site physique particulier, à une application, à un type d'application ou à une combinaison des éléments ci-dessus.

3.2.4 régulation, contrôle par retour d'état: Algorithme de contrôle où une variable mesurée est comparée à sa valeur désirée afin de générer un signal de commande qui est traité à partir d'un signal d'erreur, de manière à réduire l'amplitude de l'erreur.

3.2.5 boucle de régulation: Ensemble des équipements d'Entrée/Sortie, des algorithmes et des autres modules matériels et logiciels utilisés pour mettre en oeuvre la fonction de régulation.

3.2.6 Contrôle-Commande de supervision: Algorithme de contrôle consistant à ajuster de façon intermittente les boucles de régulation qui fonctionnent de manière indépendante; par exemple, modifications du point de consigne à partir d'une source extérieure.

3.2.7 système de Contrôle-Commande (SCC): Système où une manipulation ou une commande délibérée est utilisée pour obtenir la valeur prescrite d'une variable donnée.

3.2.8 contrôleur: Dispositif qui fonctionne automatiquement pour réguler une variable contrôlée.

3.2.9 contrôleur PID: Contrôleur qui utilise une régulation proportionnelle, intégrale, dérivée.

3.2.10 dispositif: Appareil chargé d'exécuter une fonction prescrite.

3.2.11 élément: Composant d'un dispositif ou d'un système.

3.2.12 matériel: Equipement physique directement impliqué dans l'exécution de fonctions de mesures et de Contrôle-Commande de procédés industriels.

3.2.13 historique: Système dont la fonction est d'archiver les enregistrements des actions effectuées au sein de systèmes de Contrôle-Commande de procédés.

3.2.14 dispositif de liaison: Dispositif assurant une fonction de passerelle entre deux réseaux dissemblables; par exemple, entre le réseau principal et le réseau de Contrôle-Commande ou entre le réseau principal et le Réseau de terrain.

3.2.15 contrôleur monoboucle: Contrôleur assurant le fonctionnement d'une boucle de régulation.

3.2 Definitions unique to this part of ISO/IEC 9506

For the purpose of this part of ISO/IEC 9506, the following definitions also apply:

3.2.1 batch control computer: A computer used to monitor and control the execution of a product that is manufactured in discrete batches.

3.2.2 cascade structure: A control structure in which the output variable of one controller is the reference variable for one or more secondary control loops.

3.2.3 configuration: The result of customizing a general control system for a particular physical location, application, class of application, or any combination thereof.

3.2.4 control, feedback: Control in which a measured variable is compared to its desired value to produce an actuating error signal that is acted upon in such a way as to reduce the magnitude of the error.

3.2.5 control loop: The collection of I/O equipment, algorithms and other hardware and software modules used to implement feedback control.

3.2.6 control, supervisory: Control in which the control loops operate independently subject to intermittent corrective action; e.g. set point changes from an external source.

3.2.7 control system: A system in which deliberate guidance or manipulation is used to achieve a prescribed value of a variable.

3.2.8 controller: A device that operates automatically to regulate a controlled variable.

3.2.9 controller, PID: A controller that produces proportional plus integral (reset) plus derivative (rate) control.

3.2.10 device: An apparatus for performing a prescribed function.

3.2.11 element: A component of a device or system.

3.2.12 hardware: Physical equipment directly involved in performing industrial process measuring and controlling functions.

3.2.13 historian: A system whose function is to archive records of the actions taken within the process control system.

3.2.14 linking device: A device which provides a gateway function between two dissimilar networks; for example, between the backbone and the control network, or between the backbone and the Fieldbus.

3.2.15 loop controller: A controller which operates a control loop.

3.2.16 contrôleur multiboucle: Contrôleur assurant le fonctionnement de plusieurs boucles de régulation.

3.2.17 console opérateur: Un système chargé de la supervision ou de la commande d'un procédé ou les deux à la fois. Une console opérateur comprend l'objet Station Opérateur de l'ISO/IEC 9506-1. Elle est en général capable d'afficher des vues graphiques en couleur avec mise à jour des données en temps réel.

3.2.18 paramètre: Grandeur ou caractéristique considérée comme une constante mais qui peut évoluer ou être ajustée.

3.2.19 procédé: Transformation physico-chimique de matière ou conversion d'énergie; par exemple, modification de pression, de température, de vitesse, de potentiel électrique, etc.

3.2.20 Contrôle-Commande de procédé: Régulation ou commande de variables influençant le comportement d'un procédé pour obtenir de manière efficace un produit avec la qualité et la quantité désirées.

3.2.21 application de surveillance de procédé: Application qui peut assurer l'une des fonctions suivantes: observation des tendances du procédé, lancement d'actions permettant d'optimiser le procédé, implication d'une technologie de système expert.

3.2.22 action réflexe: Voir traitement réflexe.

3.2.23 traitement réflexe: Action exécutée sans consulter un opérateur ou le calculateur de supervision, conditionnée généralement par l'apparition d'une condition ou d'un événement spécifique.

3.2.24 régulation: Synonyme de contrôle pour retour d'état. (Voir 3.2.4)

3.2.25 mise en forme du signal: Action portant sur un signal pour le rendre utile au dispositif qui l'utilisera; par exemple, suppression du bruit et inversion du signal.

3.2.26 inversion de signal: Traitement du signal où le signe (+,-) du signal est «inversé» sur l'autre signe.

4 Abréviations

CBB (Conformance Building Block) Bloc élémentaire de conformité

SNCC (Distributed Process Control System) Système de SNCC

MMS (Manufacturing Message Specification) Spécification de Messagerie Industrielle

PDU (Protocol Data Unit) Unité de données de protocole

PICS (Proforma Implementation Conformance Statement) Déclaration pro forma de conformité de réalisation

PID (Proportional Integral Derivative) Proportionnelle, Intégrale, Dérivée

CC (Programmable Controller) Contrôleur Programmable

SCC (Process Control System) SCC

VMD (Virtual Manufacturing Device) Equipement virtuel de production

3.2.16 multiloop controller: A controller which operates more than one control loop.

3.2.17 operator's console: A system for the monitoring or operation of a process, or both. An operator's console includes the Operator Station object of ISO/IEC 9506-1. It is usually capable of colour graphic displays with live updating data.

3.2.18 parameter: A quantity or property treated as a constant but which may sometimes vary or be adjusted.

3.2.19 process: Physical or chemical change of matter or conversion of energy; e.g. change in pressure, temperature, speed, electrical potential, etc.

3.2.20 process control: The regulation or manipulation of variables influencing the conduct of a process in such a way as to obtain a product of desired quality and quantity in an efficient manner.

3.2.21 process monitoring application: An application that may provide any of the following functions: observe process trends, initiate actions for process optimization, involve the use of expert system technology.

3.2.22 reflexive action: See reflexive processing.

3.2.23 reflexive processing: Action taken without consultation with an operator or supervisory computer, usually conditioned by the occurrence of specified condition or event.

3.2.24 regulatory control: A synonym for feedback control. (See 3.2.4.)

3.2.25 signal conditioning: The act of operating on a signal to make it useful to the device which will utilize the signal. Examples include noise suppression and inversion.

3.2.26 signal inversion: A process in signal conditioning in which the sign of the signal (+, -) is "inverted" to the other sign.

4 Abbreviations

CBB	Conformance Building Block
DCS	Distributed Process Control System
MMS	Manufacturing Message Specification
PDU	Protocol Data Unit
PICS	Proforma Implementation Conformance Statement
PID	Proportional Integral Derivative
PC	Programmable Controller
PCS	Process Control System
VMD	Virtual Manufacturing Device

5 Description de l'application

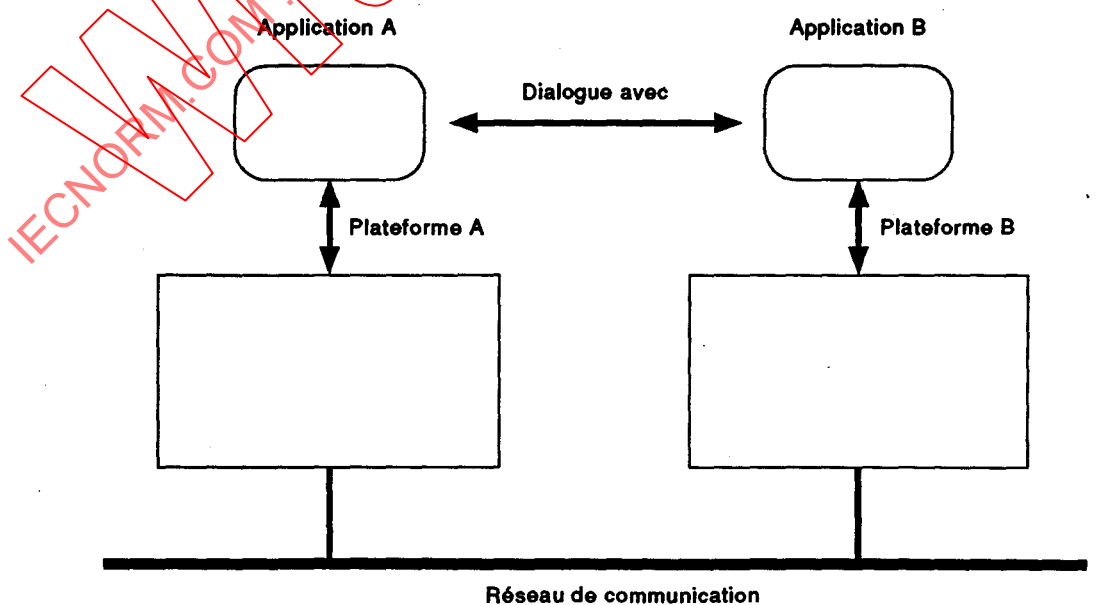
Les interactions entre applications dans l'Environnement de Contrôle-Commande de Procédé apparaissent entre des homologues qui communiquent pour échanger des informations de Contrôle-Commande ou de supervision ou les deux à la fois. Au cours de la durée de vie de tous les exemples d'associations d'application, un système réel peut adopter le rôle client ou le rôle serveur ou les deux à la fois. La présente partie de l'ISO/IEC 9506 ne définit aucune restriction quant au comportement du client, autre que la prescription implicite selon laquelle le système jouant le rôle du client doit être capable d'émettre des demandes de service appropriées et de recevoir des réponses adéquates. Dans les environnements très simples, une simple association d'application peut suffire entre deux homologues communicants. Dans des environnements plus complexes, plusieurs associations d'application peuvent s'avérer nécessaires.

5.1 Modèles de Contrôle-Commande de procédé

5.1.1 Modèles de communication industrielle de procédé

5.1.1.1 Modèles de communication entre homologues

La figure 2 illustre la communication entre homologues qui peut être appliquée entre un client et un serveur collaborant dans un Environnement de Contrôle-Commande de Procédé afin d'optimiser le comportement du système de Contrôle-Commande. Il est admis d'appliquer ce modèle à la communication entre une application de traitement par lots et un équipement de surveillance et de Contrôle-Commande ou entre une application de surveillance de procédé et un grand système de Contrôle-Commande de procédé réparti où le système de Contrôle-Commande de procédé réparti est considéré comme un seul contrôleur programmable comportant de nombreux éléments de surveillance et de Contrôle-Commande. Ce modèle peut être utilisé entre deux systèmes de Contrôle-Commande de procédé répartis, chacun de ces systèmes étant considéré comme un seul contrôleur et peut également être appliqué aux interactions entre éléments d'un système de Contrôle-Commande de procédé réparti ou entre éléments de Contrôle-Commande répartis sur un bus de communication de Contrôle-Commande de procédé.



CEI 407194

Figure 2 – Interaction dans un environnement d'homologues

5 Application description

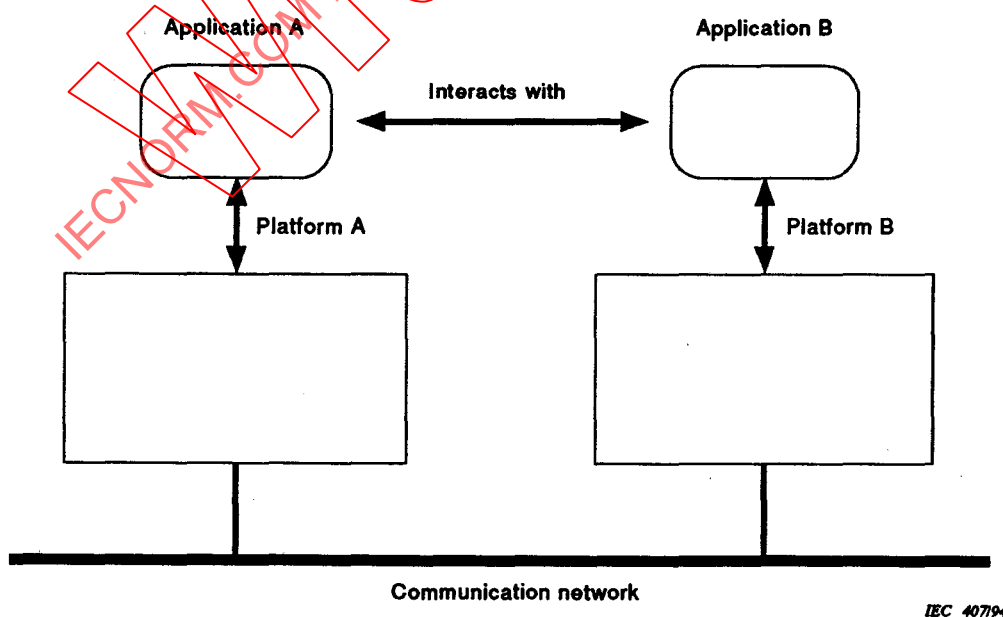
The interactions between applications in the Process Control Environment take place between communicating peers in support of the exchange of control information, or monitoring information, or both. During the lifetime of any instance of an application association, a real system may adopt the client role, or the server role, or both. This part of ISO/IEC 9506 places no restrictions on the behaviour of a client, other than the implied requirement that the system acting in the client role be capable of issuing appropriate requests and receiving appropriate responses. In very simple environments, a single application association may suffice between two communicating peers. In more complex environments, there may be a necessity for more than one application association.

5.1 Process control models

5.1.1 Process manufacturing communication models

5.1.1.1 Peer to peer communication model

Figure 2 illustrates peer to peer communication which may be applied between a cooperating client and server in the Process Control Environment for the purpose of optimizing behaviour in a control system. This model may be applied to communications between a batch control application and a monitoring and control device or between a process monitoring application and a large, distributed process control system, where the distributed process control system is treated as a single device containing numerous monitoring and control elements. This model may be used between two distributed process control systems, each treated as a single device, and may also be applied to interactions between elements of a distributed process control system or between control elements distributed on a control communications highway.



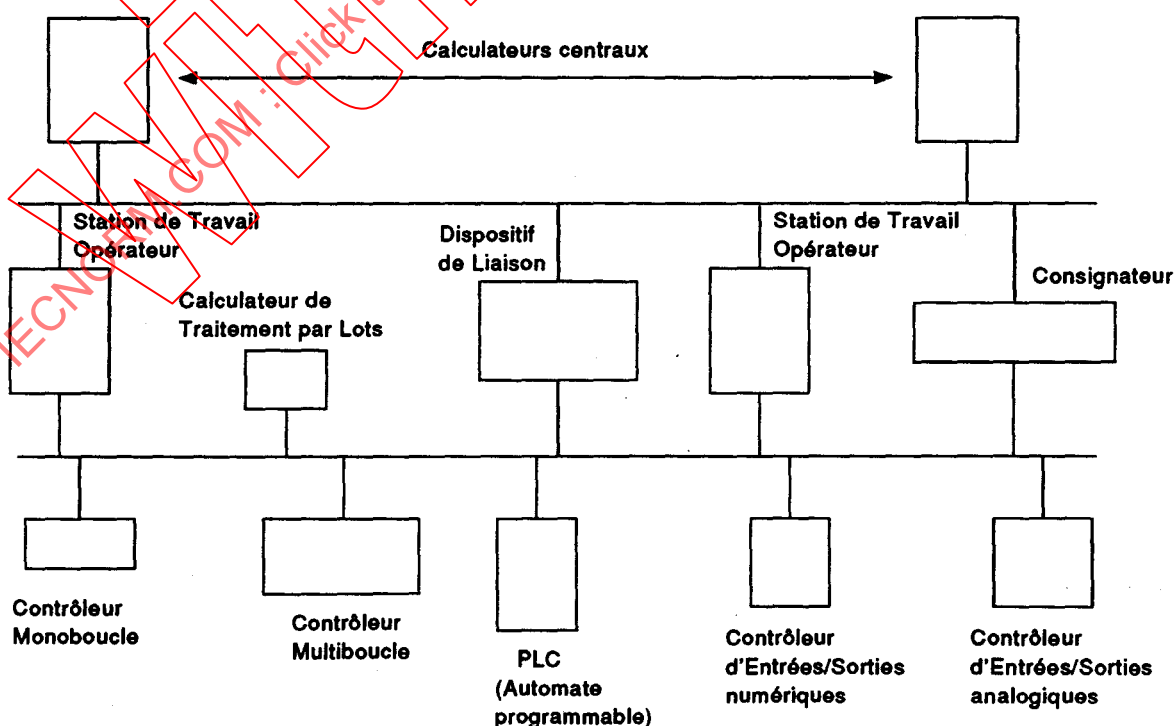
IEC 40794

Figure 2 – Interaction in a peer to peer environment

5.1.1.2 Exemple d'une configuration à plusieurs étages

La figure 3 illustre un exemple de configuration matérielle qui intègre un système de Contrôle-Commande de procédé avec un ou plusieurs calculateurs centraux. Le réseau interne du système de Contrôle-Commande de procédé réparti est représenté par la ligne horizontale inférieure et ses éléments constitutifs sont illustrés dans les cases encadrées reliées au réseau interne. Les éléments illustrés sur la figure sont représentatifs de ceux que l'on rencontre généralement dans des systèmes de Contrôle-Commande répartis et comprennent des contrôleurs d'entrée/sortie analogiques et numériques, des contrôleurs mono- ou multiboucle, un contrôleur programmable (PC), des calculateurs de traitement par lots, un consigneur, des postes de travail opérateurs et un dispositif (de liaison) qui assure une fonction de connexion (virtuellement, une passerelle) entre le réseau interne du système de Contrôle-Commande de procédé réparti et le réseau de supervision. Le réseau de supervision est représenté par la ligne horizontale supérieure et les calculateurs centraux sont reliés au réseau de supervision.

Dans cet exemple, une application de Contrôle-Commande de Production fonctionnant sur un ordinateur central peut être utilisée pour télécharger des recettes et récupérer des données produits. Le calculateur central peut également communiquer avec les consoles opérateurs pour configurer les affichages ou fournir des instructions aux opérateurs. Le calculateur central ou la console opérateur peuvent également récupérer les données à partir du serveur d'historique. Dans ce cas, le consigneur peut agir comme client en recueillant des données en provenance des contrôleurs et comme serveur en fournissant des données au calculateur central ou à la console opérateur. Lorsque les caractéristiques spécifiques de l'application le nécessitent, certains équipements peuvent être munis de connexions à la fois avec le réseau de Contrôle-Commande et le réseau de supervision. Une telle architecture pourrait par exemple comprendre un consigneur et une console opérateur qui peuvent s'appuyer d'une part sur le réseau de Contrôle-Commande pour l'acquisition de données et d'autre part sur le réseau de supervision pour des communications ayant trait à la supervision.



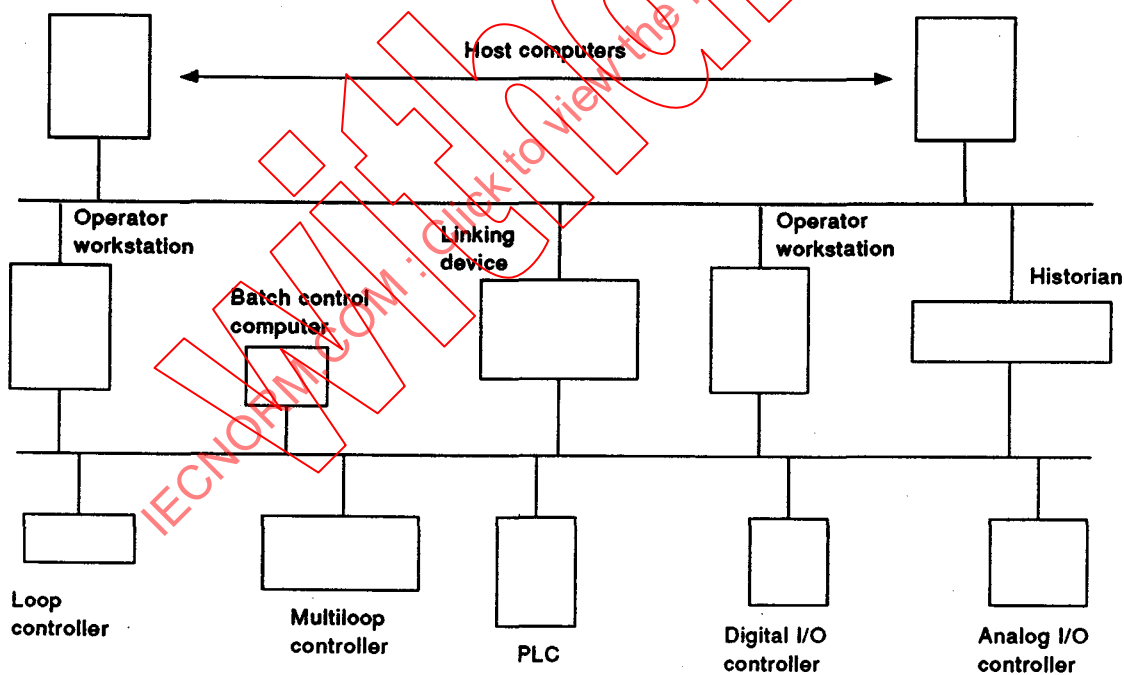
CEI 408/94

Figure 3 – Exemple d'une configuration matérielle

5.1.1.2 Multi-tiered example

Figure 3 shows an example of a hardware configuration which integrates a distributed process control system with one or more host computers. The distributed process control system's internal network is represented by the lower horizontal line, and its component elements are shown as boxes attached to the internal network. The elements shown are representative of those commonly found in distributed control systems, and include analog and digital input/output controllers, single and multiloop controllers, a programmable controller (PC), batch control computers, a historian, operator workstations and a device (linking device) which performs a linking (potentially a gateway) function between the distributed process control system's internal network and the supervisory network. The supervisory network is represented by the upper horizontal line, and the host computers are attached to the supervisory network.

In this example, a Production Control application operating on a host computer may be used to download batch recipes and retrieve batch product data. The host computer may also communicate with the operator's consoles to setup displays or give operator instructions. The host computer or operator's console may also recall the data from the historian. In this case the historian may act as a client when collecting data from controllers, and as a server when providing data for the host computer or the operator's console. Where application circumstances warrant, some equipment may be provided with connections to both the control network and the supervisory network. One example of such an architecture might include a historian and an operator's console, which may rely on the control network for data acquisition, and may additionally rely on the supervisory network for communications of a supervisory nature.



IEC 408/94

Figure 3 – Example of hardware configuration

5.2 Fonctions de Contrôle-Commande de procédé

5.2.1 Support de régulation

5.2.1.1 Généralités

Dans l'Environnement de Contrôle-Commande de Procédé, la régulation est assurée par l'utilisation de Blocs fonctionnels tels que le conditionnement des signaux, les conversions mathématiques, les caractérisations X-Y et les boucles de régulation PID (Proportionnelle, Intégrale, Dérivée). Ces blocs fonctionnels représentent des réalisations de fonctions dans un système de Contrôle-Commande de procédé réel.

Il est admis de combiner des fonctions simples pour constituer des fonctions complexes en reliant la sortie d'un Bloc à l'entrée d'un autre Bloc. Chaque Bloc est associé à un ensemble de paramètres de Bloc qui représentent les entrées, les sorties et les variables d'état internes requises par la fonction.

5.2.1.2 Protection

Pendant le fonctionnement du système de Contrôle-Commande de procédé, les blocs sont sujets à modification. Ces modifications peuvent prendre la forme d'un réglage des paramètres des blocs pour un fonctionnement optimal. Il est nécessaire de protéger les Blocs contre toute modification non autorisée; pour cela, il faut que l'identité et le privilège du demandeur soient déterminés avant la modification du Bloc. Dans le cadre de cette identification, un demandeur doit être classé selon qu'il s'agit d'un opérateur humain, d'un programme informatique, d'une sollicitation de Bloc distant ou d'un noeud de réseau distant.

5.2.1.3 Groupes de Blocs

Etant donné qu'en général un système de Contrôle-Commande de procédé typique utilise un grand nombre d'instances de Blocs, il est nécessaire de disposer d'une gestion efficace des groupes de Blocs. Ceci comprend les moyens de téléchargement et de télé-sauvegarde des blocs, de démarrage et d'arrêt des groupes de Blocs et autres traitements de ces blocs par groupes.

5.2.2 Gestion des événements et des alarmes

5.2.2.1 Généralités

Du point de vue fonctionnel, il est nécessaire de surveiller en permanence le procédé contrôlé et d'identifier, en tant qu'«événements procédé», l'occurrence de certaines conditions relatives à l'état du procédé en cours. La gestion d'un événement procédé implique la faculté de communiquer l'événement correspondant à une application distante. Il est également nécessaire du point de vue fonctionnel de créer une «demande de notification», soit par, soit au nom d'un homologue communicant qui entraînera la création et l'émission d'un message de notification vers l'homologue communicant, ledit message contenant des informations relatives au passage à la condition et à l'état procédé spécifiés.

Lorsque l'occurrence d'un événement est communiquée, il est nécessaire du point de vue fonctionnel d'inclure des chaînes de caractères ou des numéros d'identification dans le message communiquant l'occurrence. Selon les caractéristiques spécifiques de l'application, le contenu de la chaîne ou du numéro d'identification peut être identique ou différent pour chaque occurrence rapportée. Ces informations sont désignées par les termes «enrichissements d'affichage».

5.2 *Process control functions*

5.2.1 *Support for regulatory control*

5.2.1.1 *General*

In the Process Control Environment, regulatory control is conceptually accomplished through the utilization of Blocks, such as signal conditioning, mathematical conversions, X-Y characterizations and Proportional-Integral-Derivative (PID) control loops. Blocks represent real instances of functions in a real process control system.

Simple functions may be combined to form complex functions by connecting the output of one Block to the input of another Block. Each Block has associated with it a set of Block parameters which represent the inputs, outputs and internal state variables required by the function.

5.2.1.2 *Protection*

Blocks are subject to alteration during operation of the process control system. This alteration may take the form of tuning block parameters for optimal operation. It is a requirement that Blocks be protected against unauthorized modification; in order to accomplish this, it is necessary that the identity and privilege of the requester be determined before a Block is modified. As part of this identification, a requester needs to be categorized as to whether it is a human operator, a computer program, a remote Block instance, or a remote network node.

5.2.1.3 *Groups of Blocks*

Since, in general, there will be a large number of Block instances employed in a typical process control system, it is a requirement that there be an efficient way of handling groups of Blocks. This includes facilities for downloading and uploading Blocks, starting and stopping groups of Blocks, and otherwise dealing with them in groups.

5.2.2 *Management of events and alarms*

5.2.2.1 *General*

There is a functional requirement to continuously monitor processes under control, and identify as "process events" the occurrence of certain conditions related to the state of the process under control. Management of a process event implies an ability to communicate the occurrence of the process event to a remote application. There is a functional requirement for the creation of a "notification request", either by or on behalf of a communicating peer, which will result in the creation and transmission of a notification message to the communicating peer containing information about the transition into the specified condition and process state.

When the occurrence of an event is communicated, there is a functional requirement to allow inclusion of textual strings, or identifying numbers, in the message communicating the occurrence. Depending on application considerations, the content of the string or identifying number may be identical for each reported occurrence, or different for each reported occurrence. This information is referred to as "display enhancements".

NOTE – Il a été prévu d'inclure à la fois des nombres et des chaînes de caractères afin de satisfaire aux exigences qui prescrivent l'affichage de textes directement sur la console opérateur, ou l'utilisation de valeurs numériques pour indexer un tableau contenant des chaînes multiples dans des langues différentes.

Dans certaines circonstances déterminées par une application ou la forme prise par un processus, la réaction à un événement de ce processus peut comporter des «actions réflexes» appelées à être exécutées sous la forme d'actions autonomes qui résultent de la détection de l'événement du processus. Il est de règle de communiquer les résultats de ces actions réflexes par transmission à des homologues désireux de connaître ces résultats. Au titre de l'efficacité, les résultats des actions réflexes sont communiqués avec le compte-rendu d'occurrence de l'événement du processus. Il faudra alors prévoir une demande de notification spécifiant la demande de traitement réflexe.

EXEMPLES

- 1 Une action réflexe peut être un simple acte comme, par exemple, le lancement d'une Invocation de Programme qui peut donner lieu à plusieurs actions complexes gérées par le système.
- 2 Une action réflexe peut être une mesure qui doit être prise dans l'intérêt de la sécurité du personnel ou de l'équipement comme, par exemple, le déversement du contenu d'un réservoir d'extinction dans une réaction chimique exothermique qui s'emballe.

Lorsque les événements procédé doivent être acquittés par une entité distante, qu'il s'agisse d'un opérateur humain ou d'une application de supervision, l'événement est appelé «alarme» et du point de vue fonctionnel, il est nécessaire de disposer des techniques correspondantes de gestion d'alarmes telles qu'affichage, enregistrement et acquittement.

5.2.2.2 Regroupements d'événement et d'alarmes

Dans certaines applications de Contrôle-Commande de procédé, il est nécessaire du point de vue fonctionnel de pouvoir grouper des événements procédé en «Groupes d'événements procédé», à des fins de consultation des états et de gestion des alarmes. Dans la plupart des cas, ces groupes sont associés à une section de procédé ou un ensemble d'équipements ou, dans d'autres cas, représentent des regroupements de groupes ou un niveau de rapports hiérarchiques de groupes. Il est nécessaire du point de vue fonctionnel que les rapports hiérarchiques entre groupes soient préservés tout en permettant les modifications de groupes à des niveaux hiérarchiques inférieurs pour les repercuter dans des groupes de niveau supérieur.

EXEMPLE

Il conviendrait que l'ajout d'un capteur de température à un groupe de pompes d'alimentation soit automatiquement pris en compte lorsque les éléments de groupes relatifs à la chaudière sont appelés.

Il est également prescrit du point de vue fonctionnel de pouvoir modifier et rétablir de façon dynamique le niveau de priorité des groupes d'événements procédé.

NOTE – Cette faculté est utile lors du démarrage et de l'arrêt du procédé, ainsi que pour la maintenance de l'installation, car elle permet de gérer les groupes de conditions événementielles et d'alarmes en fonction des unités de l'installation affectée. La priorité des événements individuels est décrite dans l'ISO/IEC 9506-1.

5.2.2.3 Masquage d'alarmes

Lors des périodes de maintenance, lors du démarrage ou de mise à l'arrêt non prévue du procédé, il est nécessaire du point de vue fonctionnel de masquer les notifications d'événements procédé qui notifient des conditions qui sont délibérément anormales. Il est également nécessaire du point de vue fonctionnel de disposer de la faculté de masquer groupes d'événements procédé correspondant aux unités d'installation ou d'équipement affectées. Lorsque les conditions sont stables, il est en outre nécessaire du point de vue fonctionnel que les compte-rendus d'événements et d'alarmes retournent à la normale.

NOTE – Provision has been included for both numbers and text strings due to the requirement to be able to display the string directly on an operator's console, or to use the number to index into an array containing multiple strings in different languages.

Under certain circumstances determined by an application or a configuration process, the response to a process event may include "reflexive actions" which are executed as autonomous actions, as a direct result of detection of the process event. A requirement exists to communicate the results of the reflexive action to communicating peers which have a desire to be aware of the results. As an efficiency move, the results of reflexive actions are communicated with the report of the occurrence of a process event. The creation of a notification request will be used to specify a request for reflexive processing.

EXAMPLES

- 1 A reflexive action may be a simple act, such as the starting of a Program Invocation, which may lead to several complex actions managed by the system.
- 2 A reflexive action may be an action which must be taken in the interests of personal safety or safety to equipment. The dumping of the contents of a quench tank into an exothermic chemical reaction which is running away constitutes another example.

When process events require acknowledgement by a remote entity, whether a human operator or a supervisory application, the event is termed an "alarm," and there is a functional requirement for related alarm management techniques such as display, recording and acknowledgement.

5.2.2.2 Event and alarm groupings

In certain process control applications, there is a functional requirement for a capability to group process events into "Process Event Groups", for purposes of status inquiry and alarm management. In most cases, such groups are associated with a unit of equipment or plant, or in some cases represent a grouping of groups, or an instance of a hierarchical relationship of groups. It is a functional requirement that the hierarchical relationships between groups be preserved, while allowing changes of groups at lower levels in the hierarchy to be reflected in higher level groups.

EXAMPLE

The addition of a temperature sensor to a feedwater pump group should be automatically included when the points in the boiler support groups are requested.

An additional function requirement is a capability to dynamically override the priority of individual events in process event groups.

NOTE – This capability has utility during process start-up and shutdown, as well as facility maintenance, where groups of event and alarm conditions are managed in correspondence to units of affected plant. Priority of individual events is described in ISO/IEC 9506-1.

5.2.2.3 Alarm suppression

During maintenance periods, process start-up or abnormal shutdown, there is a functional requirement to suppress notifications of process events which are reporting conditions known to be abnormal. There is a functional requirement for a capability to perform event and alarm suppression on groups of process events, corresponding to affected units of plant or equipment. When conditions are stable, an additional functional requirement exists to return event and alarm reporting to normal.

5.2.3 *Besoins pour la fabrication par lots*

Dans de nombreuses applications de l'industrie de procédé, la production s'effectue par «lots», c'est-à-dire des ensembles de grandes quantités de matériaux à transformer en produits finis. De manière générale, un calculateur - le calculateur de traitement par lots - contiendra le schéma général de cette fabrication par lots et le procédé sera piloté sous le contrôle d'un ou de plusieurs calculateurs de Contrôle-Commande de procédé. L'examen des interactions entre le calculateur de traitement par lots et les calculateurs de Contrôle-Commande de procédé fait ressortir cinq types de besoins.

5.2.3.1 *Détermination de l'état d'un traitement par lots*

Le calculateur de traitement par lots doit être capable de déterminer l'état du procédé de traitement par lots ou l'état des parties constituantes du procédé de traitement par lots tel qu'il est lu au niveau du calculateur de Contrôle-Commande de procédé.

5.2.3.2 *Contrôle-Commande d'un traitement par lots*

Le calculateur de traitement par lots doit être capable de piloter le procédé de traitement par lots en modifiant l'état des applications de Contrôle-Commande fonctionnant dans le calculateur de Contrôle-Commande de procédé.

5.2.3.3 *Modification du Contrôle-Commande d'un procédé de traitement par lots*

Dans certains cas, il peut être nécessaire à un opérateur humain présent au niveau du calculateur de Contrôle-Commande de procédé de prendre la main pour piloter lui-même le procédé. Dans ce cas, il est nécessaire que le calculateur de Contrôle-Commande de procédé doit suspendre sa fonction de pilotage selon une procédure particulière.

5.2.3.4 *Communication directe avec un opérateur humain*

Dans certains cas, il peut être nécessaire au calculateur de Contrôle-Commande de procédé par lots d'afficher des informations destinées à un opérateur humain présent au niveau du calculateur de Contrôle-Commande de procédé pour lui demander d'entrer des données.

5.2.3.5 *Synchronisation et coordination*

Le calculateur de Contrôle-Commande de procédé peut avoir besoin de coordonner l'utilisation de ressources avec d'autres procédés dans l'Environnement de Contrôle-Commande de Procédé. Ces ressources peuvent être une source commune d'alimentation en matières premières ou un périphérique partagé comme, par exemple, une imprimante de consignation d'erreurs.

5.2.4 *Besoins pour le stockage et la récupération d'historiques*

Il est nécessaire du point de vue fonctionnel de conserver les informations concernant un ou des procédés particuliers. Il convient que les informations soient disponibles en ordre chronologique lorsqu'elles sont récupérées. Les informations qui sont en général mémorisées dans les «historiques» comprennent des événements procédé datés, les historiques de variables de procédé importantes, enregistrées soit conjointement aux événements de procédé, soit indépendamment d'eux, ainsi que les commentaires saisis par l'opérateur humain.

5.2.3 *Support for batch manufacturing*

In many process industry applications, production occurs in "batches", i.e. aggregations of large quantities of material to be made into final products. Typically, one computer, the "batch control computer", will contain the master plan for this batch manufacture, and the process will be carried out under the control of one or more process control computers. In dealing with the interactions between the batch control computer and the process control computers, five types of requirements can be identified.

5.2.3.1 *Determination of the status of a batch process*

The batch control computer needs to be able to determine the status of the batch process or the status of the components of the batch process as sensed at the process control computer.

5.2.3.2 *Control of a batch process*

The batch control computer needs to be able to control the batch process by changing the state of the control applications operating in the process control computer.

5.2.3.3 *Alteration of the control of a batch process*

In some cases, it may be necessary for a human operator present at the process control computer to assume control of the process. In such circumstances, there is need for the batch control computer to relinquish control in an orderly way.

5.2.3.4 *Direct communication with a human operator*

In some cases, it may be necessary for the batch control computer to display information to a human operator present at the process control computer to solicit his input.

5.2.3.5 *Synchronization and coordination*

The batch control computer may need to coordinate the use of resources with other processes in the Process Control Environment. Such resources may be a common source of supply of raw materials or a shared device such as an error logging printer.

5.2.4 *Support for creation and retrieval of historical records*

There is a functional requirement for the retention of information related to a particular process or processes. The information should be available in chronological order when retrieved. Information typically stored in "historical records" includes time-stamped records of process events, time-stamped records of process variables of interest, either recorded in conjunction with records of process events or independently, and comments entered by a human operator.

5.2.5 *Besoins pour la configuration et la gestion du système de Contrôle-Commande de procédé*

Il est prescrit du point de vue fonctionnel de transférer des données de configuration d'un calculateur central vers un système de Contrôle-Commande de procédé opérationnel ou non configuré; de même, il est prescrit du point de vue fonctionnel de pouvoir récupérer des données de configuration pour stockage hors système après mise au point (dé-bogage). Les données de configuration comprennent, de manière non exhaustive, le code programme, les tables de données ainsi que des corrections de code.

5.3 *Modèles d'Application*

5.3.1 *Bloc*

Le Bloc représente une réalisation [instance] de fonction de Contrôle-Commande ou de surveillance d'un système de Contrôle-Commande de procédé. Les aspects fonctionnels d'un Bloc, nécessaires pour spécifier son comportement en termes de communication, sont décrits dans ce paragraphe. D'autres informations relatives aux Blocs et à leur utilisation dans l'Industrie de procédé sont fournies en annexe B.

EXEMPLE:

Les Blocs peuvent être utilisés pour le conditionnement des signaux, des fonctions de calcul mathématiques ou complexes ainsi que pour des fonctions de surveillance.

Modèle: Bloc

Attribut-Clé: Etiquette de Bloc
 Attribut: EnService (VRAI, FAUX)
 Attribut: Nom Algorithme
 Attribut: Mode-Etat
 Attribut: Liste des Paramètres de Bloc

5.3.1.1 *Etiquette de Bloc (Block Tag)*

Cet attribut est le nom du Bloc. Il est unique pour l'ensemble du système de Contrôle-Commande de procédé. Il est attribué par l'utilisateur final ou le gestionnaire de configuration du système de Contrôle-Commande de procédé.

5.3.1.2 *EnService (InService)*

Cet attribut indique si le fonctionnement du Bloc est actif (VRAI) ou non (FAUX).

5.3.1.3 *Nom Algorithme*

Cet attribut permet d'identifier l'algorithme fonctionnel utilisé. Chaque Bloc dans un système de Contrôle-Commande réel se réfère à un algorithme particulier; cependant, l'utilisation d'un algorithme commun pour des multiblocs n'est pas exclue à condition que les références individuelles de chaque Bloc soient maintenues.

5.3.1.4 *Mode-Etat (Mode-State)*

Cet attribut régit le fonctionnement du Bloc et la protection de ses éléments. D'autres informations sont fournies en annexe B.

5.2.5 Support for process control system configuration and management

A functional requirement exists for the transfer of configuration data from a host computer into an unconfigured as well as into an operational process control system; conversely, a functional requirement exists for a capability to retrieve configuration data for off-system storage following debugging. Configuration data includes, but is not limited to, program code, data tables and code patches.

5.3 Application Models

5.3.1 Block

The Block conceptually represents an installed instance of a control or monitoring function in a process control system. The basic functional aspects of a Block necessary for prescribing communication behaviour are described here. Further information about Blocks and their use in the process industry may be found in annex B.

EXAMPLE:

Blocks may be utilized to perform signal conditioning, mathematical or complex monitoring and control functions.

Model: Block

Key Attribute: Block Tag

Attribute: InService (TRUE, FALSE)

Attribute: Algorithm reference

Attribute: Mode-State

Attribute: List of Block Parameters

5.3.1.1 Block Tag

This attribute is the name of the Block. It is unique throughout the process control system. It is assigned by the end user or configuration manager of the process control system.

5.3.1.2 InService

This attribute indicates whether (TRUE) or not (FALSE) the Block is actively operating.

5.3.1.3 Algorithm Reference

This attribute contains a reference to the function algorithm. There is a reference for each Block in a real control system; however, use of a common algorithm for multiple Blocks is not precluded so long as individual references from each Block are maintained.

5.3.1.4 Mode-State

This attribute governs the operation of the Block and the protection of its elements. Further information can be found in annex B.

5.3.1.5 *Liste des Paramètres du Bloc*

Cet attribut est la liste des paramètres du Bloc qui représente les entrées, les sorties ainsi que les variables d'état internes du Bloc.

5.3.2 *Algorithme*

L'objet Algorithme représente l'algorithme utilisé par un Bloc donné pour transformer ses entrées en sorties. Il s'agit d'un objet séparé destiné à refléter le fait que plusieurs Blocs peuvent utiliser un algorithme commun.

Objet: Algorithme

Attribut-Clé: Nom de l'Algorithme

Attribut: Contenu de l'Algorithme

5.3.2.1 *Nom de l'Algorithme*

Cet attribut est le nom associé à l'algorithme.

5.3.2.2 *Contenu de l'Algorithme*

Cet attribut représente la substance de l'algorithme, le programme ou les règles permettant de déterminer la sortie sur la base de l'entrée et de l'état interne.

6 *Mappage du contexte de Contrôle-Commande de procédé*

Le présent article prescrit la manière dont les ressources de la MMS doivent être utilisées pour satisfaire aux besoins de communication de l'Environnement de Contrôle-Commande de Procédé décrit dans l'article précédent. Le présent article fait également le lien entre les modèles de systèmes de Contrôle-Commande de procédé et le modèle abstrait d'Équipement Virtuel de Production (VMD) décrit dans l'ISO/IEC 9506-1.

6.1 *Mappage du modèle de Contrôle-Commande de procédé par rapport au VMD*

Chaque dispositif d'un système de Contrôle-Commande de procédé offrant des capacités de communication de Classe A en tant que serveur, doit être mappé sur au moins un VMD.

EXEMPLE

Un contrôleur programmable engagé dans une application de Contrôle-Commande de procédé peut être représenté par un seul VMD. Un système de Contrôle-Commande de procédé réparti, vu du point de vue du calculateur de Contrôle-Commande de production par le biais d'une passerelle, peut également comprendre un seul VMD. Un système de Contrôle-Commande de procédé réparti, vu du point de vue d'un élément interne du système de Contrôle-Commande, tel qu'une console opérateur, pourrait comporter plusieurs VMD, avec au moins un VMD représentant chaque station reliée au système de Contrôle-Commande de procédé réparti. Un grand système de Contrôle-Commande de procédé réparti, avec de multiples connexions au réseau de supervision par le biais de la passerelle système, de l'historique et de la console opérateur, pourrait être considéré du point de vue du réseau de supervision comme des VMD multiples.

6.2 *Définition des objets de Contrôle-Commande de procédé mappant des Domaines*

6.2.1 *L'objet Bloc*

L'objet Bloc doit mapper un Domaine et une Invocation de Programme liés à ce Domaine. Ce Domaine doit contenir des objets Variables Nommées, spécifiques au Domaine, qui

5.3.1.5 *List of Block Parameters*

This attribute is the list of Block parameters which represents the inputs, outputs and internal state variables of the Block.

5.3.2 *Algorithm*

The Algorithm object represents the algorithm used by a Block instance to transform its inputs into its outputs. It is a separate object to reflect the fact that many Block instances may share a common algorithm.

Object: Algorithm

Key Attribute: Algorithm Name

Attribute: Algorithm content

5.3.2.1 *Algorithm Name*

This attribute is the name associated with the algorithm.

5.3.2.2 *Algorithm content*

This attribute represents the substance of the algorithm, the program or rules by which the output is determined based on the input and the internal state.

6 *Process control context mapping*

This clause prescribes how the facilities of MMS shall be used to satisfy the communication requirements of the Process Control Environment described in the previous clause. This clause also relates the models of process control systems to the abstract model of the Virtual Manufacturing Device (VMD) described in ISO/IEC 9506-1.

6.1 *Mapping the process control model to the VMD*

Each device of a process control system that offers Class A communication capabilities as a server shall be mapped onto at least one VMD.

EXAMPLE:

A programmable controller engaged in a process control application may consist of a single VMD. A large, distributed process control system, when viewed through a gateway from a production control computer, may also consist of a single VMD. A distributed process control system, when viewed from an internal element of the control system, such as an operator's console, may contain multiple VMDs with at least one VMD representing each station connected to the distributed process control system. A large distributed process control system with multiple connections to the supervisory network from the system gateway, the historian and operator's console may appear, from the supervisory network, as multiple VMDs.

6.2 *Definition of process control objects that map to Domains*

6.2.1 *The Block object*

The Block object shall map to a Domain and to a Program Invocation bound to this Domain. The Domain shall contain Domain specific Named Variable objects that shall

doivent représenter les entrées, les sorties, les paramètres d'état internes du Bloc. L'attribut Mode-Etat du Bloc doit également être représenté comme un objet Variable Nommée du Domaine. Si la syntaxe de l'Etiquette de Bloc est compatible avec celle d'un Identificateur (voir 7.6.2 de l'ISO/IEC 9506-2), le nom du Domaine doit être le même que celui de l'Etiquette de Bloc. Si la syntaxe de l'Etiquette de Bloc n'est pas compatible avec celle de l'Identificateur, aucune exigence ne s'applique au nom de Domaine. Cependant, l'annexe B.4 recommande une méthode permettant de déduire de l'Etiquette de Bloc un nom de Domaine.

6.2.2 L'objet Algorithme

L'objet Algorithme doit mapper un Domaine. Le nom de Domaine doit être le nom de l'Algorithme.

6.3 Définition des objets de Contrôle-Commande de procédé qui mappent des Invocations de Programme

6.3.1 L'objet Bloc

L'objet Bloc doit mapper un Domaine et une Invocation de Programme liés à ce domaine. Si la syntaxe de l'Etiquette de Bloc est compatible avec celle d'un Identificateur (voir 7.6.2 de l'ISO/IEC 9506-2), le nom de l'Invocation de Programme doit être le même que celui de l'Etiquette de Bloc. Si la syntaxe de l'Etiquette de Bloc n'est pas compatible avec celle de l'Identificateur, aucune exigence ne s'applique au nom d'Invocation de Programme. Cependant, l'annexe B.4 recommande une méthode permettant de déduire de l'Etiquette de Bloc un nom d'Invocation de Programme.

L'Invocation de Programme représentant le Bloc doit normalement être liée à deux Domaines, un dont le nom est le même que l'Invocation de Programme qui supporte les Variables Nommées qui représentent les entrées, les sorties, les variables d'état internes, et le Mode-Etat du Bloc. L'autre Domaine (qui peut être partageable) représente l'algorithme d'exécution du Bloc. Selon la nature du Bloc, il peut y avoir des Domaines supplémentaires liés à l'Invocation de Programme. Dans certains cas, le Bloc entier peut être représenté par un seul Domaine lié à une Invocation de Programme.

L'état de l'Invocation de Programme est utilisé dans ce modèle pour représenter l'attribut EnService du Bloc; lorsque l'Invocation de Programme est à l'état de REPOS ou ARRETE, l'attribut EnService est FAUX. Lorsque l'Invocation de Programme est à l'état EN EXECUTION, l'attribut EnService est VRAI.

6.4 Prescriptions de Contrôle-Commande de procédé affectant d'autres objets MMS

6.4.1 Extensions à l'objet Condition Événementielle

L'objet Condition Événementielle de l'article 15 de l'ISO/IEC 9506-1 est étendu par les attributs suivants:

Objet: Condition Événementielle

Tout attribut MMS défini

Attribut: Classe d'Enrichissement d'Affichage (TEXTE, NOMBRE, INDEFINI)

Contrainte: Classe d'Enrichissement d'Affichage = TEXTE

represent the inputs, outputs, internal state parameters of the Block. The Mode-State attribute of the Block shall also be represented as a Named Variable object of the Domain. If the syntax of the Block Tag is compatible with the syntax of an Identifier (see 7.6.2 of ISO/IEC 9506-2), the name of the Domain shall be the same as the Block Tag. If the syntax of the Block Tag is not compatible with Identifier syntax, there are no requirements on the Domain name. However, annex B.4 provides guidance on a recommended method for deriving a Domain name from the Block Tag.

6.2.2 *The Algorithm object*

The Algorithm object shall map to a Domain. The Domain name shall be the name of the Algorithm.

6.3 *Definition of process control objects that map to Program Invocations*

6.3.1 *The Block object*

The Block object shall map to a Domain and to a Program Invocation bound to this Domain. If the syntax of the Block Tag is compatible with the syntax of an Identifier (see 7.6.2 of ISO/IEC 9506-2), the name of the Program Invocation shall be the same as the Block Tag. If the syntax of the Block Tag is not compatible with Identifier syntax, there are no requirements on the Program Invocation name. However, annex B.4 provides guidance on a recommended method for deriving a Program Invocation name from the Block Tag.

The Program Invocation representing the Block will normally have two Domains bound to it, one whose name is the same as the Program Invocation supports the Named Variables that represent the inputs, outputs, internal state variables, and Mode-State of the Block. The other Domain (which may be shareable) represents the algorithm for the Block execution. Depending on the nature of the Block, there may be additional Domains bound to the Program Invocation. In some cases, the entire Block may be represented by a single Domain bound to a Program Invocation.

The state of the Program Invocation is used in this model to represent the InService attribute of the Block; when the Program Invocation is in the IDLE or STOPPED state, the InService attribute is FALSE. When the Program Invocation is in the RUNNING state, the InService attribute is TRUE.

6.4 *Process control requirements that affect other MMS objects*

6.4.1 *Extensions to the Event Condition object*

The Event Condition object in clause 15 of ISO/IEC 9506-1 is extended by the following attributes:

Object: Event Condition

All MMS defined attributes

Attribute: Display Enhancement Class (TEXT, NUMBER, UNDEFINED)

Constraint: Display Enhancement Class = TEXT

Attribut: Enrichissement d’Affichage (Chaîne de caractères)
 Contrainte: Classe d’Enrichissement d’Affichage = NOMBRE
 Attribut: Enrichissement d’Affichage (Entier)
 Attribut: Priorité de préemption d’un Groupe
 Attribut: Liste des Références de Liste de Conditions Événementielles de référence

6.4.1.1 *Classe d’Enrichissement d’Affichage*

Cet attribut identifie le type d’attribut Enrichissement d’Affichage de l’objet Condition Événementielle. Si la valeur de cet attribut est TEXTE, l’Enrichissement d’Affichage est une chaîne de caractères. Si la valeur de cet attribut est NOMBRE, l’Enrichissement d’Affichage est un entier. Si la valeur de cet attribut est INDEFINIE, l’attribut Enrichissement d’Affichage n’est pas présent.

6.4.1.2 *Enrichissement d’Affichage*

Cet attribut est utilisé pour des informations utiles à la préparation des affichages opérateurs au niveau de l’utilisateur MMS recevant la notification de l’événement. Ce type d’attribut est soit une chaîne de caractères, soit un entier, selon la valeur de l’attribut de Classe d’Enrichissement d’Affichage.

6.4.1.3 *Priorité de préemption d’un groupe*

Cet attribut doit contenir une valeur qui est soit INDEFINIE, soit une valeur entière entre zéro (0) et cent vingt-sept (127). Le zéro doit représenter le niveau de priorité le plus haut et cent vingt-sept doit représenter le niveau de priorité le plus bas. Si la valeur de cet attribut est définie, elle doit représenter une priorité de préemption qui doit être utilisée par le VMD en lieu et place de la valeur contenue dans l’attribut Priorité. Si la valeur de l’attribut Priorité de préemption d’un Groupe est INDEFINIE, le VMD doit utiliser la valeur de l’attribut Priorité en déterminant l’importance de l’objet Condition Événementielle. Lorsqu’un objet Condition Événementielle est créé au moyen du service DefineEventCondition (DéfinirConditionÉvénementielle), la valeur de cet attribut doit être initialisée à INDEFINIE.

EXEMPLE

Prenons le cas où une priorité d’objet Condition Événementielle est modifiée par un service agissant sur une Liste Condition Événementielle, puis ensuite individuellement modifiée par le service AlterEventCondition Monitoring (ModifierSupervisionConditionÉvénementielle), puis modifiée encore au niveau priorité de groupe par un service agissant sur une Liste de Conditions Événementielles. Après le premier changement, l’attribut Intervention sur Priorité de Groupe est changé et devient prioritaire sur l’attribut Priorité. L’attribut Priorité n’est pas changé mais n’est pas pris en compte étant donné qu’il y a intervention sur priorité de groupe en vigueur. L’attribut Priorité ne change pas par suite de la modification de Priorité, mais n’est pas pris en compte tant que l’intervention sur priorité de groupe n’a pas été retirée.

L’attribut Priorité de préemption d’un Groupe impose une priorité absolue, que la valeur soit supérieure ou inférieure à la valeur de l’attribut Priorité.

Il n’est pas nécessaire de mettre en oeuvre cet attribut si les objets Liste de Conditions Événementielles ne sont pas mis en oeuvre.

6.4.1.4 *Liste des références de Liste de Condition Événementielle de référence*

Cet attribut doit contenir une liste des références aux objets Liste de Conditions Événementielles qui font référence à l’objet de Condition Événementielle spécifique. L’objet

Attribute: Display Enhancement (String)

Constraint: Display Enhancement Class = NUMBER

Attribute: Display Enhancement (Integer)

Attribute: Group Priority Override

Attribute: List of referencing Event Condition List references

6.4.1.1 *Display Enhancement Class*

This attribute identifies the type of the Display Enhancement attribute of the Event Condition object. If the value of this attribute is TEXT, the Display Enhancement is of type character string. If the value of this attribute is NUMBER, the Display Enhancement is of type integer. If the value of this attribute is UNDEFINED, the Display Enhancement attribute is not present.

6.4.1.2 *Display Enhancement*

This attribute is used to contain information useful in preparing operator displays at the MMS user receiving notification of the event. The type of this attribute is either character string or integer, depending on the value of the Display Enhancement Class attribute.

6.4.1.3 *Group Priority Override*

This attribute shall contain a value that is either UNDEFINED or an integer value between zero (0) and one hundred twenty-seven (127). Zero shall represent the highest priority and one hundred twenty-seven shall represent the lowest priority. If the value of this attribute is defined, it shall represent an override priority value that shall be utilized by the VMD in place of the value contained in the Priority attribute. If the value of the Group Priority Override attribute is UNDEFINED, the VMD shall utilize the value of the Priority attribute in determining the importance of the Event Condition object. When an Event Condition object is created using the Define Event Condition service, the value of this attribute shall be initialized to UNDEFINED.

EXAMPLE:

Consider the case in which an Event Condition object's priority is modified using a service operating on an Event Condition List, and then individually altered using the AlterEventConditionMonitoring service, and then modified again at the group priority level using a service operating on an Event Condition List. Following the first change, the Group Priority Override attribute has changed and takes precedence over the Priority attribute. The Priority attribute has not changed, but is not considered since there is a group override in effect. The Priority attribute does change following the Priority change, but is still not considered until the group override is removed.

The Group Priority Override attribute imposes an absolute override, whether the value is higher or lower than the value of the Priority attribute.

This attribute need not be implemented if Event Condition List objects are not supported.

6.4.1.4 *List of referencing Event Condition List references*

This attribute shall contain a list of references to Event Condition List objects that reference the specific Event Condition object. The Event Condition List object is defined

Liste de Conditions Événementielles est défini dans le paragraphe 6.5.1 de la présente partie de l'ISO/IEC 9506. Lorsqu'un objet Condition Événementielle est créé par le service DefineEventCondition (DéfinirConditionÉvénementielle), la valeur de cet attribut doit avoir comme valeur initiale, liste vide.

Il n'est pas nécessaire de mettre en oeuvre cet attribut si les objets Liste de Conditions Événementielles ne sont pas mis en oeuvre.

6.4.2 Extensions aux objets Enveloppe Événementielle

L'objet Enveloppe Événementielle décrit dans l'article 15 de l'ISO/IEC 9506-1, est étendu comme suit:

Objet: Enveloppe Événementielle

Tout attribut MMS défini

Attribut: Classe d'Enrichissement d'Affichage (TEXTE, NOMBRE, INDEFINI)

Contrainte: Classe d'Enrichissement d'Affichage = TEXTE

Attribut: Enrichissement d'Affichage (Chaîne de caractères)

Contrainte: Classe d'Enrichissement d'Affichage = NOMBRE

Attribut: Enrichissement d'Affichage (Entier)

6.4.2.1 Classe d'Enrichissement d'Affichage

Cet attribut identifie le type d'attribut Enrichissement d'Affichage de l'objet. Si la valeur de cet attribut est TEXTE, l'Enrichissement d'Affichage est une chaîne de caractères. Si la valeur de cet attribut est NOMBRE, l'Enrichissement d'Affichage est un entier. Si la valeur de cet attribut est INDEFINIE, l'attribut Enrichissement d'Affichage n'est pas présent.

6.4.2.2 Enrichissement d'Affichage

Cet attribut est utilisé pour fournir des informations utiles à la préparation des affichages opérateur au niveau de l'utilisateur MMS recevant notification de l'événement. Ce type d'attribut est soit une chaîne de caractères, soit un entier, selon la valeur de l'attribut Classe d'Enrichissement d'Affichage.

6.5 Définitions des nouveaux objets abstraits MMS

6.5.1 Objet Liste de Condition Événementielle

L'objet Liste de Condition Événementielle doit être utilisé pour référencer des groupes d'objets Condition Événementielle qui doivent être manipulés comme des groupes. La prise en charge du CBB RECL (voir 6.6.3) indique la prise en charge des Listes de Condition Événementielle qui peuvent faire référence à d'autres Listes de Condition Événementielle. Si ce CBB n'est pas pris en charge, la liste peut ne contenir que des références à des Conditions Événementielles.

Objet: Liste de Conditions Événementielles

Attribut-Clé: Nom de Liste de Conditions Événementielles

Attribut: Destructible par MMS

in 6.5.1 of this part of ISO/IEC 9506. When an Event Condition object is created using the DefineEventCondition service, the value of this attribute shall be initialized to an empty list.

This attribute need not be implemented if Event Condition List objects are not supported.

6.4.2 Extensions to Event Enrollment objects

The Event Enrollment object in described in clause 15 of ISO/IEC 9506-1 is extended as follows:

Object: Event Enrollment

All MMS defined attributes

Attribute: Display Enhancement Class (TEXT, NUMBER, UNDEFINED)

Constraint: Display Enhancement Class = TEXT

Attribute: Display Enhancement (String)

Constraint: Display Enhancement Class = NUMBER

Attribute: Display Enhancement (Integer)

6.4.2.1 Display Enhancement Class

This attribute identifies the type of the Display Enhancement attribute of the Event Enrollment object. If the value of this attribute is TEXT, the Display Enhancement is of type character string. If the value of this attribute is NUMBER, the Display Enhancement is of type integer. If the value of this attribute is UNDEFINED, the Display Enhancement attribute is not present.

6.4.2.2 Display Enhancement

This attribute is used to contain information useful in preparing operator displays at the MMS user receiving notification of the event. The type of this attribute is either character string or integer, depending on the value of the Display Enhancement Class attribute.

6.5 Definitions of new MMS abstract objects

6.5.1 Event Condition List object

The Event Condition List object shall be used to reference groups of Event Condition objects that are required to be operated on as groups. Support of the RECL (see 6.6.3) CBB indicates support for Event Condition Lists that may refer to other Event Condition Lists. If this CBB is not supported, the list may contain only references to Event Conditions.

Object: Event Condition List

Key Attribute: Event Condition List name

Attribute: MMS Deletable

Attribut: Liste des références de Condition Événementielle

Attribut: Liste des références de Liste de Conditions Événementielles

Attribut: Liste des références de Liste de Conditions Événementielles de référence

6.5.1.1 *Nom de Liste de Conditions Événementielles*

Cet attribut, du type Nom d'Objet, doit être le nom d'identification de la Liste Conditions Événementielles. Un nom de Liste de Condition Événementielle peut avoir une portée VMD, Domaine ou AA.

6.5.1.2 *Destructible par MMS*

Cet attribut, du type booléen, doit indiquer si l'objet peut (VRAI) ou ne peut pas (FAUX) être détruit par le service défini au paragraphe 7.3.15 de la présente partie de l'ISO/IEC 9506.

6.5.1.3 *Liste des références de Conditions Événementielles*

Cet attribut doit contenir une liste des références aux objets Condition Événementielle. La portée des services agissant sur l'objet Liste de Conditions Événementielles doit comprendre des objets se trouvant à la fois dans l'attribut Liste des références de Conditions Événementielles et dans l'attribut Liste des références de Liste de Conditions Événementielles.

Du fait des contraintes de visibilité, si le nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, cet attribut doit contenir uniquement des références aux objets Condition Événementielle qui ont une portée VMD ou Domaine. Si l'attribut nom de Liste de Conditions Événementielles a une portée AA, cet attribut peut faire référence à des objets Condition Événementielle de n'importe quelle portée.

6.5.1.4 *Liste des références de Liste de Conditions Événementielles*

Cet attribut doit contenir une liste des références à des objets Liste de Conditions Événementielles qui sont les subordonnés hiérarchiques de l'objet Liste de Conditions Événementielles. Cet attribut, ainsi que celui se trouvant dans les objets Liste de Conditions Événementielles subordonnés ne doivent pas pouvoir contenir des références circulaires. La portée des services agissant sur l'objet Liste de Conditions Événementielles doit comprendre des objets inclus à la fois dans l'attribut Liste de Conditions Événementielles et dans l'attribut Liste des références de Liste de Conditions Événementielles. L'attribut Liste de références de Liste de Conditions Événementielles ne doit être présent que si le CBB RECL a été pris en compte.

Du fait des contraintes de visibilité, si le nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, cet attribut doit contenir uniquement des références aux objets Liste de Conditions Événementielles qui ont une portée VMD ou Domaine. Si l'attribut nom de Liste de Conditions Événementielles a une portée AA, cet attribut peut faire référence à des objets de Liste de Conditions Événementielles de n'importe quelle portée.

6.5.1.5 *Liste référant les référence de Liste de Conditions Événementielles*

Cet attribut doit contenir une liste des références aux autres objets de Liste de Conditions Événementielles qui font référence à cet objet de Liste de Conditions Événementielles

Attribute: List of Event Condition references

Attribute: List of Event Condition List references

Attribute: List of referencing Event Condition List references

6.5.1.1 *Event Condition List name*

This attribute, of type Object Name, shall be the name by which the Event Condition List is identified. An Event Condition List name may have VMD, Domain-specific or AA-specific scope.

6.5.1.2 *MMS deletable*

This attribute, of type Boolean, shall indicate whether (TRUE) or not (FALSE) this object may be deleted through the use of the service defined in 7.3.15 of this part of ISO/ IEC 9506.

6.5.1.3 *List of Event Condition references*

This attribute shall contain a list of references to Event Condition objects. The scope of services operating on the Event Condition List object shall include objects in both the List of Event Condition references attribute and the List of Event Condition List references attribute.

Because of visibility constraints, if the Event Condition List name has VMD-specific or Domain-specific scope, this attribute shall contain references only to Event Condition objects which have VMD-specific or Domain-specific scope. If the Event Condition List name attribute has AA-specific scope, this attribute may reference Event Condition objects of any scope.

6.5.1.4 *List of Event Condition List references*

This attribute shall contain a list of references to Event Condition List objects that are hierarchically subordinate to this Event Condition List object. It shall not be permissible for this attribute, nor for this attribute in subordinate Event Condition List objects, to contain circular references. The scope of services operating on the Event Condition List object shall include objects included in both the List of Event Condition references attribute and the List of Event Condition List references attribute. The List of Event Condition List references attribute shall be present only if the RECL CBB has been negotiated.

Because of visibility constraints, if the Event Condition List name has VMD-specific or Domain-specific scope, this attribute shall contain references only to Event Condition List objects which have VMD-specific or Domain-specific scope. If the Event Condition List name attribute has AA-specific scope, this attribute may reference Event Condition List objects of any scope.

6.5.1.5 *List of referencing Event Condition List references*

This attribute shall contain a list of references to other Event Condition List objects that reference this specific Event Condition List object. If there are no references to this Event

spécifique. S'il n'y a pas de référence à cet objet de Liste de Conditions Événementielles, la valeur de cet attribut doit être une liste vide. Cet attribut ne doit être pris en compte que si le CBB RECL a été pris en compte.

NOTE – Cet attribut est nécessaire pour décrire pleinement les procédures de service agissant sur l'objet Liste de Conditions Événementielles. Cet attribut n'est ni visible ni modifiable par les services traités dans cette partie de l'ISO/IEC 9506.

6.5.2 *Objet Contrôle-Commande d'Unité*

L'objet Contrôle-Commande d'Unité doit représenter un ensemble d'objets MMS représentant des groupes de Domaines et/ou d'Invocations de Programme qui peut être chargé et géré comme une unité collective. Les téléchargements et télésauvegardes peuvent être effectués sur un objet Contrôle-Commande d'Unité pour transmettre de manière efficace les informations représentant des Domaines multiples vers et à partir d'un système de Contrôle-Commande de procédé en une ou plusieurs séquences de téléchargement ou de télésauvegarde. Bien qu'il soit destiné à satisfaire aux exigences de manipulation de groupes de Blocs, l'objet Contrôle-Commande d'Unité peut être utilisé pour manipuler des groupes de Domaines et/ou d'Invocations de Programme. Les services qui agissent sur l'objet Contrôle-Commande d'Unité sont décrits dans 7.3.1 à 7.3.13.

NOTE – L'objet Contrôle-Commande d'Unité peut être utilisé pour réduire le nombre de PDU nécessaire au téléchargement d'un grand nombre de Domaines ou pour créer un grand nombre d'Invocations de Programme. Les règles concernant l'applicabilité du regroupement de ces objets ne s'inscrivent pas dans le domaine d'application de la présente partie de l'ISO/IEC 9506.

Objet: Contrôle-Commande d'Unité

Attribut-Clé: Nom de Contrôle-Commande d'Unité

Attribut: Destructible par MMS (VRAI, FAUX)

Attribut: Liste de références de Domaine

Attribut: Liste de références d'Invocation de Programme

6.5.2.1 *Nom de Contrôle-Commande d'Unité*

Cet attribut doit uniquement identifier l'objet Contrôle-Commande d'Unité au niveau du VMD. Le nom du Contrôle-Commande d'Unité a une portée VMD.

6.5.2.2 *Destructible par MMS*

Cet attribut doit indiquer si l'objet Contrôle-Commande d'Unité peut (VRAI) ou ne peut pas (FAUX) être détruite par le service DeleteUnitControl (DétruireContrôleCommande d'Unité).

6.5.2.3 *Liste des références de Domaine*

Cet attribut doit identifier les objets Domaine qui constituent l'objet Contrôle-Commande d'Unité, et qui peuvent être affectés par des opérations sur l'objet Contrôle-Commande d'Unité.

6.5.2.4 *Liste des références d'Invocation de Programme*

Cet attribut doit identifier les objets Invocation de Programme qui constituent l'objet Contrôle-Commande d'Unité, et qui peuvent être affectés par des opérations sur l'objet Contrôle-Commande d'Unité.

Condition List object, the value of this attribute shall be an empty list. This attribute shall be present only if the RECL CBB has been negotiated.

NOTE – This attribute is necessary to fully describe the service procedures that operate on the Event Condition List object. This attribute is not visible nor modifiable via the services provided in this part of ISO/IEC 9506.

6.5.2 Unit Control object

The Unit Control object shall represent a collection of MMS objects representing groups of Domains and/or Program Invocations that may be loaded and managed as a collective unit. Downloads and uploads may be performed on a Unit Control object to efficiently transmit the information representing multiple Domains to and from a process control system in one or more download or upload sequences. While introduced to satisfy the requirement of manipulating groups of Blocks, the Unit Control object may be used to manipulate groups of Domains and/or Program Invocations. Services that operate on the Unit Control object are described in 7.3.1 through 7.3.13.

NOTE – The Unit Control object may be utilized to minimize the number of PDU's necessary to download large numbers of Domains or to create large numbers of Program Invocations. Rules concerning the applicability of grouping of these objects is outside the scope of this part of ISO/IEC 9506.

Object: Unit Control

Key Attribute: Unit Control Name

Attribute: MMS Deletable (TRUE, FALSE)

Attribute: List of Domain references

Attribute: List of Program Invocation references

6.5.2.1 Unit Control Name

This attribute shall uniquely identify the Unit Control object at the VMD. The name scope of the Unit Control object shall be VMD-specific.

6.5.2.2 MMS Deletable

This attribute shall indicate whether (TRUE) or not (FALSE) the Unit Control object may be deleted using the DeleteUnitControl service.

6.5.2.3 List of Domain references

This attribute shall identify the Domain objects that are constituents of the Unit Control object, and that may be affected by operations on the Unit Control object.

6.5.2.4 List of Program Invocation references

This attribute shall identify the Program Invocation objects that are constituents of the Unit Control object, and may be affected by operations on the Unit Control object.

6.6 Conformité des paramètres

La présente partie de l'ISO/IEC 9506 introduit trois nouveaux blocs de conformité (CBB) de paramètres dans la spécification du système.

6.6.1 DES

Le bloc de conformité de paramètre DES doit établir la validité de la forme textuelle du paramètre enrichissement d'affichage, lorsqu'il apparaît dans une demande ou lorsqu'il est indiqué dans une table de service.

Si DES est pris en charge, ce paramètre est valide. Autrement, ce paramètre n'est pas valide. Si DES n'est pas pris en charge, une demande spécifiant ce paramètre doit entraîner une erreur de protocole.

Si DES est pris en charge, DEI ne doit pas être pris en charge.

6.6.2 DEI

Le bloc de conformité de paramètre DEI doit établir la validité de la forme numérique du paramètre d'enrichissement d'affichage, lorsqu'il apparaît dans une demande ou lorsqu'il est indiqué dans une table de service.

Si DEI est pris en charge, le paramètre est valide. Autrement, le paramètre n'est pas valide. Si DEI n'est pas pris en charge, une demande spécifiant ce paramètre doit entraîner une erreur de protocole.

Si DEI est pris en charge, DES ne doit pas être pris en charge.

6.6.3 RECL

Le bloc de conformité de paramètre RECL doit établir la validité du paramètre Liste des Listes de Conditions Événementielles, lorsqu'il apparaît dans une table de service.

Si RECL est pris en charge, ce paramètre est valide. Autrement, ce paramètre n'est pas valide. Une demande ou une réponse spécifiant ce paramètre doit entraîner une erreur de protocole. Si RECL n'est pas pris en charge, une demande qui, si elle est honorée, nécessiterait une réponse spécifiant ce paramètre, doit entraîner une erreur de service précisant une classe d'erreur égale à ACCES et un code d'erreur égal à OBJECT-ACCESS-UNSUPPORTED (ACCES-OBJET-NON-SUPPORTE).

7 Services

7.1 Utilisation des services ACSE

Pour prendre en charge certaines opérations MMS qui affectent l'état du système de Contrôle-Commande de procédé, le serveur MMS doit exiger que le client MMS ait fourni le titre AP dans le service A-ASSOCIATE et en outre que le client MMS ait spécifié l'unité fonctionnelle d'Authentification de ACSE et fourni des valeurs appropriées de Classification de Privilège et d'Identificateur de Privilèges qui font partie de la Valeur d'Authentification du service A-ASSOCIATE. Ces valeurs doivent devenir des valeurs des attributs correspondants de l'objet Association d'Application (voir annexe A). Si le client MMS n'a pas fourni ces valeurs, les attributs correspondants doivent être mis sur la valeur INDEFINIE, ou le serveur MMS peut, s'il le désire, refuser d'établir l'association.

6.6 *Parameter conformance*

This part of ISO/IEC 9506 introduces three new parameter conformance building blocks (CBBs) in the system specification.

6.6.1 *DES*

The DES parameter conformance building block shall establish the validity of the text form of the display enhancement parameter, whenever it occurs in a request or indication in a service table.

If DES is supported, the parameter is valid. Otherwise, the parameter is invalid. If DES is not supported, a request specifying this parameter shall constitute a protocol error.

If DES is supported, DEI shall not be supported.

6.6.2 *DEI*

The DEI parameter conformance building block shall establish the validity of the numeric form of the display enhancement parameter, whenever it occurs in a request or indication in a service table.

If DEI is supported, the parameter is valid. Otherwise, the parameter is invalid. If DEI is not supported, a request specifying this parameter shall constitute a protocol error.

If DEI is supported, DES shall not be supported.

6.6.3 *RECL*

The RECL parameter conformance building block shall establish the validity of the List of Event Condition List parameter, whenever it occurs in a service table.

If RECL is supported, this parameter is valid. Otherwise, the parameter is invalid. A request or response specifying this parameter shall constitute a protocol error. If RECL is not supported, a request which, if honoured, would require a response specifying this parameter shall result in a service error specifying error class equal to ACCESS and error code equal to OBJECT-ACCESS-UNSUPPORTED.

7 *Services*

7.1 *Use of ACSE services*

In order to support certain MMS operations that affect the state of the process control system, the MMS server shall require that the MMS client has provided the AP-title in the A-ASSOCIATE service and further that the MMS client has specified the Authentication functional unit of ACSE and provided proper values of Privilege Classification and Privilege Identifier as part of the Authentication Value of the A-ASSOCIATE service. These values shall become values of the corresponding attributes of the Application Association object (see annex A). If the MMS-client has not supplied such values, the corresponding attributes shall be set to a value of UNDEFINED, or the MMS server may, at its option, refuse establishment of the association.

Pour pouvoir définir la syntaxe du mécanisme d'Authentification, la présente partie de l'ISO/IEC 9506 attribue la valeur ASN.1 de l'objet identificateur et le module ASN.1 associé à la sélection externe de valeur d'Authentification.

```
ISO/IEC-9506-Authentication { iso/iec standard 9506 part (6)
authentication (3) }
```

```
DEFINITIONS::= BEGIN
```

```
Authentication-syntax::= SEQUENCE {
    user                      VisibleString,
    password                  OCTET STRING OPTIONAL,
    privilege classification  INTEGER OPTIONAL
}
```

```
END
```

7.2 Utilisation des services MMS

Le présent paragraphe prescrit l'utilisation des services et du protocole définis dans l'ISO/IEC 9506-1 et dans l'ISO/IEC 9506-2, des extensions à certains de ces services, ainsi que le protocole permettant de prendre en charge les extensions.

7.2.1 Nom de contexte d'application de procédé

Pour pouvoir utiliser une application qui ne contient que l'ACSE et MMS en tant que ASE, la présente section de l'ISO/IEC 9506 utilise la valeur de l'identificateur d'objet et la valeur du descripteur d'objet définies dans 17.12 de l'ISO/IEC 9506-2.

7.2.2 Définition de la syntaxe abstraite de Contrôle-Commande de procédé

La présente partie de l'ISO/IEC 9506 attribue la valeur ASN.1 d'identificateur d'objet

```
{iso/iec standard 9506 part (6) mms-process-syntax-version (1) }
```

à la syntaxe abstraite définie dans le présent article.

7.2.3 Définition du module ASN.1 spécifique au Contrôle-Commande de procédé

Les services et le protocole MMS ont été développés pour être utilisés par une large gamme d'équipements de fabrication. Cette section définit les services de Contrôle-Commande de procédé et le protocole pour des éléments identifiés comme nécessitant une définition de norme d'accompagnement dans la MMS. Il convient d'utiliser ces définitions lorsque la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 est prise en compte.

Toutes les définitions ASN.1 fournies dans la présente partie de l'ISO/IEC 9506 sont incluses dans le Module ASN.1 «ISO-9506-MMS-PROCESS-1». Les instructions de début et de fin indiquant que chaque définition ASN.1 fournie fait partie de ce module sont omises afin de faciliter la lecture du document. Chaque définition ASN.1 fournie contient implicitement l'instruction:

```
ISO/IEC-9506-MMS-PROCESS-1 { iso/iec standard 9506 part (6)
mms-process-module-version1 (2) }
```

```
DEFINITIONS::= BEGIN
```

au début de la définition et le mot-clé «END» à la fin de la définition.

For the purpose of providing a syntax definition for the Authentication mechanism, this part of ISO/IEC 9506 assigns the ASN.1 object identifier value and the associated ASN.1 module for the external choice of Authentication-value.

```
ISO/IEC-9506-Authentication {iso/iec standard 9506 part (6)
authentication(3) }
```

```
DEFINITIONS ::= BEGIN
```

```
Authentication-syntax ::= SEQUENCE {
    user                      VisibleString,
    password                  OCTET STRING OPTIONAL,
    privilege-classification  INTEGER OPTIONAL
}
```

```
END
```

7.2 Use of MMS services

This subclause specifies the use of services and protocol defined in ISO/IEC 9506-1 and ISO/IEC 9506-2, extensions to some of those services, and protocol to support the extensions.

7.2.1 Process application context name

For the purpose of being able to use an application that only contains the ACSE and MMS as ASEs, this part of ISO/IEC 9506 uses the object identifier value and the object descriptor value defined in 17.12 of ISO/IEC 9506-2.

7.2.2 Process control abstract syntax definition

This part of ISO/IEC 9506 assigns the ASN.1 object identifier value

```
{iso/iec standard 9506 part (6) mms-process-syntax-version(1) }
```

to the abstract syntax defined in this clause.

7.2.3 Process control specific ASN.1 module definition

The MMS services and protocol were developed to be used by a wide range of manufacturing devices. This section defines the process control services and protocol for those elements identified as requiring companion standard definition in MMS. These definitions should be used when the abstract syntax defined in this part of ISO/IEC 9506 is negotiated.

All ASN.1 definitions provided in this part of ISO/IEC 9506 are part of the ASN.1 Module "ISO-9506-MMS-PROCESS-1." The beginning and closing statements indicating that each ASN.1 definition provided is a part of this module is omitted in order to make reading of the document easier. Each ASN.1 definition provided implicitly contains the statement:

```
ISO/IEC-9506-MMS-PROCESS-1 {iso/iec standard 9506 part (6) mms-process-
module-version1(2) }
```

```
DEFINITIONS ::= BEGIN
```

at the beginning of the definition and contains the keyword "END" at the end of the definition.

NOTE - ISO-9506-MMS-PROCESS-1 représente le numéro de la version 1 de la Norme d'Accompagnement MMS (ISO/IEC 9506-6) pour la conduite de Procédé.

De nombreux termes et abréviations utilisés dans le présent article utilisent la terminologie des descriptions de services et de protocole MMS (voir article 5 de l'ISO/IEC 9506-1 et l'article 5 de l'ISO/IEC 9506-2). Il convient en particulier de se reporter à l'article 5 de l'ISO/IEC 9506-1 pour les règles générales d'interprétation des tables de service utilisés dans la présente partie de l'ISO/IEC 9506.

IMPORTS MMSpdu,

ParameterSupportOptions,
ServiceSupportOptions,
Integer16,
StatusResponse,
Identifier,
ObjectName,
ProgramInvocationState
FileName,
ApplicationReference,
Priority,
EventTime,
EC-State

FROM MMS-General-Module-1

{iso/iec standard 9506 part (2) mms-general-module-version1 (2)};

7.2.4 Services relatifs au VMD

7.2.4.1 Extensions de service GetNameList

Les services GetNameList doivent être étendus pour inclure l'objet Liste de Conditions Événementielles et l'objet Contrôle-Commande d'Unité.

7.2.4.1.1 Paramètre de service Classe d'Objet NA

Ce paramètre, du type entier, doit identifier l'objet Liste de Conditions Événementielles et l'objet Contrôle-Commande d'Unité. La valeur zéro (0) doit représenter l'objet Liste de Conditions Événementielles et la valeur un (1) doit représenter l'objet Contrôle-Commande d'Unité.

7.2.4.1.2 Extensions du protocole

La syntaxe abstraite du paramètre CsAdditionalObjectClasses (Classes d'Objet Supplémentaire NA) doit être précisée comme suit:

```
CsAdditionalObjectClasses ::= IMPLICIT INTEGER {
    eventConditionList          (0),
    unitControl                  (1)
}
```

NOTE – ISO-9506-MMS-PROCESS-1 represents version number 1 of the MMS (ISO/IEC 9506-6) Companion Standard for Process Control.

Many of the terms and abbreviations used in this clause use the terminology of MMS service and protocol descriptions (see clause 5 of ISO/IEC 9506-1 and clause 5 of ISO/IEC 9506-2). In particular, refer to clause 5 of ISO/IEC 9506-1 for general rules on how to interpret the service tables in this part of ISO/IEC 9506.

```

IMPORTS MMSpdu,
    ParameterSupportOptions,
    ServiceSupportOptions,
    Integer16,
    StatusResponse,
    Identifier,
    ObjectName,
    ProgramInvocationState,
    FileName,
    ApplicationReference,
    Priority,
    EventTime,
    EC-State
FROM MMS-General-Module-1
    {iso/iec standard 9506 part (2) mms-general-module-version1 (2) };

```

7.2.4 VMD support services

7.2.4.1 GetNameList service extensions

The GetNameList service shall be extended to include the Event Condition List object and the Unit Control object.

7.2.4.1.1 CS Object Class service parameter

This parameter, of type integer, shall identify the Event Condition List object and the Unit Control object. The value zero (0) shall represent the Event Condition List object, and the value one (1) shall represent the Unit Control object.

7.2.4.1.2 Protocol extensions

The abstract syntax of the CsAdditionalObjectClasses parameter shall be specified as follows:

```

CsAdditionalObjectClasses ::= IMPLICIT INTEGER {
    eventConditionList      (0),
    unitControl              (1)
}

```

7.2.5 Services de gestion d'événements

Le présent paragraphe définit l'extension des services de Gestion d'Evénements.

7.2.5.1 Service DefineEventCondition (DéfinirConditionEvénementielle)

Le présent paragraphe définit l'extension du service DefineEventCondition.

7.2.5.1.1 Extensions du paramètre

Le service DefineEventCondition doit être étendu pour inclure la spécification de la valeur de l'attribut Enrichissement de l'Affichage de l'objet Condition Événementielle. La structure des extensions du paramètre DefineEventCondition est définie dans le tableau 1.

Tableau 1 – Extensions de paramètre DefineEventCondition

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Enrichissement d'Affichage	M	M(=)			
Chaîne d'Enrichissement d'Affichage	S	S(=)			DES
Indice d'Enrichissement d'Affichage	S	S(=)			DEI
Pas d'Enrichissement	S	S(=)			

7.2.5.1.1.1 Enrichissement d'affichage

La sélection de ce paramètre indique que l'attribut Enrichissement d'Affichage de la Condition Événementielle doit être modifié par ce service. Si ce paramètre est sélectionné, l'un des paramètres suivants doit apparaître.

7.2.5.1.1.1.1 Chaîne d'Enrichissement d'Affichage

Ce paramètre, du type chaîne de caractères, est la forme textuelle du paramètre Enregistrement d'Affichage. Cette sélection peut uniquement être effectuée si le CBB DES a été pris en compte.

7.2.5.1.1.1.2 Indice d'Enrichissement d'Affichage

Ce paramètre, du type entier, est la forme numérique du paramètre d'Enrichissement d'Affichage. Cette sélection peut uniquement être effectuée si le CBB DEI a été pris en compte.

7.2.5.1.1.1.3 Pas d'Enrichissement

Ce paramètre, du type à valeur NULLE, indique qu'aucun Enrichissement d'Affichage n'est présent. Ce paramètre doit être sélectionné si ni DES ni DEI n'ont été pris en compte.

7.2.5 Event management services

This subclause specifies the extension of the Event Management services.

7.2.5.1 DefineEventCondition service

This subclause specifies the extension of the DefineEventCondition service.

7.2.5.1.1 Parameter extensions

The DefineEventCondition service shall be extended to include specification of the value of the Display Enhancement attribute of the Event Condition object. The structure of the DefineEventCondition parameter extensions is specified in table 1.

Table 1 – DefineEventCondition parameter extensions

Parameter name	Req	Ind	Rep	Cnf	CBB
Display Enhancement	M	M(=)			
Display Enhancement string	S	S(=)			DES
Display Enhancement index	S	S(=)			DEI
No Enhancement	S	S(=)			

7.2.5.1.1.1 Display Enhancement

Selection of this parameter indicates that the Display Enhancement attribute of the Event Condition shall be altered by this service. If this parameter is selected, one of the following parameters shall appear.

7.2.5.1.1.1.1 Display Enhancement string

This parameter, of type character string, is the string form of the Display Enhancement parameter. This selection may be made only if the DES CBB has been negotiated.

7.2.5.1.1.1.2 Display Enhancement index

This parameter, of type integer, is the numeric form of the Display Enhancement parameter. This selection may be made only if the DEI CBB has been negotiated.

7.2.5.1.1.1.3 No Enhancement

This parameter, of type NULL, specifies that no Display Enhancement is present. This parameter shall be selected if neither DES nor DEI has been negotiated.

7.2.5.1.2 Extensions de procédure de service

La procédure de service du 15.2.2 de l'ISO/IEC 9506-1 doit être exécutée. Si la chaîne d'Enrichissement d'Affichage a été sélectionnée, la valeur de l'attribut Classe d'Enrichissement d'Affichage de la Condition Événementielle doit être TEXTE et l'attribut Enrichissement d'Affichage doit être mis sur la valeur du paramètre chaîne d'Enrichissement d'Affichage. Si l'indice Enrichissement d'Affichage est sélectionné, l'attribut Classe Enrichissement d'Affichage de la Condition Événementielle doit être NOMBRE et l'attribut d'Enrichissement d'Affichage doit être mis sur la valeur du paramètre indice d'Enrichissement d'Affichage. Autrement, l'attribut de Classe d'Enrichissement d'Affichage doit être mis sur INDEFINI.

7.2.5.1.3 Extensions de protocole

Les extensions de l'argument de service DefineEventCondition doivent être CS-DefineEventCondition-Request (Demande- Définition de Condition Événementielle-NA).

```
CS-DefineEventCondition-Request ::= [0] Choice {
    enhancementString [0] IMPLICIT VisibleString,
    enhancementIndex [1] IMPLICIT INTEGER,
    noEnhancement      NULL
}
```

NOTE – Etant donné la manière dont Confirmed-RequestPDU (PDU Demande-Confirmée) est prescrit dans l'ISO/IEC 9506-2, une valeur NULLE spécifiée comme extension de la Norme d'Accompagnement n'est pas transmise. De ce fait, le paramètre est considéré non inclus dans le protocole. Il est cependant prescrit de préciser la valeur NULLE au niveau de l'interface de service.

7.2.5.2 Extensions de service DeleteEventCondition (Détruire Condition Événementielle)

La procédure de service du service DeleteEventCondition doit être étendue de manière à inclure une vérification permettant de s'assurer que la valeur de l'attribut Liste de références de Liste de Conditions Événementielles de référence de l'objet Condition Événementielle spécifié équivaut à une liste vide. La vérification doit être effectuée avant d'exécuter la procédure spécifiée dans l'ISO/IEC 9506-1. Si la valeur de cet attribut n'équivaut pas à une liste vide, l'objet Condition Événementielle spécifié doit être considéré comme Candidat Non Détruit, la Condition Événementielle ne doit pas être détruite et une Réponse(-) doit être renvoyée.

7.2.5.3 Service GetEventConditionAttributes (Obtenir Attribut de Condition Événementielle)

Ce paragraphe prescrit l'extension du service GetEventConditionAttributes.

7.2.5.3.1 Extensions du paramètre

La réponse du service GetEventConditionAttributes doit être étendue pour inclure le paramètre d'Intervention sur Priorité de Groupe, le paramètre Liste des Listes de Conditions Événementielles de référence et le paramètre Enrichissement d'Affichage. La structure des extensions du paramètre GetEventConditionAttributes est définie dans le tableau 2.

7.2.5.1.2 *Service procedure extensions*

The service procedure of 15.2.2 of ISO/IEC 9506-1 shall be performed. If the Display Enhancement string has been selected, the value of the Display Enhancement Class attribute of the Event Condition shall be TEXT and the Display Enhancement attribute shall be set to the value of the Display Enhancement string parameter. If the Display Enhancement index is selected, the Display Enhancement Class attribute of the Event Condition shall be NUMBER and the Display Enhancement attribute shall be set to the value of the Display Enhancement index parameter. Otherwise the Display Enhancement Class attribute shall be set to UNDEFINED.

7.2.5.1.3 *Protocol extensions*

The extensions to the DefineEventCondition service argument shall be the CS-DefineEventCondition-Request.

```
CS-DefineEventCondition-Request ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

NOTE – As a result of the way in which Confirmed-RequestPDU is specified in ISO/IEC 9506-2, a NULL specified as a Companion Standard extension is not transmitted. The effect is as though the parameter were not included in the protocol. Specification of the NULL is required at the service interface, however.

7.2.5.2 *DeleteEventCondition service extensions*

The service procedure of the DeleteEventCondition service shall be extended to include a verification that the value of the List of referencing Event Condition List references attribute of the specified Event Condition object is equal to an empty list. The verification shall be performed prior to performing the procedure specified in ISO/IEC 9506-1. If the value of this attribute is not equal to an empty list, the specified Event Condition object shall be counted as a Candidate Not Deleted, the Event Condition shall not be deleted, and a Response(-) shall be returned.

7.2.5.3 *GetEventConditionAttributes service*

This subclause specifies the extension of the GetEventConditionAttributes service.

7.2.5.3.1 *Parameter extensions*

The GetEventConditionAttributes service response shall be extended to include the Group Priority Override parameter, the List of referencing Event Condition Lists parameter and the Display Enhancement parameter. The structure of the GetEventConditionAttributes parameter extensions is specified in table 2.

Tableau 2 – Extensions du paramètre GetEventConditionAttributes

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Intervention sur Priorité de Groupe			C	C (=)	
Liste des Listes de Conditions Événementielles de référence			C	C (=)	
Enrichissement d’Affichage			M	M (=)	
Chaîne d’Enrichissement d’Affichage			S	S (=)	
Indice d’Enrichissement d’Affichage			S	S (=)	
Pas d’Enrichissement			S	S (=)	

7.2.5.3.2 Intervention sur Priorité de Groupe

Ce paramètre, du type entier, doit contenir la valeur de l'attribut Intervention sur Priorité de Groupe de l'objet Condition Événementielle.

7.2.5.3.3 Liste des Listes de Conditions Événementielles de référence

Ce paramètre doit contenir une liste de noms, extraits du contenu de l'attribut Liste des références de Liste de Conditions Événementielles de références de l'objet Condition Événementielle spécifié. Chaque nom doit être égal à la valeur de l'attribut nom de l'objet Liste de Conditions Événementielles de référence.

7.2.5.3.4 Enrichissement de l’Affichage

Ce paramètre indique la valeur de l'attribut Classe d'Enrichissement d’Affichage de l'objet Condition Événementielle. Selon sa valeur, l'un des paramètres suivants doit être sélectionné.

7.2.5.3.4.1 Chaîne d’Enrichissement d’Affichage

Ce paramètre, du type chaîne de caractères, est la forme textuelle du paramètre Enrichissement d’Affichage.

7.2.5.3.4.2 Indice d’Enrichissement d’Affichage

Ce paramètre, du type entier, est la forme numérique du paramètre Enrichissement d’Affichage.

7.2.5.3.4.3 Pas d’Enrichissement

Ce paramètre, du type à valeur NULLE, indique qu'aucun Enrichissement d’Affichage n'est présent.

7.2.5.3.5 Extensions de procédure de service

La procédure de service du 15.4.2 de l'ISO/IEC 9506-1 doit être exécutée. Si l'attribut Classe d'Enrichissement d’Affichage de la Condition Événementielle à la valeur TEXTE, le

Table 2 – GetEventConditionAttributes parameter extensions

Parameter name	Req	Ind	Rep	Cnf	CBB
Group Priority Override			C	C(=)	
List of referencing Event Condition Lists					
Condition Lists			C	C(=)	
Display Enhancement			M	M(=)	
Display Enhancement string			S	S(=)	
Display Enhancement index			S	S(=)	
No Enhancement			S	S(=)	

7.2.5.3.2 Group Priority Override

This parameter, of type integer, shall contain the value of the Group Priority Override attribute of the Event Condition object.

7.2.5.3.3 List of referencing Event Condition Lists

This parameter shall contain a list of names, derived from the contents of the List of referencing Event Condition List references attribute of the specified Event Condition object. Each name shall be equal to the value of the name attribute of a referencing Event Condition List object.

7.2.5.3.4 Display Enhancement

This parameter indicates the value of the Display Enhancement Class attribute of the Event Condition object. Depending on its value, one of the following parameters shall be selected.

7.2.5.3.4.1 Display Enhancement string

This parameter, of type character string, is the string form of the Display Enhancement parameter.

7.2.5.3.4.2 Display Enhancement index

This parameter, of type integer, is the numeric form of the Display Enhancement parameter.

7.2.5.3.4.3 No Enhancement

This parameter, of type NULL, specifies that no Display Enhancement is present.

7.2.5.3.5 Service procedure extensions

The service procedure of 15.4.2 of ISO/IEC 9506-1 shall be performed. If the Display Enhancement Class attribute of the Event Condition has the value TEXT, the Display

paramètre chaîne de caractères d'Enrichissement d'Affichage doit être sélectionné. Si l'attribut Classe d'Enrichissement d'Affichage a la valeur NOMBRE, l'indice d'Enrichissement d'Affichage doit être sélectionné. Si l'attribut Classe d'Enrichissement d'Affichage a la valeur INDEFINIE, le paramètre Pas d'Enrichissement doit être sélectionné.

La procédure de service du service GetEventConditionAttributes doit être étendue pour inclure la valeur de l'attribut d'Intervention sur Priorité de Groupe de l'objet Condition Événementielle en tant que paramètre d'Intervention sur Priorité de Groupe et la valeur de l'attribut Liste des références de Liste de Conditions Événementielles de référence en tant que valeur du paramètre Liste des Listes de Conditions Événementielles de référence. Ces paramètres doivent apparaître si, et seulement si, un ou plusieurs des services pour des objets Liste de Conditions Événementielles sont supportés. La valeur de l'attribut Enrichissement d'Affichage de l'objet Condition Événementielle doit être incluse comme paramètre d'Enrichissement d'Affichage.

7.2.5.3.6 Extensions du protocole

Les extensions du protocole de réponse au service GetEventConditionAttributes doivent être CS-GetEvent Condition Attributes-Response (Obtenir Réponse Attributs de Condition Événementielle-NA).

```
CS-GetEventConditionAttributes-Response ::= CHOICE {
  supplied          SEQUENCE {
    GroupPriorityOverride [0] CHOICE {
      priority          [0] IMPLICIT Priority,
      undefined         [1] IMPLICIT NULL
    } OPTIONAL
    listOfEventConditionList [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
    displayEnhancement [2] CHOICE {
      enhancementString [0] IMPLICIT VisibleString,
      enhancementIndex  [1] IMPLICIT INTEGER,
      noEnhancement     [2] IMPLICIT NULL
    },
    default           NULL
  }
}
```

Le choix par défaut doit être sélectionné si, et seulement si, groupPriorityOverride n'est pas émis, le champ listOfEventConditionList n'est pas transmis et la valeur noEnhancement est sélectionnée dans le champ Enrichissement d'Affichage.

7.2.5.3.7 GroupPriorityOverride

La priorité doit être sélectionnée lorsque la valeur du paramètre Intervention sur Priorité de Groupe de la réponse de service GetEventConditionAttributes n'est pas égale à INDEFINIE; autrement, INDEFINIE doit être sélectionnée.

Enhancement string parameter shall be selected. If the Display Enhancement Class attribute has the value NUMBER, the Display Enhancement index shall be selected. If the Display Enhancement Class attribute has the value UNDEFINED, the No Enhancement parameter shall be selected.

The service procedure of the GetEventConditionAttributes service shall be extended to include the value of the Group Priority Override attribute of the Event Condition object as the Group Priority Override parameter, and the value of the List of referencing Event Condition List references attribute as the value of the List of referencing Event Condition Lists parameter. These parameters shall only appear if, and only if, one or more of the services for Event Condition List objects are supported. The value of the Display Enhancement attribute of the Event Condition object shall be included as the Display Enhancement parameter.

7.2.5.3.6 Protocol extensions

The extensions to the GetEventConditionAttributes service response protocol shall be the CS-GetEventConditionAttributes-Response.

```
CS-GetEventConditionAttributes-Response ::= CHOICE {
  supplied          SEQUENCE {
    GroupPriorityOverride [0] CHOICE {
      priority          [0] IMPLICIT Priority,
      undefined         [1] IMPLICIT NULL
    } OPTIONAL,
    listOfEventConditionList [1] IMPLICIT SEQUENCE OF ObjectName OPTIONAL,
    displayEnhancement [2] CHOICE {
      enhancementString [0] IMPLICIT VisibleString,
      enhancementIndex  [1] IMPLICIT INTEGER,
      noEnhancement     [2] IMPLICIT NULL
    } },
  default          NULL
}
```

The default choice shall be chosen if, and only if, the groupPriorityOverride is not transmitted, the listOfEventConditionList field is not transmitted, and the noEnhancement choice is selected for the display Enhancement field.

7.2.5.3.7 GroupPriorityOverride

The priority choice shall be selected when the value of the Group Priority Override parameter of the GetEventConditionAttributes service response is not equal to UNDEFINED; otherwise, the UNDEFINED choice shall be selected.

7.2.5.3.8 Enrichissement d’Affichage

Si la valeur de l’attribut Classe d’Enrichissement d’Affichage de l’objet Condition Événementielle est égale à TEXTE, la Chaîne d’enrichissement doit être sélectionnée. Si la valeur de l’attribut Classe d’Enrichissement d’Affichage de l’objet Condition Événementielle est égale à NOMBRE, l’Indice d’Enrichissement doit être sélectionné. Autrement, pas d’Enrichissement doit être sélectionné.

7.2.5.4 Service AlterEventConditionMonitoring (Modifier Surveillance de Condition Événementielle)

Ce paragraphe spécifie les extensions du service AlterEventConditionMonitoring.

7.2.5.4.1 Extensions du paramètre

Le service AlterEventConditionMonitoring doit être étendu pour inclure la spécification de la valeur de l’attribut Enrichissement d’Affichage de l’objet Condition Événementielle. La structure des extensions du paramètre AlterEventConditionMonitoring est définie dans le tableau 3.

Tableau 3 – Extensions du paramètre AlterEventConditionMonitoring

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Enrichissement d’Affichage	S	S (=)			
Chaîne d’Enrichissement d’Affichage	S	S (=)			DES
Indice d’Enrichissement d’Affichage	S	S (=)			DEI
Pas d’Enrichissement	S	S (=)			
Affichage Inchangé	S	S (=)			

7.2.5.4.1.1 Enrichissement de l’Affichage

La sélection de ce paramètre indique que l’attribut Enrichissement d’Affichage de la Condition Événementielle doit être modifiée par ce service. Si ce paramètre est sélectionné, l’un des paramètres suivants doit apparaître.

7.2.5.4.1.1.1 Chaîne d’Enrichissement d’Affichage

Ce paramètre, du type chaîne de caractères, est la forme textuelle du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DES a été pris en compte.

7.2.5.4.1.1.2 Indice d’Enrichissement d’Affichage

Ce paramètre, du type entier, est la forme numérique du paramètre d’Enregistrement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DEI a été pris en compte.

7.2.5.3.8 Display Enhancement

If the value of the Display Enhancement Class attribute of the Event Condition object is equal to TEXT, the enhancementString choice shall be selected. If the value of the Display Enhancement Class attribute of the Event Condition object is equal to NUMBER, the Enhancement Index choice shall be selected. Otherwise the noEnhancement choice shall be selected.

7.2.5.4 AlterEventConditionMonitoring service

This subclause specifies the extensions to the AlterEventConditionMonitoring service.

7.2.5.4.1 Parameter extensions

The AlterEventConditionMonitoring service shall be extended to include specification of the value of the Display Enhancement attribute of the Event Condition object. The structure of the AlterEventConditionMonitoring parameter extensions is specified in table 3.

Table 3 – AlterEventConditionMonitoring parameter extensions

Parameter name	Req	Ind	Resp	Cnf	CBB
Display Enhancement	S	S(=)			
Display Enhancement string	S	S(=)			DES
Display Enhancement index	S	S(=)			DEI
No Enhancement	S	S(=)			
Unchanged Display	S	S(=)			

7.2.5.4.1.1 Display Enhancement

Selection of this parameter indicates that the Display Enhancement attribute of the Event Condition shall be altered by this service. If this parameter is selected, one of the following parameters shall appear.

7.2.5.4.1.1.1 Display Enhancement string

This parameter, of type character string, is the string form of the Display Enhancement parameter. This selection may be made only if the DES CBB has been negotiated.

7.2.5.4.1.1.2 Display Enhancement index

This parameter, of type integer, is the numeric form of the Display Enhancement parameter. This selection may be made only if the DEI CBB has been negotiated.

7.2.5.4.1.1.3 *Pas d'Enrichissement*

Ce paramètre, du type à valeur NULLE, indique qu'aucun Enrichissement d'Affichage n'est présent. Ce paramètre doit être sélectionné si, ni DES, ni DEI n'ont été pris en compte.

7.2.5.4.1.2 *Affichage Inchangé*

Si le paramètre est sélectionné, la valeur en cours de l'attribut Enrichissement d'Affichage de la Condition Événementielle ne doit pas être modifiée.

7.2.5.4.2 *Extensions de procédure de service*

Si l'Enrichissement d'Affichage est sélectionné, la valeur de l'attribut Enrichissement d'Affichage de l'objet Condition Événementielle doit être modifiée. Si la chaîne d'Enrichissement d'Affichage est sélectionnée, l'attribut Classe d'Enrichissement d'Affichage de la Condition Événementielle doit être mis sur TEXTE et l'attribut Enrichissement d'Affichage doit être mis sur la valeur du paramètre chaîne de l'Enrichissement d'Affichage. Si l'indice Enrichissement d'Affichage est sélectionné, l'attribut Classe d'Enrichissement d'Affichage de la Condition Événementielle doit être mis sur NOMBRE et l'attribut Enrichissement d'Affichage doit être mis sur la valeur du paramètre indice de l'Enrichissement d'Affichage. Autrement, la Classe d'Enrichissement d'Affichage doit être mise sur INDEFINIE.

7.2.5.4.3 *Extensions de protocole*

L'extension de l'argument de service `AlterEventConditionMonitoring` doit être `CS-AlterEventConditionMonitoring-Request` (Demande-Modifier Surveillance de Condition Événementielle-NA).

```
CS-AlterEventConditionMonitoring-Request ::= SEQUENCE {
    changeDisplay CHOICE {
        enhancementString [0] IMPLICIT VisibleString,
        enhancementIndex [1] IMPLICIT INTEGER,
        noEnhancement [2] NULL
    } OPTIONAL }

```

La sélection d'Enrichissement d'Affichage pour l'Option d'Affichage doit être indiquée en incluant le champ `changeDisplay` (changement d'Affichage) dans la PDU. Si `Unchanged Display` (Affichage Inchangé) est sélectionné, le champ `changeDisplay` n'a pas lieu d'être.

7.2.5.5 *Service DefineEventEnrollment (Définir Enveloppe Événementielle)*

Ce paragraphe spécifie les extensions du service `DefineEventEnrollment`.

7.2.5.5.1 *Extensions du paramètre*

Le service `DefineEventEnrollment` doit être étendu pour inclure la spécification de l'attribut Enrichissement d'Affichage de l'objet Enveloppe Événementielle. La structure des extensions du paramètre `DefineEventEnrollment` est définie dans le tableau 4.

7.2.5.4.1.1.3 *No Enhancement*

This parameter, of type NULL, specifies that no Display Enhancement is present. This parameter shall be selected if neither DES nor DEI has been negotiated.

7.2.5.4.1.2 *Unchanged Display*

If this parameter is selected, the Display Enhancement attribute of the Event Condition shall not be changed from its present value.

7.2.5.4.2 *Service procedure extensions*

If the Display Enhancement selection is made, the value of the Display Enhancement attribute of the Event Condition object shall be altered. If the Display Enhancement string is selected, the Display Enhancement Class attribute of the Event Condition object shall be set to TEXT and the Display Enhancement attribute shall be set to the value of the Display Enhancement string parameter. If the Display Enhancement index is selected, the Display Enhancement Class attribute of the Event Condition object shall be set to NUMBER and the Display Enhancement attribute shall be set to the value of the Display Enhancement index parameter. Otherwise, the Display Enhancement Class shall be set to UNDEFINED.

7.2.5.4.3 *Protocol extensions*

The extension to the AlterEventConditionMonitoring service argument shall be the CS-AlterEventConditionMonitoring-Request.

```
CS-AlterEventConditionMonitoring-Request ::= SEQUENCE {
    changeDisplay      CHOICE {
        enhancementString [0] IMPLICIT VisibleString,
        enhancementIndex  [1] IMPLICIT INTEGER,
        noEnhancement     [2] NULL
    } OPTIONAL
}
```

The choice of Display Enhancement for Display Option shall be indicated by the inclusion of the changeDisplay field in the PDU. If Unchanged Display is chosen, the changeDisplay field shall not occur.

7.2.5.5 *DefineEventEnrollment service*

This subclause specifies the extensions to the DefineEventEnrollment service.

7.2.5.5.1 *Parameter extensions*

The DefineEventEnrollment service shall be extended to include specification of the Display Enhancement attribute of the Event Enrollment object. The structure of the DefineEventEnrollment parameter extensions is specified in table 4.

Tableau 4 – Extensions du paramètre DefineEventEnrollment

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Enrichissement d’Affichage	M	M (=)			
Chaîne d’Enrichissement d’Affichage	S	S (=)			DES
Indice d’Enrichissement d’Affichage	S	S (=)			DEI
Pas d’Enrichissement	S	S (=)			

7.2.5.5.1.1 Enrichissement d’Affichage

La valeur de ce paramètre indique le type d’Enrichissement d’Affichage désiré. Selon sa valeur, l’un des paramètres suivants doit être sélectionné.

7.2.5.5.1.1.1 Chaîne d’Enrichissement d’Affichage

Ce paramètre, du type chaîne de caractères, est la forme textuelle du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DES a été pris en compte.

7.2.5.5.1.1.2 Indice d’Enrichissement d’Affichage

Ce paramètre, du type entier, est la forme numérique du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DEI a été pris en compte.

7.2.5.5.1.1.3 Pas d’Enrichissement

Ce paramètre, du type à valeur NULLE, indique qu’aucun Enrichissement d’Affichage n’est présent. Ce paramètre doit uniquement être sélectionné si ni DES, ni DEI n’ont été pris en compte.

7.2.5.5.2 Extensions de procédure du service

La procédure de service du paragraphe 15.12.2 de l’ISO/IEC 9506-1 doit être exécutée. Si la chaîne Enrichissement d’Affichage a été sélectionnée, la valeur de l’attribut Classe d’Enrichissement d’Affichage de l’Enveloppe Événementielle doit être TEXTE et l’attribut Enrichissement d’Affichage doit être mis sur la valeur du paramètre chaîne d’Enrichissement d’Affichage. Si l’indice Enrichissement d’Affichage est sélectionné, l’attribut Classe Enrichissement d’Affichage de l’Enveloppe Événementielle doit être NOMBRE et l’attribut Enrichissement d’Affichage doit être mis sur la valeur du paramètre indice d’Enrichissement d’Affichage. Autrement, l’attribut Classe d’Enrichissement d’Affichage doit être mis sur INDEFINI.

7.2.5.5.3 Extensions de protocole

Les extensions de l’argument de service DefineEventEnrollment doivent être CS-DefineEventEnrollment-Request (Demande-Définir Enveloppe Événementielle-NA).

Table 4 – DefineEventEnrollment parameter extensions

Parameter name	Req	Ind	Rep	Cnf	CBB
Display Enhancement	M	M(=)			
Display Enhancement string	S	S(=)			DES
Display Enhancement index	S	S(=)			DEI
No Enhancement	S	S(=)			

7.2.5.5.1.1 Display Enhancement

The value of this parameter indicates the type of Display Enhancement desired. Depending on its value, one of the following parameters shall be selected.

7.2.5.5.1.1.1 Display Enhancement string

This parameter, of type character string, is the string form of the Display Enhancement parameter. This selection may be made only if the DES CBB has been negotiated.

7.2.5.5.1.1.2 Display Enhancement index

This parameter, of type integer, is the numeric form of the Display Enhancement parameter. This selection may be made only if the DEI CBB has been negotiated.

7.2.5.5.1.1.3 No Enhancement

This parameter, of type NULL, specifies that no Display Enhancement is present. This parameter shall be selected if neither DES nor DEI has been negotiated.

7.2.5.5.2 Service procedure extensions

The service procedure of 15.12.2 of ISO/IEC 9506-1 shall be performed. If the Display Enhancement string has been selected, the value of the Display Enhancement Class attribute of the Event Enrollment shall be TEXT and the Display Enhancement attribute shall be set to the value of the Display Enhancement string parameter. If the Display Enhancement index is selected, the Display Enhancement Class attribute of the Event Enrollment shall be NUMBER and the Display Enhancement attribute shall be set to the value of the Display Enhancement index parameter. Otherwise, the Display Enhancement Class attribute shall be set to UNDEFINED.

7.2.5.5.3 Protocol extensions

The extensions to the DefineEventEnrollment service argument shall be the CS-DefineEventEnrollment-Request.

```

CS-DefineEventEnrollment-Request ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
    
```

NOTE – Etant donné la manière dont Confirmed-RequestPDU (PDU Demande-Confirmée) est prescrit dans l'ISO/IEC 9506-2, une valeur NULLE spécifiée comme extension de la Norme d'Accompagnement n'est pas transmise. De ce fait, le paramètre est considéré non- inclus dans le protocole. Il est cependant prescrit de préciser la valeur NULLE au niveau de l'interface de service.

7.2.5.6 Service GetEventEnrollmentAttributes (ObtenirAttributsd'EnveloppeEvénementielle)

Ce paragraphe spécifie l'extension du service GetEventEnrollmentAttributes.

7.2.5.6.1 Extensions du paramètre

La réponse au service GetEventEnrollmentAttributes doit être étendue pour inclure le paramètre Enrichissement d'Affichage. La structure des extensions du paramètre Get Event EnrollmentAttributes est définie dans le tableau 5.

Tableau 5 – Extensions du paramètre GetEventEnrollmentAttributes

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Enrichissement d'Affichage			M	M(=)	
Chaîne d'Enrichissement d'Affichage			S	S(=)	
Indice d'Enrichissement d'Affichage			S	S(=)	
Pas d'Enrichissement			S	S(=)	

7.2.5.6.1.1 Enrichissement d'Affichage

La valeur de ce paramètre indique le type d'Enrichissement d'Affichage désiré. Selon sa valeur, l'un des paramètres suivants doit être sélectionné.

7.2.5.6.1.1.1 Chaîne d'Enrichissement d'Affichage

Ce paramètre, du type chaîne de caractères, est la forme textuelle du paramètre Enrichissement d'Affichage.

7.2.5.6.1.1.2 Indice d'Enrichissement d'Affichage

Ce paramètre, du type entier, est la forme numérique du paramètre Enrichissement d'Affichage.

7.2.5.6.1.1.3 Pas d'Enrichissement

Ce paramètre, du type à valeur NULLE, spécifie qu'aucun Enrichissement d'Affichage n'est présent.

```

CS-DefineEventEnrollment-Request ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}

```

NOTE – As a result of the way in which Confirmed-RequestPDU is specified in ISO/IEC 9506-2, a NULL specified as a Companion Standard extension is not transmitted. The effect is as though the parameter were not included in the protocol. Specification of the NULL is required at the service interface, however.

7.2.5.6 GetEventEnrollmentAttributes service

This subclause specifies the extension of the GetEventEnrollmentAttributes service.

7.2.5.6.1 Parameter extensions

The GetEventEnrollmentAttributes service response shall be extended to include the Display Enhancement parameter. The structure of the GetEventEnrollmentAttributes parameter extensions is specified in table 5.

Table 5 – GetEventEnrollmentAttributes parameter extensions

Parameter name	Req	Ind	Rep	Cnf	CBB
Display Enhancement	M	M(=)			
Display Enhancement string	S	S(=)			
Display Enhancement index	S	S(=)			
No Enhancement	S	S(=)			

7.2.5.6.1.1 Display Enhancement

The value of this parameter indicates the type of Display Enhancement desired. Depending on its value, one of the following parameters shall be selected.

7.2.5.6.1.1.1 Display Enhancement string

This parameter, of type character string, is the string form of the Display Enhancement parameter.

7.2.5.6.1.1.2 Display Enhancement index

This parameter, of type integer, is the numeric form of the Display Enhancement parameter.

7.2.5.6.1.1.3 No Enhancement

This parameter, of type NULL, specifies that no Display Enhancement is present.

7.2.5.6.2 Extensions de procédure de service

La procédure de service du 15.14.2 de l'ISO/IEC 9506-1 doit être exécutée. Si l'attribut Classe d'Enrichissement d'Affichage de l'Enveloppe Événementielle a la valeur TEXTE, le paramètre chaîne d'Enrichissement d'Affichage doit être sélectionné. Si l'attribut Classe d'Enrichissement d'Affichage a la valeur NOMBRE, l'indice d'Enrichissement d'Affichage doit être sélectionné. Si l'attribut Classe d'Enrichissement d'Affichage a la valeur INDEFINIE, le paramètre Pas d'Enrichissement doit être sélectionné.

7.2.5.6.3 Extensions du protocole

Les extensions du protocole de réponse au service GetEventEnrollmentAttributes doivent être CS-GetEventEnrollmentAttributes-Response (Réponse-Obtenir Attributs d'Enveloppe Événementielle-NA).

```
CS-GetEventEnrollmentAttributes-Response ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

NOTE – Etant donné la manière dont Confirmed-ResponsePDU (PDU Réponse-Confirmée) est prescrit dans l'ISO/IEC 9506-2, une valeur NULLE spécifiée comme extension de la Norme d'Accompagnement n'est pas transmise. De ce fait, le paramètre est considéré non inclus dans le protocole. Il est cependant prescrit de préciser la valeur NULLE au niveau de l'interface de service.

7.2.5.7 Service AlterEventEnrollment (Modifier Enveloppe Événementielle)

Ce paragraphe spécifie l'extension du service AlterEventEnrollment.

7.2.5.7.1 Extensions du paramètre

Le service AlterEventEnrollment doit être étendu pour inclure la spécification de l'attribut Enrichissement d'Affichage de l'objet Enveloppe Événementielle. La structure des extensions du paramètre AlterEventEnrollment est indiquée dans le tableau 6.

Tableau 6 – Extensions du paramètre AlterEventEnrollment

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Enrichissement d'Affichage	S	S (=)			
Chaîne d'Enrichissement d'Affichage	S	S (=)			DES
Indice d'Enrichissement d'Affichage	S	S (=)			DEI
Pas d'Enrichissement	S	S (=)			
Affichage inchangé	S	S (=)			

7.2.5.6.2 Service procedure extensions

The service procedure of 15.14.2 of ISO/IEC 9506-1 shall be performed. If the Display Enhancement Class attribute of the Event Enrollment has the value TEXT, the Display Enhancement string parameter shall be selected. If the Display Enhancement Class attribute has the value NUMBER, the Display Enhancement index shall be selected. If the Display Enhancement Class attribute has the value UNDEFINED, the No Enhancement parameter shall be selected.

7.2.5.6.3 Protocol extensions

The extensions to the GetEventEnrollmentAttributes service response protocol shall be the CS-GetEventEnrollmentAttributes-Response.

```
CS-GetEventEnrollmentAttributes-Response ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

NOTE – As a result of the way in which Confirmed-ResponsePDU is specified in ISO/IEC 9506-2, a NULL specified as a Companion Standard extension is not transmitted. The effect is as though the parameter were not included in the protocol. Specification of the NULL is required at the service interface, however.

7.2.5.7 AlterEventEnrollment service

This subclause specifies the extension of the AlterEventEnrollment service.

7.2.5.7.1 Parameter extensions

The AlterEventEnrollment service shall be extended to include specification of the Display Enhancement attribute of the Event Enrollment object. The structure of the AlterEvent-Enrollment parameter extensions is specified in table 6.

Table 6 – AlterEventEnrollment parameter extensions

Parameter number	Req	Ind	Rsp	Cnf	CBB
Display Enhancement	S	S (=)			
Display Enhancement string	S	S (=)			DES
Display Enhancement index	S	S (=)			DEI
No Enhancement	S	S (=)			
Unchanged Display	S	S (=)			

7.2.5.7.1.1 *Enrichissement d’Affichage*

La sélection de ce paramètre indique que l’attribut Enrichissement d’Affichage de la Condition Événementielle doit être modifié par ce service. Si ce paramètre est sélectionné, l’un des paramètres suivants doit apparaître.

7.2.5.7.1.1.1 *Chaîne d’Enrichissement d’Affichage*

Ce paramètre, du type chaîne de caractères, est la forme textuelle du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DES a été pris en compte.

7.2.5.7.1.1.2 *Indice Enrichissement d’Affichage*

Ce paramètre, du type entier, est la forme numérique du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DEI a été pris en compte.

7.2.5.7.1.1.3 *Pas d’Enrichissement*

Ce paramètre, du type à valeur NULLE, spécifie qu’aucun Enrichissement d’Affichage n’est présent. Ce paramètre doit être sélectionné si ni DES, ni DEI n’ont été pris en compte.

7.2.5.7.1.2 *Affichage Inchangé*

Si ce paramètre est sélectionné, la valeur en cours de l’attribut Enrichissement d’Affichage de l’Enveloppe Événementielle ne doit pas être modifiée.

7.2.5.7.2 *Extensions de procédure de service*

Si l’Enrichissement d’Affichage est sélectionné, la valeur de l’attribut Enrichissement d’Affichage de l’objet Enveloppe Événementielle doit être modifiée. Si la chaîne d’Enrichissement d’Affichage est sélectionnée, l’attribut Classe d’Enrichissement d’Affichage de l’Enveloppe Événementielle doit être mis sur TEXTE et l’attribut Enrichissement d’Affichage doit être mis sur la valeur du paramètre chaîne d’Enrichissement d’Affichage. Si l’indice d’Enrichissement d’Affichage est sélectionné, l’attribut Classe d’Enrichissement d’Affichage de l’Enveloppe Événementielle doit être mis sur NOMBRE et l’attribut Enrichissement d’Affichage doit être mis sur la valeur du paramètre indice d’Enrichissement d’Affichage. Autrement, la Classe Enrichissement d’Affichage doit être mise sur INDEFINIE.

7.2.5.7.3 *Extensions de protocole*

L’extension de l’argument de service AlterEventEnrollment doit être leCS-Alter Event Enrollment-Request (Demande-Modifier Enveloppe Événementielle-NA).

```
CS-AlterEventEnrollment-Request ::= SEQUENCE {
    changeDisplay      CHOICE {
        enhancementString [0] IMPLICIT VisibleString,
        enhancementIndex  [1] IMPLICIT INTEGER,
        noEnhancement     [2] NULL
    } OPTIONAL }
```

7.2.5.7.1.1 *Display Enhancement*

Selection of this parameter indicates that the Display Enhancement attribute of the Event Condition shall be altered by this service. If this parameter is selected, one of the following parameters shall appear.

7.2.5.7.1.1.1 *Display Enhancement string*

This parameter, of type character string, is the string form of the Display Enhancement parameter. This selection may be made only if the DES CBB has been negotiated.

7.2.5.7.1.1.2 *Display Enhancement index*

This parameter, of type integer, is the numeric form of the Display Enhancement parameter. This selection may be made only if the DEI CBB has been negotiated.

7.2.5.7.1.1.3 *No Enhancement*

This parameter, of type NULL, specifies that no Display Enhancement is present. This parameter shall be selected if neither DES nor DEI has been negotiated.

7.2.5.7.1.2 *Unchanged Display*

If the parameter is selected, the Display Enhancement attribute of the Event Enrollment shall not be changed from its present value.

7.2.5.7.2 *Service procedure extensions*

If the Display Enhancement selection is made, the value of the Display Enhancement attribute of the Event Enrollment object shall be altered. If the Display Enhancement string is selected, the Display Enhancement Class attribute of the Event Enrollment shall be set to TEXT and the Display Enhancement attribute shall be set to the value of the Display Enhancement string parameter. If the Display Enhancement index is selected, the Display Enhancement Class attribute of the Event Enrollment shall be set to NUMBER and the Display Enhancement attribute shall be set to the value of the Display Enhancement index parameter. Otherwise, the Display Enhancement Class shall be set to UNDEFINED.

7.2.5.7.3 *Protocol extensions*

The extension to the AlterEventEnrollment service argument shall be the CS-AlterEventEnrollment-Request.

```
CS-AlterEventEnrollment-Request ::= SEQUENCE {
  changeDisplay      CHOICE {
    enhancementString [0] IMPLICIT VisibleString,
    enhancementIndex  [1] IMPLICIT INTEGER,
    noEnhancement     [2] NULL
  } OPTIONAL }
```

La sélection Enrichissement d’Affichage pour l’Option d’Affichage doit être indiquée par l’inclusion du champ changeDisplay (Changement d’Affichage) dans la PDU. Si Unchanged Display (Affichage Inchangé) est sélectionné, le champ changeDisplay ne doit pas apparaître.

7.2.5.8 Service EventNotification (Notification d’Événement)

Le présent paragraphe spécifie les extensions du service EventNotification.

7.2.5.8.1 Extensions du paramètre

Le service EventNotification doit être étendu pour inclure la spécification de l’attribut Enrichissement d’Affichage de l’objet Condition Événementielle. La structure des extensions du paramètre EventNotification est définie dans le tableau 7.

Tableau 7 – Extensions du paramètre EventNotification

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Enrichissement d’Affichage	M	M(=)			
Chaine d’Enrichissement d’Affichage	S	S(=)			DES
Indice d’Enrichissement d’Affichage	S	S(=)			DEI
Pas d’Enrichissement	S	S(=)			

7.2.5.8.1.1 Enrichissement d’Affichage

La valeur de ce paramètre indique le type d’Enrichissement d’Affichage désiré. Selon sa valeur, l’un des paramètres suivants doit être sélectionné.

7.2.5.8.1.1.1 Chaine d’Enrichissement d’Affichage

Ce paramètre, du type chaine de caractères, est la forme textuelle du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DES a été pris en compte.

7.2.5.8.1.1.2 Indice d’Enrichissement d’Affichage

Ce paramètre, du type entier, est la forme numérique du paramètre Enrichissement d’Affichage. Cette sélection peut uniquement être effectuée si le CBB DEI a été pris en compte.

7.2.5.8.1.1.3 Pas d’Enrichissement

Ce paramètre, du type à valeur NULLE, indique qu’aucun Enrichissement d’Affichage n’est présent. Ce paramètre doit être sélectionné si ni DES, ni DEI n’ont été pris en compte.

7.2.5.8.2 Extensions de procédure du service

La procédure de service du 15.17.2 de l’ISO/IEC 9506-1 doit être exécutée, en incluant la valeur du paramètre Enrichissement d’Affichage dans la PDU de réponse. Si la valeur de

The choice of Display Enhancement for Display Option shall be indicated by the inclusion of the changeDisplay field in the PDU. If Unchanged Display is chosen, the changeDisplay field shall not occur.

7.2.5.8 EventNotification service

This subclause specifies the extensions to the EventNotification service.

7.2.5.8.1 Parameter extensions

The EventNotification service shall be extended to include specification of the Display Enhancement attribute of the Event Condition object. The structure of the EventNotification parameter extensions is specified in table 7.

Table 7 – EventNotification parameter extensions

Parameter name	Req	Ind	Rep	Cnf	CBB
Display Enhancement	M	M(=)			
Display Enhancement string	S	S(=)			DES
Display Enhancement index	S	S(=)			DEI
No Enhancement	S	S(=)			

7.2.5.8.1.1 Display Enhancement

The value of this parameter indicates the type of Display Enhancement desired. Depending on its value, one of the following parameters shall be selected.

7.2.5.8.1.1.1 Display Enhancement string

This parameter, of type character string, is the string form of the Display Enhancement parameter. This selection may be made only if the DES CBB has been negotiated.

7.2.5.8.1.1.2 Display Enhancement index

This parameter, of type integer, is the numeric form of the Display Enhancement parameter. This selection may be made only if the DEI CBB has been negotiated.

7.2.5.8.1.1.3 No Enhancement

This parameter, of type NULL, specifies that no Display Enhancement is present. This parameter shall be selected if neither DES nor DEI has been negotiated.

7.2.5.8.2 Service procedure extensions

The service procedure of 15.17.2 of ISO/IEC 9506-1 shall be performed, including the value of the Display Enhancement parameter in the response PDU. If the value of the

l'attribut Classe d'Enrichissement d'Affichage de l'objet Enveloppe Événementielle référencée dans la demande Condition Événementielle n'est pas égale à INDEFINIE, la valeur du paramètre Enrichissement d'Affichage doit résulter de la valeur de l'attribut Enrichissement d'Affichage de l'objet Enveloppe Événementielle. Autrement, la valeur du paramètre Enrichissement d'Affichage doit résulter de la valeur de l'attribut Enrichissement d'Affichage de l'objet Condition Événementielle référencé dans la demande Notification d'Événement.

7.2.5.8.3 Extensions du protocole

Les extensions de l'argument du service EventNotification doivent être CS-EventNotification (Notification d'Événement-NA).

```
CS-EventNotification ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

NOTE – Etant donné la manière dont Unconfirmed-PDU (PDU Nonconfirmée) est prescrit dans l'ISO/IEC 9506-2, une valeur NULLE spécifiée comme extension de la Norme d'Accompagnement n'est pas transmise. De ce fait, le paramètre est considéré non inclus dans le protocole. Il est cependant prescrit de préciser la valeur NULLE au niveau de l'interface de service.

7.2.5.9 Paramètre Additional Detail (Élément Supplémentaire) de Condition Événementielle

Le paramètre Additional Detail qui est utilisé dans le service Get Alarm Summary (Obtenir Relevé d'Alarme) et dans le service Get Alarm Enrollment Summary (Obtenir Relevé d'Enveloppe d'Alarme) est prescrit comme étant le paramètre d'Enrichissement d'Affichage défini en 6.4.1.1. La valeur de ce paramètre doit résulter des attributs de l'objet Condition Événementielle référencé dans la réponse de service correspondante.

7.2.5.9.1 Protocole du paramètre Additional Detail

La présente partie de l'ISO/IEC 9506 définit le paramètre EN-Additional-Detail comme suit:

```
EN-Additional-Detail ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

7.2.6 Autres paramètres

Les paramètres ci-dessous, prescrits pour le Module Général MMS et constituant des Extensions de Norme d'Accompagnement pour les services MMS ne sont pas utilisés dans la Norme d'Accompagnement de Contrôle-Commande de Procédé et sont mis sur la valeur NULLE.

```
CS-Status-Request ::= NULL
CS-Input-Request  ::= NULL
CS-Output-Request ::= NULL
```

Display Enhancement Class attribute of the Event Enrollment object referenced in the Event Condition request is not equal to UNDEFINED, the value of the Display Enhancement parameter shall be derived from the value of the Display Enhancement attribute of the Event Enrollment object. Otherwise, the value of the Display Enhancement parameter shall be derived from the value of the Display Enhancement attribute of the Event Condition object referenced in the Event Notification request.

7.2.5.8.3 Protocol extensions

The extensions to the EventNotification service argument shall be the CS-Event Notification.

```
CS-EventNotification ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

NOTE – As a result of the way in which Unconfirmed-PDU is specified in ISO/IEC 9506-2, a NULL specified as a Companion Standard extension is not transmitted. The effect is as though the parameter were not included in the protocol. Specification of the NULL is required at the service interface, however.

7.2.5.9 Event Condition Additional Detail

The Additional Detail parameter that is used in the Get Alarm Summary service and in the Get Alarm Enrollment Summary service is specified to be the Display Enhancement parameter defined in 6.4.1.1. The value of this parameter shall be derived from the attributes of the Event Condition object referenced in the respective service response.

7.2.5.9.1 Additional Detail protocol

This part of ISO/IEC 9506 defines the EN-Additional-Detail production as follows:

```
EN-Additional-Detail ::= [0] Choice {
    enhancementString      [0] IMPLICIT VisibleString,
    enhancementIndex       [1] IMPLICIT INTEGER,
    noEnhancement          NULL
}
```

7.2.6 Other productions

The following productions, required for the MMS General Module and representing Companion Standard Extensions to the MMS services, are not used in the Process Control Companion Standard, and are set equal to NULL.

```
CS-Status-Request ::= NULL
CS-Input-Request  ::= NULL
CS-Output-Request ::= NULL
```

CS-InitiateDownloadSequence-Request::= NULL
CS-DownloadSegment-Request::= NULL
CS-TerminateDownloadSequence-Request::= NULL
CS-InitiateUploadSequence-Request::= NULL
CS-UploadSegment-Request::= NULL
CS-TerminateUploadSegment-Request::= NULL
CS-RequestDomainDownload-Request::= NULL
CS-RequestDomainUpload-Request::= NULL
CS-LoadDomainContent-Request::= NULL
CS-StoreDomainContent-Request::= NULL
CS-DeleteDomain-Request::= NULL
CS-GetDomainAttributes-Request::= NULL
CS-CreateProgramInvocation-Request::= NULL
CS-DeleteProgramInvocation-Request::= NULL
CS-Start-Request::= NULL
CS-Stop-Request::= NULL
CS-Reset-Request::= NULL
CS-Resume-Request::= NULL
CS-Kill-Request::= NULL
CS-GetProgramInvocationAttributes-Request::= NULL
CS-DeleteEventCondition-Request::= NULL
CS-GetEventConditionAttributes-Request::= NULL
CS-ReportEventConditionStatus-Request::= NULL
CS-TriggerEvent-Request::= NULL
CS-DefineEventAction-Request::= NULL
CS-DeleteEventAction-Request::= NULL
CS-GetEventActionAttributes-Request::= NULL
CS-ReportEventActionStatus-Request::= NULL
CS-DeleteEventEnrollment-Request::= NULL
CS-ReportEventEnrollmentStatus-Request::= NULL
CS-GetEventEnrollmentAttributes-Request::= NULL
CS-AcknowledgeEventNotification-Request::= NULL
CS-GetAlarmSummary-Request::= NULL
CS-GetAlarmEnrollmentSummary-Request::= NULL
CS-ReadJournal-Request::= NULL
CS-WriteJournal-Request::= NULL
CS-InitializeJournal-Request::= NULL
CS-ReportJournalStatus-Request::= NULL
CS-CreateJournal-Request::= NULL
CS-DeleteJournal-Request::= NULL
CS-GetCapabilityList-Request::= NULL
CS-Status-Response::= NULL
CS-Input-Response::= NULL
CS-Output-Response::= NULL
CS-InitiateDownloadSequence-Response::= NULL

CS-InitiateDownloadSequence-Request ::= NULL
CS-DownloadSegment-Request ::= NULL
CS-TerminateDownloadSequence-Request ::= NULL
CS-InitiateUploadSequence-Request ::= NULL
CS-UploadSegment-Request ::= NULL
CS-TerminateUploadSequence-Request ::= NULL
CS-RequestDomainDownload-Request ::= NULL
CS-RequestDomainUpload-Request ::= NULL
CS-LoadDomainContent-Request ::= NULL
CS-StoreDomainContent-Request ::= NULL
CS-DeleteDomain-Request ::= NULL
CS-GetDomainAttributes-Request ::= NULL
CS-CreateProgramInvocation-Request ::= NULL
CS-DeleteProgramInvocation-Request ::= NULL
CS-Start-Request ::= NULL
CS-Stop-Request ::= NULL
CS-Reset-Request ::= NULL
CS-Resume-Request ::= NULL
CS-Kill-Request ::= NULL
CS-GetProgramInvocationAttributes-Request ::= NULL
CS-DeleteEventCondition-Request ::= NULL
CS-GetEventConditionAttributes-Request ::= NULL
CS-ReportEventConditionStatus-Request ::= NULL
CS-TriggerEvent-Request ::= NULL
CS-DefineEventAction-Request ::= NULL
CS-DeleteEventAction-Request ::= NULL
CS-GetEventActionAttributes-Request ::= NULL
CS-ReportEventActionStatus-Request ::= NULL
CS-DeleteEventEnrollment-Request ::= NULL
CS-ReportEventEnrollmentStatus-Request ::= NULL
CS-GetEventEnrollmentAttributes-Request ::= NULL
CS-AcknowledgeEventNotification-Request ::= NULL
CS-GetAlarmSummary-Request ::= NULL
CS-GetAlarmEnrollmentSummary-Request ::= NULL
CS-ReadJournal-Request ::= NULL
CS-WriteJournal-Request ::= NULL
CS-InitializeJournal-Request ::= NULL
CS-ReportJournalStatus-Request ::= NULL
CS-CreateJournal-Request ::= NULL
CS-DeleteJournal-Request ::= NULL
CS-GetCapabilityList-Request ::= NULL
CS-Status-Response ::= NULL
CS-Input-Response ::= NULL
CS-Output-Response ::= NULL
CS-InitiateDownloadSequence-Response ::= NULL

CS-DownloadSegment-Response::= NULL
CS-TerminateDownloadSequence-Response::= NULL
CS-InitiateUploadSequence-Response::= NULL
CS-UploadSegment-Response::= NULL
CS-TerminateUploadSequence-Response::= NULL
CS-RequestDomainDownload-Response::= NULL
CS-RequestDomainUpload-Response::= NULL
CS-LoadDomainContent-Response::= NULL
CS-StoreDomainContent-Response::= NULL
CS-DeleteDomain-Response::= NULL
CS-CreateProgramInvocation-Response::= NULL
CS-DeleteProgramInvocation-Response::= NULL
CS-Start-Response::= NULL
CS-Stop-Response::= NULL
CS-Resume-Response::= NULL
CS-Reset-Response::= NULL
CS-Kill-Response::= NULL
CS-DefineEventCondition-Response::= NULL
CS-DeleteEventCondition-Response::= NULL
CS-GetEventConditionAttributes-Response::= NULL
CS-ReportEventConditionStatus-Response::= NULL
CS-AlterEventConditionMonitoring-Response::= NULL
CS-TriggerEvent-Response::= NULL
CS-DefineEventAction-Response::= NULL
CS-DeleteEventAction-Response::= NULL
CS-GetEventActionAttributes-Response::= NULL
CS-ReportEventActionStatus-Response::= NULL
CS-DefineEventEnrollment-Response::= NULL
CS-DeleteEventEnrollment-Response::= NULL
CS-AlterEventEnrollment-Response::= NULL
CS-ReportEventEnrollmentStatus-Response::= NULL
CS-AcknowledgeEventNotification-Response::= NULL
CS-GetAlarmSummary-Response::= NULL
CS-GetAlarmEnrollmentSummary-Response::= NULL
CS-ReadJournal-Response::= NULL
CS-WriteJournal-Response::= NULL
CS-InitializeJournal-Response::= NULL
CS-ReportJournalStatus-Response::= NULL
CS-CreateJournal-Response::= NULL
CS-DeleteJournal-Response::= NULL
CS-GetCapabilityList-Response::= NULL
CS-UnsolicitedStatus-Response::= NULL
AdditionalService-Error::= NULL
AdditionalUnconfirmedService::= NULL
EE-Additional-Detail::= NULL

CS-DownloadSegment-Response ::= NULL
CS-TerminateDownloadSequence-Response ::= NULL
CS-InitiateUploadSequence-Response ::= NULL
CS-UploadSegment-Response ::= NULL
CS-TerminateUploadSequence-Response ::= NULL
CS-RequestDomainDownload-Response ::= NULL
CS-RequestDomainUpload-Response ::= NULL
CS-LoadDomainContent-Response ::= NULL
CS-StoreDomainContent-Response ::= NULL
CS-DeleteDomain-Response ::= NULL
CS-CreateProgramInvocation-Response ::= NULL
CS-DeleteProgramInvocation-Response ::= NULL
CS-Start-Response ::= NULL
CS-Stop-Response ::= NULL
CS-Resume-Response ::= NULL
CS-Reset-Response ::= NULL
CS-Kill-Response ::= NULL
CS-DefineEventCondition-Response ::= NULL
CS-DeleteEventCondition-Response ::= NULL
CS-GetEventConditionAttributes-Response ::= NULL
CS-ReportEventConditionStatus-Response ::= NULL
CS-AlterEventConditionMonitoring-Response ::= NULL
CS-TriggerEvent-Response ::= NULL
CS-DefineEventAction-Response ::= NULL
CS-DeleteEventAction-Response ::= NULL
CS-GetEventActionAttributes-Response ::= NULL
CS-ReportEventActionStatus-Response ::= NULL
CS-DefineEventEnrollment-Response ::= NULL
CS-DeleteEventEnrollment-Response ::= NULL
CS-AlterEventEnrollment-Response ::= NULL
CS-ReportEventEnrollmentStatus-Response ::= NULL
CS-AcknowledgeEventNotification-Response ::= NULL
CS-GetAlarmSummary-Response ::= NULL
CS-GetAlarmEnrollmentSummary-Response ::= NULL
CS-ReadJournal-Response ::= NULL
CS-WriteJournal-Response ::= NULL
CS-InitializeJournal-Response ::= NULL
CS-ReportJournalStatus-Response ::= NULL
CS-CreateJournal-Response ::= NULL
CS-DeleteJournal-Response ::= NULL
CS-GetCapabilityList-Response ::= NULL
CS-UnsolicitedStatus ::= NULL
AdditionalService-Error ::= NULL
AdditionalUnconfirmedService ::= NULL
EE-Additional-Detail ::= NULL

JOU-Additional-Detail::= NULL
 CS-GetNameList-Request::= NULL
 CS-GetNameList-Response::= NULL

7.3 Définition et utilisation des services spécifiques de Contrôle-Commande de procédé

7.3.1 Control Element (Elément de Contrôle-Commande)

L'Elément de Contrôle-Commande est un paramètre complexe utilisé dans plusieurs services pour décrire un élément unique d'un objet Contrôle-Commande d'Unité.

7.3.1.1 Structure

La structure du paramètre Elément de Contrôle-Commande est définie dans le tableau 8.

Tableau 8 – Paramètre d'Elément de Contrôle-Commande

Nom du paramètre	Rép	Cnf
Début de Définition de Domaine	S	S(=)
Nom de Domaine	M	M(=)
Liste des Capacités	M	M(=)
Partageable	M	M(=)
Données de Chargement	M	M(=)
Continuer Définition de Domaine	S	S(=)
Nom de Domaine	M	M(=)
Données de Chargement	M	M(=)
Fin de Définition de Domaine	S	S(=)
Nom de Domaine	M	M(=)
Définition d'Invocation de Programme	S	S(=)
Nom d'Invocation de Programme	M	M(=)
Liste de Domaines	M	M(=)
Réutilisable	M	M(=)
Surveillance	U	U(=)
Type de Surveillance	C	C(=)
Etat d'Invocation de Programme	C	C(=)

7.3.1.1.1 Début de Définition de Domaine

La sélection de ce paramètre doit indiquer qu'un nouveau Domaine va être défini. S'il est sélectionné, les paramètres suivants doivent apparaître.

7.3.1.1.1.1 Nom de Domaine

Ce paramètre, du type identificateur, doit identifier le Domaine à traiter.

7.3.1.1.1.2 Liste des Capacités

Ce paramètre, du type liste de chaîne de caractères, doit identifier les capacités associées à ce Domaine.

JOU-Additional-Detail ::= NULL
 CS-GetNameList-Request ::= NULL
 CS-GetNameList-Response ::= NULL

7.3 Definition and use of process control specific services

7.3.1 Control Element

The Control Element is a complex parameter used in several services to describe a single element of a Unit Control object.

7.3.1.1 Structure

The structure of the Control Element parameter is shown in table 8.

Table 8 – Control Element Parameter

Parameter name	Req	Cnf
Begin Domain Definition	S	S (=)
Domain Name	M	M (=)
List of Capabilities	M	M (=)
Shareable	M	M (=)
Load Data	M	M (=)
Continue Domain Definition	S	S (=)
Domain Name	M	M (=)
Load Data	M	M (=)
End Domain Definition	S	S (=)
Domain Name	M	M (=)
Program Invocation Definition	S	S (=)
Program Invocation Name	M	M (=)
List of Domains	M	M (=)
Reusable	M	M (=)
Monitor	U	U (=)
Monitor Type	C	C (=)
Program Invocation State	C	C (=)

7.3.1.1.1 Begin Domain Definition

Selection of this parameter shall indicate that a new Domain is about to be defined. If selected, the following parameters shall appear.

7.3.1.1.1.1 Domain Name

This parameter, of type identifier, shall identify the Domain to be processed.

7.3.1.1.1.2 List of Capabilities

This parameter, of type list of character string, shall identify the capabilities associated with this Domain.

7.3.1.1.1.3 Partageable

Ce paramètre, du type booléen, doit indiquer, s'il est VRAI, que le domaine est partageable, comme défini dans 10.8.1.1.3 de l'ISO/IEC 9506-1.

7.3.1.1.1.4 Données à charger

Ce paramètre doit être le contenu initial (ou total) du Domaine.

7.3.1.1.2 Continuer Définition de Domaine

La sélection de ce paramètre doit indiquer qu'un Domaine identifié dans un Élément de Contrôle-Commande précédent va avoir d'autres données qui lui seront associées. S'il est sélectionné, les paramètres suivants doivent apparaître.

7.3.1.1.2.1 Nom du Domaine

Ce paramètre, du type identificateur, doit identifier le Domaine à traiter.

7.3.1.1.2.2 Données à charger

Ce paramètre doit être le contenu (partiel) du Domaine.

7.3.1.1.3 Fin de Définition de Domaine

La sélection de ce paramètre doit indiquer qu'un Domaine identifié dans un Élément de Contrôle-Commande précédent est à présent complet. S'il est sélectionné, le paramètre suivant doit apparaître.

7.3.1.1.3.1 Nom du Domaine

Ce paramètre, du type identificateur, doit identifier le Domaine à traiter.

7.3.1.1.4 Définition d'Invocation de Programme

La sélection de ce paramètre doit indiquer qu'une Définition d'Invocation de Programme suit. S'il est sélectionné, les paramètres suivants doivent apparaître.

7.3.1.1.4.1 Nom d'Invocation de Programme

Ce paramètre, du type identificateur, doit indiquer l'Invocation de Programme à définir.

7.3.1.1.4.2 Liste de Domaines

Ce paramètre, du type liste d'identificateurs, doit indiquer les Domaines liés à l'Invocation de Programme.

7.3.1.1.4.3 Réutilisable

Ce paramètre, du type booléen, doit être utilisé comme prescrit dans 11.2.1.1.3 de l'ISO/IEC 9506-1.

7.3.1.1.1.3 *Shareable*

This parameter, of type boolean, shall indicate if TRUE that the Domain is shareable, as defined in 10.8.1.1.3 of ISO/IEC 9506-1.

7.3.1.1.1.4 *Load Data*

This parameter shall be the initial (or the total) content of the Domain.

7.3.1.1.2 *Continue Domain Definition*

Selection of this parameter shall indicate that a Domain identified in a previous Control Element is about to have more data associated with it. If selected, the following parameters shall appear.

7.3.1.1.2.1 *Domain Name*

This parameter, of type identifier, shall identify the Domain to be processed.

7.3.1.1.2.2 *Load Data*

This parameter shall be the (partial) content of the Domain.

7.3.1.1.3 *End Domain Definition*

Selection of this parameter shall indicate that a Domain identified in a previous Control Element is now complete. If selected, the following parameter shall appear.

7.3.1.1.3.1 *Domain Name*

This parameter, of type identifier, shall identify the Domain to be processed.

7.3.1.1.4 *Program Invocation Definition*

Selection of this parameter shall indicate that a Program Invocation Definition follows. If selected, the following parameters shall appear.

7.3.1.1.4.1 *Program Invocation Name*

This parameter, of type identifier, shall indicate the Program Invocation to be defined.

7.3.1.1.4.2 *List of Domains*

This parameter, of type list of Identifier, shall indicate the Domains that are bound to the Program Invocation.

7.3.1.1.4.3 *Reusable*

This parameter, of type boolean, shall be utilized as specified in 11.2.1.1.3 of ISO/IEC 9506-1.

7.3.1.1.4.4 Surveillance

Ce paramètre, du type booléen, doit être utilisé comme prescrit dans 11.2.1.1.4 de l'ISO/IEC 9506-1.

7.3.1.1.4.4.1 Type de Surveillance

Ce paramètre, du type booléen, doit être présent si, et uniquement si, la valeur du paramètre Surveillance est VRAIE. L'utilisation de ce paramètre est définie dans 11.2.1.1.4.1 de l'ISO/IEC 9506-1.

7.3.1.1.4.5 Etat d'une Invocation de Programme

Ce paramètre, du type entier, doit indiquer, s'il est présent, l'état de l'Invocation du Programme. Lorsqu'il est utilisé avec le service UnitControlLoad (Chargement de Contrôle-Commande d'Unité), il doit indiquer l'état dans lequel l'Invocation de Programme doit être mise. Lorsqu'il est utilisé avec le service UnitControlUpload (Télesauvegarde de Contrôle-Commande d'Unité), il doit indiquer l'état en cours de l'Invocation de Programme.

7.3.1.2 Protocole

```
ControlElement ::= CHOICE {
    beginDomainDef      [0] SEQUENCE {
        domainName      [1] IMPLICIT Identifier,
        capabilities     [2] IMPLICIT SEQUENCE OF VisibleString,
        shareable        [3] IMPLICIT BOOLEAN,
        loadData         [4] LoadData OPTIONAL,
    },
    continueDomainDef   [1] SEQUENCE {
        domainName      [1] IMPLICIT Identifier,
        loadData         [3] LoadData
    },
    endDomainDef        [1] IMPLICIT Identifier,
    piDefinition        [2] IMPLICIT SEQUENCE {
        piName          [0] IMPLICIT Identifier,
        listOfDomains   [1] IMPLICIT SEQUENCE OF Identifier,
        reusable        [2] IMPLICIT BOOLEAN DEFAULT TRUE,
        monitorType     [3] IMPLICIT BOOLEAN OPTIONAL,
        piState         [4] IMPLICIT ProgramInvocationState OPTIONAL
    }
}

LoadData ::= CHOICE {
    non-coded           [0] IMPLICIT OCTET STRING,
    coded               EXTERNAL
}
```

7.3.1.1.4.4 Monitor

This parameter, of type boolean, shall be utilized as specified in 11.2.1.1.4 of ISO/IEC 9506-1.

7.3.1.1.4.4.1 Monitor Type

This parameter, of type boolean, shall be present if, and only if, the value of the Monitor parameter is TRUE. The use of this parameter is defined in 11.2.1.1.4.1 of ISO/IEC 9506-1.

7.3.1.1.4.5 Program Invocation State

This parameter, of type integer, shall indicate, if present, the state of the Program Invocation. When used with UnitControlLoad service, it shall indicate the state in which the Program Invocation shall be placed. When used with the UnitControlUpload service, it shall indicate the actual state of the Program Invocation.

7.3.1.2 Protocol

```
ControlElement ::= CHOICE {
  beginDomainDef      [0] SEQUENCE {
    domainName          [1] IMPLICIT Identifier,
    capabilities         [2] IMPLICIT SEQUENCE OF VisibleString,
    shareable           [3] IMPLICIT BOOLEAN,
    loadData             [4] LoadData OPTIONAL
  },
  continueDomainDef   [1] SEQUENCE {
    domainName          [1] IMPLICIT Identifier,
    loadData             [3] LoadData
  },
  endDomainDef         [2] IMPLICIT Identifier,
  piDefinition         [3] IMPLICIT SEQUENCE {
    piName              [0] IMPLICIT Identifier,
    listOfDomains       [1] IMPLICIT SEQUENCE OF Identifier,
    reusable            [2] IMPLICIT BOOLEAN DEFAULT TRUE,
    monitorType         [3] IMPLICIT BOOLEAN OPTIONAL,
    piState              [4] IMPLICIT ProgramInvocationState OPTIONAL
  }
}

LoadData ::= CHOICE {
  non-coded           [0] IMPLICIT OCTET STRING,
  coded               EXTERNAL
}
```

7.3.1.2.1 Surveillance

La syntaxe abstraite du paramètre Surveillance du paramètre élément de Contrôle-Commande doit être déduite de la présence ou de l'absence du champ monitorType (Type de surveillance). Si le champ monitorType est présent, il doit indiquer que le paramètre Surveillance est VRAI. La valeur du champ MonitorType doit indiquer la valeur du paramètre Type de Moniteur de la demande de service, telle que prescrite dans 11.2.1.1.4 de l'ISO/IEC 9506-1.

7.3.2 Service InitiateUnitControlLoad

(Déclencher Chargement de Contrôle-Commande d'Unité)

Il est admis que le service InitiateUnitControlLoad soit utilisé par un client pour demander à un VMD de créer un objet Contrôle-Commande d'Unité et du préparer au chargement. La procédure de chargement utilise deux services MMS confirmés tels que ce soit le serveur MMS qui demande les éléments constituant le Contrôle-Commande d'Unité. Le tableau 9 définit la séquence des primitives de service.

Tableau 9 – Interaction des primitives Contrôle-Commande d'Unité

Client MMS	Serveur MMS
InitiateUnitControlLoad.req-->	
	<-- UnitControlLoadSegment.rep
UnitControlLoadSegment.rep -->	
...	...
	<-- UnitControlLoadSegment.req
UnitControlLoadSegment.rep -->	
	<--InitiateUnitControlLoad.rep

7.3.2.1 Structure

La structure des primitives de service qui constituent le service InitiateUnitControlLoad est définie dans le tableau 10.

Tableau 10 – Service InitiateUnitControlLoad

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle Commande d'Unité	M	M(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Erreur de Déclenchement de Contrôle			M	M(=)	
Nom du Domaine			S	S(=)	
Nom d'Invocation de Programme			S	S(=)	

7.3.1.2.1 Monitor

The abstract syntax of the Monitor parameter of the Control Element parameter shall be inferred from the presence or absence of the monitorType field. If the monitorType field is present, it shall indicate that the Monitor parameter is TRUE. The value of the monitorType field shall indicate the value of the Monitor Type parameter of the service request as specified in 11.2.1.1.4 of ISO/IEC 9506-1.

7.3.2 InitiateUnitControlLoad service

The InitiateUnitControlLoad service may be used by a client to request a VMD to create a Unit Control object and prepare it for loading. The loading process uses two confirmed MMS services such that the MMS server requests elements of the contents of the Unit Control object. Table 9 shows the sequence of the service primitives.

Table 9 – Interaction of Unit Control primitives

MMS client	MMS server
InitiateUnitControlLoad.req-->	
	<-- UnitControlLoadSegment.rep
UnitControlLoadSegment.rsp -->	
...	...
	<-- UnitControlLoadSegment.req
UnitControlLoadSegment.rsp -->	
	<--InitiateUnitControlLoad.rsp

7.3.2.1 Structure

The structure of the component service primitives of the InitiateUnitControlLoad service is shown in table 10.

Table 10 – InitiateUnitControlLoad service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Initiate Unit Control Error			M	M(=)	
Domain Name			S	S(=)	
Program Invocation Name			S	S(=)	

7.3.2.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service **InitiateUnitControlLoad**.

7.3.2.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, doit spécifier le nom du Contrôle-Commande d'Unité.

7.3.2.1.2 Résultat(+)

Le paramètre **Résultat(+)** doit indiquer que la demande de service a réussi.

7.3.2.1.3 Résultat(-)

Le paramètre **Résultat(-)** doit indiquer que la demande de service a échoué. Le paramètre **Type d'Erreur**, qui est décrit plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.2.1.3.1 Nom du Domaine

Ce paramètre doit indiquer le **Domaine** en cours de création lorsque l'erreur a été détectée. L'indication doit comporter soit ce paramètre, soit le paramètre **Nom d'Invocation de Programme**.

7.3.2.1.3.2 Nom d'Invocation de Programme

Ce paramètre doit indiquer l'**Invocation de Programme** en cours de création lorsque l'erreur a été détectée. L'indication doit comporter ce paramètre ou le paramètre **Nom de Domaine**.

7.3.2.2 Procédure de service

Si un objet **Contrôle-Commande d'Unité** de nom spécifié existe déjà, le serveur MMS doit renvoyer un **Résultat(-)**. Autrement, le serveur MMS doit ensuite émettre, le cas échéant, une ou plusieurs demandes **UnitControlLoadSegment** (**Segment de Chargement de Contrôle-Commande d'Unité**) jusqu'à ce qu'il reçoive une réponse à laquelle le paramètre **More Follows** (à Suivre) est mis sur FAUX. Il doit exécuter la procédure de service prescrite pour ce service. Si, pendant le traitement des informations contenues dans la réponse à la demande de service **UnitControlLoadSegment**, il détecte une erreur, soit lors de la création d'un **Domaine** ou lors de la création d'une **Invocation de Programme**, il doit arrêter la procédure de chargement et renvoyer un **Résultat(-)** indiquant le **Domaine** ou l'**Invocation de Programme** dont le chargement était en cours lorsque l'erreur a eu lieu.

Si aucune erreur n'a lieu lors du traitement des services **UnitControlLoadSegment**, il doit créer un objet **Contrôle-Commande d'Unité** portant le nom spécifié dans l'argument de demande et initialiser ses attributs pour référencer les **Domaines** et **Invocations de Programme** créés dans la procédure de chargement. Il doit ensuite renvoyer un **Résultat(+)**.

7.3.2.1.1 *Argument*

This parameter shall convey the parameters of the InitiateUnitControlLoad service request.

7.3.2.1.1.1 *Unit Control Name*

This parameter, of type identifier, shall specify the name of the Unit Control object.

7.3.2.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service request has succeeded.

7.3.2.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.2.1.3.1 *Domain Name*

This parameter shall indicate the Domain that was being created when the error was detected. Either this parameter or the Program Invocation Name parameter shall be selected.

7.3.2.1.3.2 *Program Invocation Name*

This parameter shall indicate the Program Invocation that was being created when the error was detected. Either this parameter or the Domain Name parameter shall be selected.

7.3.2.2 *Service procedure*

If a Unit Control object of the specified name already exists, the MMS server shall return a Result(-). Otherwise, the MMS server shall then issue one or more UnitControlLoadSegment requests, as appropriate, until it receives a response in which the More Follows parameter is set FALSE. It shall perform the service procedure prescribed for that service. If, during the processing of the information contained in the response to the UnitControlLoadSegment service request, it detects an error either in creating a Domain or a Program Invocation, it shall halt the loading process and return a Result(-) indicating the Domain or Program Invocation for which loading was in progress when the error occurred.

If no error occurs in the processing of the UnitControlLoadSegment services, it shall create a Unit Control object with the name specified in the request argument, and initialize its attributes to reference the Domains and Program Invocations created in the loading process. It shall then return a Result(+).

7.3.2.3 Protocole *InitiateUnitControlLoad*

La syntaxe abstraite de la sélection *initiateUCLoad* de l'*AdditionalService-Request* (Demande de Service Supplémentaire) et de l'*AdditionalService-Response* (Réponse de Service Supplémentaire) est spécifiée par *InitiateUnitControlLoad-Request* (Demande-Déclencher Chargement Contrôle-Commande d'Unité) et par *InitiateUnitControlLoad-Response* (Réponse-Déclencher Chargement Contrôle-Commande d'Unité), respectivement. La syntaxe abstraite de la sélection *initiateUCLoad* du *AdditionalService-Error* (Erreur-Service Supplémentaire) est spécifiée par *InitiateUnitControl-Error* (Erreur-Déclencher Contrôle-Commande d'Unité). Ces types de paramètres sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels le présent article ne fournit pas de dérivations explicites.

```
InitiateUnitControlLoad-Request ::= Identifier -- Unit Control Name
InitiateUnitControlLoad-Response ::= NULL
InitiateUnitControl-Error ::= CHOICE {
    domain                [0] IMPLICIT Identifier,
    programInvocation     [1] IMPLICIT Identifier
}
```

7.3.2.3.1 *InitiateUnitControlLoad-Request*

La syntaxe abstraite de la sélection *initiateUCLoad* du type *AdditionalService-Request* doit être *InitiateUnitControlLoad-Request*.

7.3.2.3.2 *InitiateUnitControlLoad-Response*

La syntaxe abstraite de la sélection *initiateUCLoad* du type *AdditionalService-Response* doit être *InitiateUnitControlLoad-Response*.

7.3.2.3.3 *InitiateUnitControlLoad-Error*

La syntaxe abstraite de la sélection *initiateUCLoad* du type *AdditionalService-Error* doit être *InitiateUnitControlLoad-Error*.

7.3.3 *Service UnitControlLoadSegment* (Segment de Chargement de Contrôle-Commande d'Unité)

Le service *UnitControlLoadSegment* est utilisé par un serveur MMS pour obtenir des éléments de données de chargement du client MMS.

7.3.3.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 11.

7.3.2.3 *InitiateUnitControlLoad protocol*

The abstract syntax of the initiateUCLoad choice of the AdditionalService-Request and the AdditionalService-Response is specified by the InitiateUnitControlLoad-Request and the InitiateUnitControlLoad-Response, respectively. The abstract syntax of the initiateUCLoad choice of the AdditionalService-Error is specified by the InitiateUnitControl-Error. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

InitiateUnitControlLoad-Request ::= Identifier -- Unit Control Name

InitiateUnitControlLoad-Response ::= NULL

InitiateUnitControl-Error ::= CHOICE {
 domain [0] IMPLICIT Identifier,
 programInvocation [1] IMPLICIT Identifier
 }

7.3.2.3.1 *InitiateUnitControlLoad-Request*

The abstract syntax of the initiateUCLoad choice of the AdditionalService-Request type shall be the InitiateUnitControlLoad-Request.

7.3.2.3.2 *InitiateUnitControlLoad-Response*

The abstract syntax of the initiateUCLoad choice of the AdditionalService-Response type shall be the InitiateUnitControlLoad-Response.

7.3.2.3.3 *InitiateUnitControlLoad-Error*

The abstract syntax of the initiateUCLoad choice of the AdditionalService-Error type shall be the InitiateUnitControlLoad-Error.

7.3.3 *UnitControlLoadSegment service*

The UnitControlLoadSegment service is used by an MMS server to obtain load data elements from the MMS client.

7.3.3.1 *Structure*

The structure of the component service primitives is shown in table 11.

Tableau 11 – Service UnitControlLoadSegment

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M (=)			
Nom de Contrôle-Commande d'Unité	M	M (=)			
Résultat (+)			S	S (=)	
Liste d'Éléments de Contrôle-Commande			M	M (=)	
A Suivre			M	M (=)	
Résultat (-)			S	S (=)	
Type d'Erreur			M	M (=)	

7.3.3.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service UnitControlLoadSegment.

7.3.3.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, doit identifier l'objet Contrôle-Commande d'Unité dans le VMD qui est à charger.

7.3.3.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi. Dans ce cas, les paramètres suivants doivent apparaître.

7.3.3.1.2.1 Liste d'Éléments de Contrôle-Commande

Ce paramètre doit contenir les informations nécessaires à la construction des Domaines et Invocations de Programme qui constituent l'objet Contrôle-Commande d'Unité. La présence du paramètre Etat d'Invocation de Programme de chaque Élément de Contrôle-Commande doit être une option utilisateur.

7.3.3.1.2.2 A Suivre

Ce paramètre booléen doit indiquer si d'autres demandes de service UnitControlSegment (Segment de Contrôle-Commande d'Unités) sont (VRAI) ou ne sont pas (FAUX) nécessaires pour compléter la construction d'objet Contrôle-Commande d'Unité.

7.3.3.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est décrit de manière plus détaillée dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

Table 11 – UnitControlLoadSegment service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Result(+)			S	S(=)	
List of Control Elements			M	M(=)	
More Follows			M	M(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.3.1.1 Argument

This parameter shall convey the parameters of the UnitControlLoadSegment service request.

7.3.3.1.1.1 Unit Control Name

This parameter, of type identifier, shall identify the Unit Control object in the VMD that is to be loaded.

7.3.3.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded. If success is indicated, the following parameters shall appear.

7.3.3.1.2.1 List of Control Elements

This parameter shall contain the information necessary to construct the constituent Domains and Program Invocations of the Unit Control object. The presence of the Program Invocation State parameter of each Control Element shall be a user option.

7.3.3.1.2.2 More Follows

This boolean parameter shall indicate whether (TRUE) or not (FALSE) more UnitControl-Segment service requests are needed to complete the construction of the Unit Control object.

7.3.3.1.3 Result(-)

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.3.2 Procédure de Service

Le serveur MMS doit émettre une demande UnitControlLoadSegment, spécifiant le nom d'objet Contrôle-Commande d'Unité à charger. Sur réception de la réponse, le serveur MMS doit traiter dans la réponse, chacun des éléments de Contrôle-Commande en séquence l'un après l'autre.

Pour chaque élément de la liste des éléments de Contrôle-Commande, le serveur MMS doit:

- a) Si l'Elément de Contrôle-Commande spécifie le début d'une Définition de Domaine, le serveur MMS doit vérifier qu'aucun Domaine de ce nom n'existe dans le VMD. Il doit ensuite créer le Domaine, en utilisant le paramètre Liste de Capacités prévu à cet effet et le mettre en état de chargement. Si le paramètre Données de Chargement est fourni, il doit alors commencer la procédure de CHARGEMENT, en utilisant les Données de Chargement.
- b) Si l'Elément de Contrôle-Commande spécifie la poursuite d'une Définition du Domaine, le serveur MMS doit vérifier que le Domaine existe, et qu'il est dans l'état de CHARGEMENT. Il doit ensuite poursuivre la procédure de chargement, en utilisant les paramètres Données de Chargement prévus à cet effet.
- c) Si l'Elément de Contrôle-Commande indique la fin d'une Définition de Domaine, le serveur MMS doit vérifier que le Domaine existe et qu'il est dans l'état de CHARGEMENT. Il doit ensuite mettre le Domaine en état PRET.
- d) Si l'Elément de Contrôle-Commande indique une définition d'Invocation de Programme, le serveur MMS doit vérifier que tous les Domaines existent dans le paramètre Liste de Domaines et qu'ils sont à l'état PRET ou EN-COURS-D'UTILISATION et que leur attribut partageable est VRAI. Il doit ensuite créer l'Invocation de Programme en la liant aux Domaines indiqués. Il doit placer chacun des Domaines en l'état EN-COURS-D'UTILISATION. Si le paramètre Etat d'Invocation de Programme est présent, le serveur MMS doit placer l'Invocation de Programme en l'état indiqué par ce paramètre; autrement, il doit mettre l'Invocation de Programme en état REPOS.

NOTE - Si l'association est perdue au cours d'une séquence de service UnitControlLoadSegment, de sorte qu'un Domaine se trouve dans un état intermédiaire, les dispositions du 10.1.4.1 de l'ISO/IEC 9506-1 s'appliquent. De même, on notera les restrictions du 8.3.2 de l'ISO/IEC 9506-2 sur l'utilisation du service Conclude (Conclude).

7.3.3.3 Protocole UnitControlLoadSegment

La syntaxe abstraite de la sélection uLoad de AdditionalService-Request et de AdditionalService-Response est définie par UnitControlLoadSegment-Request et UnitControlLoadSegment-Response, respectivement. Ces types sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

UnitControlLoadSegment-Request ::= Identifier -- Unit Control Name

```
UnitControlLoadSegment-Response ::= SEQUENCE {
    controleElements[0] IMPLICIT SEQUENCE OF ControleElement,
    moreFollows      [1] IMPLICIT BOOLEAN DEFAULT TRUE
}
```

7.3.3.2 Service procedure

The MMS server shall issue a UnitControlLoadSegment request specifying the name of the Unit Control object to be loaded. Upon receipt of the response, the MMS server shall process each of the control elements in the response in sequence.

For each item in the List of Control Elements the MMS server shall:

- a) If the Control Element specifies the beginning of a Domain Definition, the MMS server shall verify that no Domain of that name exists in the VMD. It shall then create the Domain, using the List of Capabilities parameter provided, and place it in the loading state. If the Load Data parameter is provided, it shall then begin the LOADING process, using the load data.
- b) If the Control Element specifies the continuation of a Domain Definition, the MMS server shall verify that the Domain exists and is in the LOADING state. It shall then continue the loading process, using the Load Data parameter provided.
- c) If the Control Element specifies the end of a Domain Definition, the MMS server shall verify that the Domain exists and that it is in the LOADING state. It shall then place the Domain in the READY state.
- d) If the Control Element specifies a Program Invocation definition, the MMS server shall verify that all the Domains in the List of Domains parameter exist and that they are in the READY state or that they are in the IN-USE state and their shareable attribute is TRUE. It shall then create the named Program Invocation, linking it to the indicated Domains. It shall place each of the Domains in the IN-USE state. If the Program Invocation State parameter is present, the MMS server shall place the Program Invocation in the state indicated by this parameter; otherwise, it shall place the Program Invocation in IDLE state.

NOTE – If the association is lost during the course of a sequence of UnitControlLoadSegment services such that a Domain is in an intermediate state, the provisions of 10.1.4.1 of ISO/IEC 9506-1 apply. Also note the restrictions that 8.3.2 of ISO/IEC 9506-2 places on the use of the Conclude service.

7.3.3.3 UnitControlLoadSegment protocol

The abstract syntax of the uCLoad choice of the AdditionalService-Request and the AdditionalService-Response is specified by the UnitControlLoadSegment-Request and the UnitControlLoadSegment-Response, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

UnitControlLoadSegment-Request ::= Identifier -- Unit Control Name

UnitControlLoadSegment-Response ::= SEQUENCE {

controlElements [0] IMPLICIT SEQUENCE OF ControlElement,
moreFollows [1] IMPLICIT BOOLEAN DEFAULT TRUE

}

7.3.3.3.1 *UnitControlLoadSegment-Request* (Demande-Segment de Chargement de Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection uCLoad du type Additional Service-Request doit être UnitControlLoadSegment-Request.

7.3.3.3.2 *UnitControlLoadSegment-Response* (Réponse-Segment de Chargement de Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection uCLoad du type Additional Service-Response doit être UnitControlLoadSegment-Response.

7.3.4 *Service UnitControlUpload* (Télésauvegarde de Contrôle-Commande d'Unité)

Le service UnitControlUpload est utilisé par un client MMS pour obtenir des éléments de données de chargement du serveur MMS. Il est admis que ce service soit invoqué plusieurs fois pour obtenir une télésauvegarde complète de l'objet de Contrôle-Commande d'Unité.

7.3.4.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 12.

Tableau 12 – Service UnitControlUpload

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle-Commande d'Unité	M	M(=)			
A la suite de	U	U(=)			
Nom du Domaine	S	S(=)			
ID de Télésauvegarde	S	S(=)			
Invocation de Programme	S	S(=)			
Résultat(+)			S	S(=)	
Liste d'Éléments de Contrôle-Commande			M	M(=)	
Élément suivant			C	C(=)	
Nom de Domaine			S	S(=)	
ID de Télésauvegarde			S	S(=)	
Invocation de Programme			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.4.1.1 *Argument*

Ce paramètre doit véhiculer les paramètres de la demande de service UnitControlUpload.

7.3.3.3.1 UnitControlLoadSegment-Request

The abstract syntax of the uCLoad choice of the AdditionalService-Request type shall be the UnitControlLoadSegment-Request.

7.3.3.3.2 UnitControlLoadSegment-Response

The abstract syntax of the uCLoad choice of the AdditionalService-Response type shall be the UnitControlLoadSegment-Response.

7.3.4 UnitControlUpload service

The UnitControlUpload service is used by an MMS client to obtain load data elements from the MMS server. This service may have to be invoked several times to obtain a complete upload of the Unit Control object.

7.3.4.1 Structure

The structure of the component service primitives is shown in table 12.

Table 12 – UnitControlUpload service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Continue After	U	U(=)			
Domain Name	S	S(=)			
Upload ID	S	S(=)			
Program Invocation	S	S(=)			
Result(+)			S	S(=)	
List of Control Elements			M	M(=)	
Next Element			C	C(=)	
Domain Name			S	S(=)	
Upload ID			S	S(=)	
Program Invocation			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.4.1.1 Argument

This parameter shall convey the parameters of the UnitControlUpload service request.

7.3.4.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type Identificateur, doit identifier l'objet Contrôle-Commande d'Unité dans le VMD à télésauvegarder.

7.3.4.1.1.2 A la Suite de

Ce paramètre facultatif doit indiquer l'endroit où doit commencer l'Elément de Contrôle-Commande suivant dans la liste des éléments constitutifs d'objet Contrôle-Commande d'Unité. Si ce paramètre est absent, la télésauvegarde doit commencer au début du Contrôle-Commande d'Unité. Si ce paramètre est présent, l'un des paramètres suivants doit être sélectionné.

7.3.4.1.1.2.1 Nom du Domaine

Ce paramètre, du type identificateur, doit indiquer le Domaine suivant à télésauvegarder.

7.3.4.1.1.2.2 ID de Télésauvegarde

Ce paramètre, du type entier, doit indiquer la machine d'état de télésauvegarde actuellement ouverte pour la poursuite de la télésauvegarde de quelque Domaine.

7.3.4.1.1.2.3 Invocation de Programme

Ce paramètre, du type identificateur, doit indiquer l'Invocation de Programme suivante dont la définition est à sauvegarder.

7.3.4.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi. Dans ce cas, les paramètres suivants doivent apparaître.

7.3.4.1.2.1 Liste d'Eléments de Contrôle-Commande

Ce paramètre doit contenir les informations nécessaires pour construire les Domaines et Invocations de Programme qui constituent l'objet Contrôle-Commande d'Unité. Le paramètre d'état Invocation de Programme doit être inclus dans chaque définition d'Invocation de Programme dans un Elément de Contrôle-Commande et sa valeur doit correspondre à l'état réel de l'Invocation de Programme.

7.3.4.1.2.2 Elément suivant

Ce paramètre facultatif doit indiquer, s'il est présent, le premier élément non transmis dans la liste d'Eléments de Contrôle-Commande, qui devrait être l'élément suivant à transmettre si une autre demande UnitControlUpload est reçue. L'absence de ce paramètre doit indiquer que le téléchargement de l'objet Contrôle-Commande d'Unité est terminé avec cette PDU. Si ce paramètre est présent, l'un des paramètres suivants doit être sélectionné.

7.3.4.1.2.2.1 Nom de Domaine

Ce paramètre, du type identificateur, doit indiquer le Domaine suivant à télésauvegarder.

7.3.4.1.1.1 *Unit Control Name*

This parameter, of type identifier, shall identify the Unit Control object in the VMD that is to be uploaded.

7.3.4.1.1.2 *Continue After*

This optional parameter shall indicate where in the list of constituents of the Unit Control object to begin the next Control Element. If this parameter is not present, the upload shall begin at the beginning of the Unit Control object. If this parameter is present, one of the following parameters shall be selected.

7.3.4.1.1.2.1 *Domain Name*

This parameter, of type identifier, shall indicate the next Domain that is to be uploaded.

7.3.4.1.1.2.2 *Upload ID*

This parameter, of type integer, shall indicate the upload state machine currently open for some Domain upload that is to be continued.

7.3.4.1.1.2.3 *Program Invocation*

This parameter, of type identifier, shall indicate the next Program Invocation whose definition is to be uploaded.

7.3.4.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service has succeeded. If success is indicated, the following parameters shall appear.

7.3.4.1.2.1 *List of Control Elements*

This parameter shall contain the information necessary to construct the constituent Domains and Program Invocations of the Unit Control object. The Program Invocation state parameter shall be included in each Program Invocation definition within a Control Element, and its value shall be set corresponding to the actual state of the Program Invocation.

7.3.4.1.2.2 *Next Element*

This optional parameter shall indicate, if present, the first element not transmitted in this list of Control Elements that should be the next element to be transmitted if another UnitControlUpload request is received. If this parameter is absent, this shall indicate that uploading of the Unit Control object is complete with this PDU. If this parameter is present, one of the following parameters shall be selected.

7.3.4.1.2.2.1 *Domain Name*

This parameter, of type identifier, shall indicate the next Domain that is to be uploaded.

7.3.4.1.2.2.2. *ID de Télésauvegarde*

Ce paramètre, du type entier, doit indiquer la machine d'état de télésauvegarde actuellement ouverte pour la poursuite du téléchargement de quelque Domaine.

7.3.4.1.2.2.3 *Invocation de Programme*

Ce paramètre, du type identificateur, doit indiquer l'Invocation de Programme suivante dont la définition est à télésauvegarder.

7.3.4.1.3 *Résultat(-)*

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est décrit plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.4.2 *Procédure de service*

Pour répondre à la demande de service UnitControlUpload, le serveur MMS doit garder les éléments qui constituent l'objet d'Unité de Contrôle-Commande dans une liste ordonnée. Un classement des Noms de Domaines, fondés sur la séquence d'interclassement de l'ISO 646, suivi d'un classement similaire des Invocations de Programme est suggéré mais non prescrit. La présente partie de l'ISO/IEC 9506 exige uniquement que l'algorithme de classement utilisé soit univoque et soit tel que toute invocation de Programme apparaisse plus tard dans le même ordre que celui de tout Domaine dont elle dépend.

Il est admis que le client MMS émette une demande UnitControlUpload indiquant une position dans ce classement, en identifiant le Domaine suivant à télésauvegarder, le Domaine dont la télésauvegarde est partiellement terminée, ou la définition d'Invocation de Programme suivante à télésauvegarder. Si le client MMS ne précise pas l'un de ces éléments, la télésauvegarde commencera à partir du début du classement.

Le serveur MMS doit vérifier la cohérence de l'attribut Liste des références de Domaine et de l'attribut Liste de références d'Invocation de Programme d'objet Contrôle-Commande d'Unité. Si des objets ainsi référencés n'existent pas, le serveur MMS doit modifier les listes correspondantes.

Le serveur MMS doit fournir des définitions de chaque élément constitutif d'objet Contrôle-Commande d'Unité dans l'ordre, selon un algorithme de classement. Pour chaque Domaine d'objet Contrôle-Commande d'Unité, le serveur MMS doit créer une Machine d'Etat de Télésauvegarde (voir 10.1.4.2 de l'ISO/IEC 9506-1) et transmettre tout ou partie du contenu du Domaine. La nécessité de la segmentation ainsi que la taille des segments doivent être déterminées localement. Pour chaque Invocation de Programme dans le Contrôle-Commande d'Unité, le serveur MMS doit transmettre un enregistrement de définition d'Invocation de Programme.

Si, pour quelque raison, l'ensemble du contenu de l'objet Contrôle-Commande d'Unité ne peut être intégré dans une seule PDU, le serveur MMS doit indiquer, dans l'ordre, l'élément suivant qui n'a pas encore été transmis.

- a) Si l'élément suivant à transmettre est un Domaine, le serveur MMS doit indiquer le nom de ce Domaine.

7.3.4.1.2.2.2 *Upload ID*

This parameter, of type integer, shall indicate the upload state machine currently open for some Domain upload that is to be continued.

7.3.4.1.2.2.3 *Program Invocation*

This parameter, of type identifier, shall indicate the next Program Invocation whose definition is to be uploaded.

7.3.4.1.2.2.4 *Result(-)*

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.4.2 *Service procedure*

For the purposes of responding to the UnitControlUpload service request, the MMS server shall maintain the constituents of a Unit Control object in an ordered list. An ordering of Domain Names based on the collating sequence of ISO 646 followed by a similar ordering of Program Invocations is suggested but not required. This part of ISO/IEC 9506 requires only that the ordering algorithm used be unambiguous and be such that any Program Invocation appears later in the ordering than any Domains on which it depends.

The MMS client may issue a UnitControlUpload request indicating a position in this ordering by identifying the next Domain to be uploaded, the Domain whose upload is partially complete, or the next Program Invocation definition to be uploaded. If the MMS client does not specify any such element, the upload is to start from the beginning of the ordering.

The MMS server shall verify the consistency of the List of Domain references attribute and the List of Program Invocation references attribute of the Unit Control object. If any objects so referenced do not exist, the MMS server shall amend the respective list.

The MMS server shall provide definitions for each constituent element of the Unit Control object in order, determined by its ordering algorithm. For each Domain in the Unit Control object, the MMS server shall create an Upload State Machine (See 10.1.4.2 of ISO/IEC 9506-1) and transmit all or part of the Domain content. The determination of the necessity of segmentation and the size of the segments shall be a local matter. For each Program Invocation in the Unit Control object, the MMS server shall transmit a Program Invocation definition record.

If, for any reason, the entire content of the Unit Control object cannot be contained within a single PDU, the MMS server shall provide an indication of the next element in the order that has not yet been transmitted.

- a) If the next element to be transmitted is a Domain, the MMS server shall indicate the name of this Domain.

- b) Si une partie du contenu du Domaine a été transmise et qu'une autre partie de ce domaine reste encore à transmettre, le serveur MMS doit indiquer l'identité de la Machine d'Etat de Télésauvegarde en cours d'activité.
- c) Si l'élément suivant à transmettre est une Invocation de Programme, le serveur MMS doit indiquer le nom de cette Invocation de Programme.

Si la transmission en cours épuise l'objet Unité de Contrôle-Commande, ce qui signifie qu'elle transmet jusqu'au dernier élément sur la liste, le serveur MMS doit omettre le paramètre Elément Suivant.

7.3.4.3 Protocole UnitControlUpload

La syntaxe abstraite de la sélection UCUpload de AdditionalService-Request et de AdditionalService-Response est spécifiée par UnitControlUpload-Request et UnitControlUpload-Response, respectivement. Ces types sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
UnitControlUpload-Request ::= SEQUENCE {
    unitControlName[0] IMPLICIT Identifier, -- Unit Control Name
    continueAfter
        CHOICE {
            domain [1] IMPLICIT Identifier,
            ulsmID [2] IMPLICIT INTEGER,
            programInvocation [3] IMPLICIT Identifier
        } OPTIONAL
    }
UnitControlUpload-Response ::= SEQUENCE {
    controleElements [0] IMPLICIT SEQUENCE OF ControleElement,
    nextElement
        CHOICE {
            domain [1] IMPLICIT Identifier,
            ulsmID [2] IMPLICIT INTEGER,
            programInvocation [3] IMPLICIT Identifier
        } OPTIONAL
    }
```

7.3.4.3.1 UnitControlUpload-Request (Demande-Télésauvegarde Contrôle-Commande d'Unité)

La syntaxe abstraite de sélection uCUpload du type AdditionalService-Request doit être UnitControlUpload-Request.

7.3.4.3.2 UnitControlUpload-Response (Réponse-Télésauvegarde Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection uCUpload du type AdditionalService-Response doit être UnitControlUpload-Response.

b) If a Domain content has been partially transmitted and more content of that Domain remains to be transmitted, the MMS server shall indicate the identify of the Upload State Machine currently active.

c) If the next element to be transmitted is a Program Invocation, the MMS server shall indicate the name of this Program Invocation.

If the present transmission exhausts the Unit Control object, that is, it transmits the last element on the list, the MMS server shall omit the Next Element parameter.

7.3.4.3 UnitControlUpload protocol

The abstract syntax of the uCUpload choice of the AdditionalService-Request and the AdditionalService-Response is specified by the UnitControlUpload-Request and the UnitControlUpload-Response, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
UnitControlUpload-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier, -- Unit Control Name
    continueAfter        CHOICE {
        domain           [1] IMPLICIT Identifier,
        ulsmID           [2] IMPLICIT INTEGER,
        programInvocation [3] IMPLICIT Identifier
    } OPTIONAL
}

UnitControlUpload-Response ::= SEQUENCE {
    controlelements [0] IMPLICIT SEQUENCE OF ControlElement,
    nextElement     CHOICE {
        domain           [1] IMPLICIT Identifier,
        ulsmID           [2] IMPLICIT INTEGER,
        programInvocation [3] IMPLICIT Identifier
    } OPTIONAL
}
```

7.3.4.3.1 UnitControlUpload-Request

The abstract syntax of the uCUpload choice of the AdditionalService-Request type shall be the UnitControlUpload-Request.

7.3.4.3.2 UnitControlUpload-Response

The abstract syntax of the uCUpload choice of the AdditionalService-Response type shall be the UnitControlUpload-Response.

7.3.5 Service StartUnitControl (Démarrage Contrôle-Commande d'Unité)

Ce service permet au client MMS de mettre à l'état EN EXECUTION toutes les Invocations de Programme qui constituent l'objet Contrôle-Commande d'Unité.

7.3.5.1 Structure

La structure des primitives qui constituent le service est définie dans le tableau 13.

Tableau 13 – Service StartUnitControl

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle Commande d'Unité	M	M(=)			
Argument d'Exécution	U	U(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Erreur Lancement Contrôle-Commande d'Unité			C	C(=)	
Nom d'Invocation de Programme			M	M(=)	
Etat d'Invocation de Programme			M	M(=)	

7.3.5.1.1 Argument

Ce paramètre véhicule le paramètre du service StartUnitControl.

7.3.5.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type Identificateur, doit identifier l'objet Contrôle-Commande d'Unité dont les Invocations de Programme qui la constituent doivent être lancées.

7.3.5.1.1.2 Argument d'Exécution

Il est admis que ce paramètre soit utilisé pour passer une information aux Invocations de Programme à lancer.

7.3.5.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi.

7.3.5.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini de manière plus détaillée dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.5 StartUnitControl service

This service allows the MMS client to place all the constituent Program Invocations of a Unit Control object into the RUNNING state.

7.3.5.1 Structure

The structure of the component service primitives is shown in table 13.

Table 13 – StartUnitControl service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Execution Argument	U	U(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Start Unit Control Error			C	C(=)	
Program Invocation Name			M	M(=)	
Program Invocation State			M	M(=)	

7.3.5.1.1 Argument

This parameter conveys the parameters of the StartUnitControl service.

7.3.5.1.1.1 Unit Control Name

This parameter, of type identifier, shall identify the Unit Control object whose constituent Program Invocations are to be started.

7.3.5.1.1.2 Execution Argument

This parameter may be used to pass an information to the Program Invocations which are to be started.

7.3.5.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded.

7.3.5.1.3 Result(-)

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.5.1.3.1 *Erreur Lancement Contrôle-Commande d'Unité*

Ce paramètre doit être inclus dans le Résultat(-) si l'échec est dû à l'échec de la procédure de Lancement résultante sur une Invocation de Programme spécifique. Si ce paramètre est inclus, les champs suivants doivent également apparaître.

7.3.5.1.3.1.1 *Nom d'Invocation de Programme*

Ce paramètre doit indiquer le nom de l'Invocation de Programme dont le service de Lancement a échoué.

7.3.5.1.3.2 *Etat d'Invocation de Programme*

Ce paramètre doit indiquer l'état résultant de l'Invocation de Programme dont le service de Lancement a échoué. Après un échec de service de Lancement, l'Invocation de Programme doit si possible retourner à son état précédent, ou être mise à l'état NON EXECUTABLE.

7.3.5.2 *Procédure de Service*

Le serveur MMS doit:

- a) Pour chaque entrée de l'attribut Liste des références d'Invocation de Programme d'objet Contrôle-Commande d'Unité, vérifier que l'Invocation de Programme existe. Si une Invocation de Programme n'existe pas, sa référence doit être retirée de l'attribut Liste des références d'Invocation de Programme.
- b) Pour chaque entrée de l'attribut Liste des références d'Invocation de Programme d'objet Contrôle-Commande d'Unité, placer l'Invocation de Programme en l'état EN EXECUTION. Ceci doit être effectué comme suit:
 - 1) Si l'Invocation de Programme est déjà en l'état EN EXECUTION, LANCEMENT ou REPRISE, ne rien faire.
 - 2) Si l'Invocation de Programme est en l'état REPOS ou REINITIALISATION, exécuter une procédure de service de LANCEMENT (voir 11.4.2 de l'ISO/IEC 9506-1).
 - 3) Si l'Invocation de Programme est en l'état ARRETE ou EN COURS D'ARRET, exécuter une procédure de Reprise (voir 11.6.2 de l'ISO/IEC 9506-1).
 - 4) Si l'Invocation de Programme est en l'état NON EXECUTABLE, renvoyer un Résultat(-) avec un paramètre d'Erreur de Lancement de Contrôle-Commande d'Unité indiquant l'Invocation de Programme qui a échoué et son état.
 - 5) Si une procédure de Lancement sur une Invocation de Programme qui la constitue échoue, envoyer un Résultat(-) avec le paramètre Erreur de Lancement de Contrôle-Commande d'Unité indiquant l'Invocation de Programme qui a échoué ainsi que son état.
- c) Autrement, renvoyer un Résultat(+).

7.3.5.3 *Protocole StartUnitControl*

La syntaxe abstraite de la sélection startUC de AdditionalService-Request et AdditionalService-Response est spécifiée par StartUnitControl-Request et StartUnitControl-Response, respectivement. Ces types sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les

7.3.5.1.3.1 *Start Unit Control Error*

This parameter shall be included in the Result(-) if the failure was due to the failure of the derived Start procedure on a specific Program Invocation. If this parameter is included, the following fields shall also appear.

7.3.5.1.3.1.1 *Program Invocation Name*

This parameter shall indicate the name of the Program Invocation whose Start service failed.

7.3.5.1.3.2 *Program Invocation State*

This parameter shall indicate the resulting state of the Program Invocation whose Start service has failed. Following an unsuccessful Start service, the Program Invocation shall be returned to its previous state if possible, or it shall be placed in the UNRUNNABLE state.

7.3.5.2 *Service procedure*

The MMS server shall:

- a) For each entry on the List of Program Invocation references attribute of the Unit Control object, verify that the Program Invocation exists. If a Program Invocation does not exist, remove its reference from the List of Program Invocation references attribute.
- b) For each entry on the List of Program Invocation references attribute of the Unit Control object, place the Program Invocation in the RUNNING state. This shall be done as follows:
 - 1) If the Program Invocation is already in the RUNNING, STARTING, or RESUMING state, do nothing.
 - 2) If the Program Invocation is in the IDLE or RESETING state, perform a Start service procedure (see 11.4.2 of ISO/IEC 9506-1).
 - 3) If the Program Invocation is in the STOPPED or STOPPING state, perform a Resume procedure (see 11.6.2 of ISO/IEC 9506-1).
 - 4) If the Program Invocation is in the UNRUNNABLE state, return a Result(-) with a Start Unit Control Error parameter indicating the failed Program Invocation and its state.
 - 5) If any Start procedure on a constituent Program Invocation fails, return a Result(-) with a Start Unit Control Error parameter indicating the failed Program Invocation and its state.
- c) Otherwise, return a Result(+).

7.3.5.3 *StartUnitControl protocol*

The abstract syntax of the startUC choice of the AdditionalService-Request and the AdditionalService-Response is specified by the StartUnitControl-Request and the StartUnitControl-Response respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of

paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```

StartUnitControl-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier, -- Unit Control Name
    executionArgument    CHOICE {
        simpleString      [1] IMPLICIT VisibleString,
        encodedString      EXTERNAL
    } OPTIONAL
}

StartUnitControl-Response ::= NULL

StartUnitControl-Error ::= SEQUENCE {
    programInvocationName [0] IMPLICIT Identifier,
    programInvocationState [1] IMPLICIT ProgramInvocationState
}
    
```

7.3.5.3.1 *StartUnitControl-Request (Demande-Lancement Contrôle-Commande d'Unité)*

La syntaxe abstraite de la sélection startUC du type AdditionalService-Request doit être StartUnitControl-Request.

7.3.5.3.2 *StartUnitControl-Response (Réponse-Lancement Contrôle-Commande d'Unité)*

La syntaxe abstraite de la sélection startUC du type AdditionalService-Response doit être StartUnitControl-Response.

7.3.5.3.3 *StartUnitControl-Error (Erreur-Lancement Contrôle-Commande d'Unité)*

La syntaxe abstraite de la sélection startUC du type AdditionalService-Error doit être StartUnitControl-Error, qui sera le sous-paramètre Nom d'Invocation de Programme et le sous-paramètre Etat d'Invocation de Programme, respectivement, du paramètre Résultat(-) de la primitive StartUnitControl.response et doit apparaître comme étant le sous-paramètre Nom d'Invocation de Programme et le sous-paramètre Etat d'Invocation de Programme, respectivement, de la primitive StartUnitControl.confirm, si elle est émise.

7.3.6 *Service StopUnitControl (Arrêt Contrôle-Commande d'Unité)*

Ce service permet au client MMS de mettre à l'état ARRETE toutes les Invocations de Programme qui constituent l'objet Contrôle-Commande d'Unité.

7.3.6.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 14.

all parameters for which explicit derivations are not provided in this clause.

```

StartUnitControl-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier, -- Unit Control Name
    executionArgument     CHOICE {
        simpleString      [1] IMPLICIT VisibleString,
        encodedString      EXTERNAL
    } OPTIONAL
}

StartUnitControl-Response ::= NULL

StartUnitControl-Error ::= SEQUENCE {
    programInvocationName [0] IMPLICIT Identifier,
    programInvocationState [1] IMPLICIT ProgramInvocationState
}

```

7.3.5.3.1 *StartUnitControl-Request*

The abstract syntax of the startUC choice of the AdditionalService-Request type shall be the StartUnitControl-Request.

7.3.5.3.2 *StartUnitControl-Response*

The abstract syntax of the startUC choice of the AdditionalService-Response type shall be the StartUnitControl-Response.

7.3.5.3.3 *StartUnitControl-Error*

The abstract syntax of the startUC choice of the AdditionalService-Error type shall be the StartUnitControl-Error, which shall be Program Invocation Name sub-parameter and the Program Invocation State sub-parameter, respectively, of the Result(-) parameter of the StartUnitControl.response primitive and shall appear as the Program Invocation Name sub-parameter and the Program Invocation State sub-parameter, respectively, of the StartUnitControl.confirm primitive, if issued.

7.3.6 *StopUnitControl service*

This service allows the MMS client to place all the constituent Program Invocations of a Unit Control object into the STOPPED state.

7.3.6.1 *Structure*

The structure of the component service primitives is shown in table 14.

Tableau 14 – Service StopUnitControl

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle-Commande d'Unité	M	M(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Erreur Arrêt Contrôle-Commande d'Unité			C	C(=)	
Nom d'Invocation de Programme			M	M(=)	
Etat d'Invocation de Programme			M	M(=)	

7.3.6.1.1 Argument

Ce paramètre doit véhiculer le paramètre du service StopUnitControl.

7.3.6.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, doit identifier l'objet Contrôle-Commande d'Unité dont les Invocations de Programme sont à arrêter.

7.3.6.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi.

7.3.6.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.6.1.3.1 StopUnitControl-Error (Erreur Arrêt Contrôle-Commande d'Unité)

Ce paramètre doit être inclus dans le Résultat(-) si l'échec est dû à un échec de la procédure d'Arrêt résultante sur une Invocation de Programme spécifique. Si ce paramètre est inclus, les champs suivants doivent également apparaître.

7.3.6.1.3.1.1 Nom d'Invocation de Programme

Ce paramètre doit indiquer le nom de l'Invocation de Programme dont le service Arrêt a échoué.

7.3.6.1.3.1.2 Etat d'Invocation de Programme

Ce paramètre doit indiquer l'état résultant de l'invocation de Programme dont le service Arrêt a échoué. Après échec d'un service Arrêt, l'Invocation de Programme doit revenir, si possible, à son état précédent ou doit être mise à l'état NON EXECUTABLE.

Table 14 – StopUnitControl service

Parameter name	Req	Ind	Resp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Stop Unit Control Error			C	C(=)	
Program Invocation Name			M	M(=)	
Program Invocation State			M	M(=)	

7.3.6.1.1 Argument

This parameter shall convey the parameter of the StopUnitControl service.

7.3.6.1.1.1 Unit Control Name

This parameter, of type identifier, shall identify the Unit Control object whose constituent Program Invocations are to be stopped.

7.3.6.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded.

7.3.6.1.3 Result(-)

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.6.1.3.1 StopUnitControl-Error

This parameter shall be included in the Result(-) if the failure was due to the failure of the derived Stop procedure on a specific Program Invocation. If this parameter is included, the following fields shall also appear.

7.3.6.1.3.1.1 Program Invocation Name

This parameter shall indicate the name of the Program Invocation whose Stop service failed.

7.3.6.1.3.1.2 Program Invocation State

This parameter shall indicate the resulting state of the Program invocation whose Stop service has failed. Following an unsuccessful Stop service, the Program Invocation shall be returned to its previous state if possible, or it shall be placed in the UNRUNNABLE state.

7.3.6.2 Procédure de service

Le serveur MMS doit:

- a) Pour chaque entrée de l'attribut Liste des références d'Invocation de Programme d'objet Contrôle-Commande d'Unité, vérifier que l'Invocation de Programme existe. Si cette condition n'est pas satisfaite, la référence à l'Invocation de Programme doit être retirée de l'attribut Liste des références d'Invocation de Programme.
- b) Pour chaque entrée de l'attribut Liste des références d'Invocation de Programme d'objet Contrôle-Commande d'Unité, placer chaque Invocation de Programme qui se trouve à l'état EN COURS D'EXECUTION à l'état ARRETE. Ceci doit être effectué comme suit:
 - 1) Si l'Invocation de Programme est déjà à l'état ARRETE, EN COURS D'ARRET, REPOS, REINITIALISATION, ou NON EXECUTABLE, ne rien faire.
 - 2) Si l'Invocation de Programme est à l'état EN EXECUTION ou LANCEMENT, exécuter une procédure d'Arrêt (voir 11.6.3 de l'ISO/IEC 9506-1).
 - 3) Si une procédure d'Arrêt sur une Invocation de Programme échoue, renvoyer un Résultat(-) avec un paramètre Erreur Arrêt de Contrôle-Commande d'Unité indiquant l'Invocation de Programme qui a échoué ainsi que son état.
- c) Renvoyer un Résultat(+).

7.3.6.3 Protocole StopUnitControl

La syntaxe abstraite de la sélection stopUC de AdditionalService-Request et AdditionalService-Response est spécifiée par StopUnitControl-Request et StopUnitControl-Response, respectivement. Ces types sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```

StopUnitControl-Request ::= Identifier -- Unit Control Name
StopUnitControl-Response ::= NULL
StopUnitControl-Error ::= SEQUENCE {
    programInvocationName    [0] IMPLICIT Identifier,
    programInvocationState    [1] IMPLICIT ProgramInvocationState
}
    
```

7.3.6.3.1 StopUnitControl-Request (Demande-Arrêt Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection stopUC du type AdditionalService-Request doit être StopUnitControl-Request.

7.3.6.3.2 StopUnitControl-Response (Réponse-Arrêt Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection stopUC du type AdditionalService-Response doit être StopUnitControl-Response.

7.3.6.3.3 StopUnitControl-Error (Erreur Arrêt Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection stopUC de AdditionalService-Error doit être StopUnitControl-Error, qui doit être le sous-paramètre Nom d'Invocation de Programme et

7.3.6.2 Service procedure

The MMS server shall:

- a) For each entry on the List of Program Invocation references attribute of the Unit Control object, verify that the Program Invocation exists. If this condition is not satisfied, remove the reference to the Program Invocation from the List of Program Invocation references attribute.
- b) For each entry on the List of Program Invocation references attribute of the Unit Control object, place each Program Invocation that is in the RUNNING state into the STOPPED state. This shall be done as follows:
 - 1) If the Program Invocation is already in the STOPPED, STOPPING, IDLE, RESETTING, or UNRUNNABLE state, do nothing.
 - 2) If the Program Invocation is in the RUNNING or STARTING state, perform a Stop procedure (see 11.6.3 of ISO/IEC 9506-1).
 - 3) If any Stop procedure on a constituent Program Invocation fails, return a Result(-) with a StopUnitControl-Error parameter indicating the failed Program Invocation and its state.
- c) Return a Result(+).

7.3.6.3 StopUnitControl protocol

The abstract syntax of the stopUC choice of the AdditionalService-Request and the AdditionalService-Response is specified by the StopUnitControl-Request and the StopUnitControl-Response respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```

StopUnitControl-Request ::= Identifier -- Unit Control Name
StopUnitControl-Response ::= NULL
StopUnitControl-Error ::= SEQUENCE {
    programInvocationName [0] IMPLICIT Identifier,
    programInvocationState [1] IMPLICIT ProgramInvocationState
}
  
```

7.3.6.3.1 StopUnitControl-Request

The abstract syntax of the stopUC choice of the AdditionalService-Request type shall be the StopUnitControl-Request.

7.3.6.3.2 StopUnitControl-Response

The abstract syntax of the stopUC choice of the AdditionalService-Response type shall be the StopUnitControl-Response.

7.3.6.3.3 StopUnitControl-Error

The abstract syntax of the stopUC choice of the AdditionalService-Error shall be the StopUnitControl-Error, which shall be Program Invocation Name sub-parameter and

le sous-paramètre Etat d'Invocation de Programme, respectivement, du paramètre Résultat(-) de la primitive StopControlUnit.response et doit apparaître comme étant le sous-paramètre Nom d'Invocation de Programme et le sous-paramètre Etat d'Invocation de Programme, respectivement, de la primitive StopUnitControl.confirm, si elle est émise.

7.3.7 Service CreateUnitControl (Créer Contrôle-Commande d'Unité)

Le service CreateUnitControl est utilisé par un client MMS pour créer un nouvel objet Contrôle-Commande d'Unité ayant un ensemble spécifié de Domaines et/ou d'Invocations de Programme.

7.3.7.1 Structure

La structure des primitives qui constituent ce service est définie dans le tableau 15.

Tableau 15 – Service CreateUnitControl

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle-Commande d'Unité	M	M(=)			
Liste de Domaines	M	M(=)			
Liste d'Invocation de Programme	M	M(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.7.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service CreateUnitControl.

7.3.7.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, est le nom qui doit être attribué à l'Unité de Contrôle-Commande nouvellement créée.

7.3.7.1.1.2 Liste de Domaines

Ce paramètre, du type liste d'identificateurs, doit indiquer zéro ou plusieurs Domaines qui doivent être référencés par l'attribut Liste de Domaines d'objet Contrôle-Commande d'Unité.

7.3.7.1.1.3 Liste d'Invocations de Programme

Ce paramètre, du type liste d'identificateurs, doit indiquer zéro ou plusieurs Invocations de Programme qui doivent être référencées par l'attribut Liste d'Invocations de Programmes d'objet Contrôle-Commande d'Unité.

7.3.7.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi.

the Program Invocation State sub-parameter, respectively, of the Result(-) parameter of the StopControlUnit.response primitive and shall appear as the Program Invocation Name sub-parameter and the Program Invocation State sub-parameter, respectively, of the StopUnitControl.confirm primitive, if issued.

7.3.7 CreateUnitControl service

The CreateUnitControl service is used by an MMS client to create a new Unit Control object with a specified set of Domains and/or Program Invocations.

7.3.7.1 Structure

The structure of the component service primitives is shown in table 15.

Table 15 – CreateUnitControl service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
List of Domains	M	M(=)			
List of Program Invocations	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.7.1.1 Argument

This parameter shall convey the parameters of the CreateUnitControl service request.

7.3.7.1.1.1 Unit Control Name

This parameter, of type identifier, is the name that shall be assigned to the newly created Unit Control object.

7.3.7.1.1.2 List of Domains

This parameter, of type list of identifier, shall specify zero or more Domains that are to be referenced by the List of Domains attribute of the Unit Control object.

7.3.7.1.1.3 List of Program Invocations

This parameter, of type list of identifier, shall specify zero or more Program Invocations that are to be referenced by the List of Program Invocations attribute of the Unit Control object.

7.3.7.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded.

7.3.7.1.3 *Résultat(-)*

Le paramètre *Résultat(-)* doit indiquer que la demande de service a échoué. Le paramètre *Type d'Erreur*, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.7.2 *Procédure de service*

Le serveur MMS doit:

- a) Créer un objet *Contrôle-Commande d'Unité* et lui attribuer le nom spécifié.
- b) Pour chaque élément du paramètre *Liste de Domaines* (le cas échéant), ajouter une référence à ce *Domaine* dans l'attribut *Liste de Domaines d'objet Contrôle-Commande d'Unité*.
- c) Pour chaque élément du paramètre *Liste d'Invocations de Programme* (le cas échéant), ajouter une référence à cette *Invocation de Programme* dans l'attribut *Liste d'Invocations de Programme d'objet Contrôle-Commande d'Unité*.
- d) Renvoyer un *Résultat(+)*.

7.3.7.3 *Protocole CreateUnitControl*

La syntaxe abstraite de la sélection *createUC* de *AdditionalService-Request* et de *AdditionalService-Response* est spécifiée par *CreateUnitControl-Request* et *CreateUnitControl-Response*, respectivement. Ces types sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
CreateUnitControl-Request ::= SEQUENCE {
    unitControl      [0] IMPLICIT Identifier, -- Unit Control
                    Name
    domains          [1] IMPLICIT SEQUENCE OF Identifier
    programInvocations [2] IMPLICIT SEQUENCE OF Identifier
}
CreateUnitControl-Response ::= NULL
```

7.3.7.3.1 *Create Unit Control-Request (Demande-Créer Contrôle-Commande d'Unité)*

La syntaxe abstraite de la sélection *createUC* du type *AdditionalService-Request* doit être *CreateUnitControl-Request*.

7.3.7.3.2 *Create Unit Control-Response (Réponse-Créer Contrôle-Commande d'Unité)*

La syntaxe abstraite de la sélection *createUC* du type *AdditionalService-Response* doit être *CreateUnitControl-Response*.

7.3.8 *Service AddToUnitControl (Ajouter à Contrôle-Commande d'Unité)*

Le service *AddToUnitControl* est utilisé par un client MMS pour ajouter des *Domaines* et/ou des *Invocations de Programme* à l'objet *Contrôle-Commande d'Unité*.

7.3.7.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.7.2 *Service procedure*

The MMS server shall:

- a) Create a Unit Control object and assign it the specified name.
- b) For each element of the List of Domains parameter (if any), add a reference to that Domain to the List of Domains attribute of the Unit Control object.
- c) For each element of the List of Program Invocations parameter (if any), add a reference to that Program Invocation to the List of Program Invocations attribute of the Unit Control object.
- d) Return a Result(+)

7.3.7.3 *CreateUnitControl protocol*

The abstract syntax of the createUC choice of the AdditionalService-Request and the AdditionalService-Response is specified by the CreateUnitControl-Request and the CreateUnitControl-Response, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
CreateUnitControl-Request ::= SEQUENCE {
    unitControl      [0] IMPLICIT Identifier, -- Unit Control Name
    domains          [1] IMPLICIT SEQUENCE OF Identifier,
    programInvocations [2] IMPLICIT SEQUENCE OF Identifier
}
CreateUnitControl-Response ::= NULL
```

7.3.7.3.1 *CreateUnitControl-Request*

The abstract syntax of the createUC choice of the AdditionalService-Request type shall be the CreateUnitControl-Request.

7.3.7.3.2 *CreateUnitControl-Response*

The abstract syntax of the createUC choice of the AdditionalService-Response type shall be the CreateUnitControl-Response.

7.3.8 *AddToUnitControl service*

The AddToUnitControl service is used by an MMS client to add Domains and/or Program Invocations to the Unit Control object.

7.3.8.1 Structure

La structure des primitives qui constituent ce service est définie dans le tableau 16.

Tableau 16 – Service AddToUnitControl

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M (=)			
Nom de Contrôle-Commande d'Unité	M	M (=)			
Liste de Domaines	M	M (=)			
Liste d'Invocation de Programme	M	M (=)			
Résultat(+)			S	S (=)	
Résultat(-)			S	S (=)	
Type d'Erreur			M	M (=)	

7.3.8.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service AddToUnitControl.

7.3.8.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, doit identifier l'objet Contrôle-Commande d'Unité dans le VMD dont la liste de constituants est à modifier.

7.3.8.1.1.2 Liste de Domaines

Ce paramètre, du type liste d'identificateurs, doit préciser zéro ou plusieurs Domaines qui doivent être ajoutées à l'attribut Liste de Domaines d'objet Contrôle-Commande d'Unité.

7.3.8.1.1.3 Liste d'Invocations de Programme

Ce paramètre, du type liste d'identificateurs, doit indiquer zéro ou plusieurs Invocations de Programme qui sont à ajouter à l'attribut Liste d'Invocations de Programme d'objet Contrôle-Commande d'Unité.

7.3.8.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi.

7.3.8.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.8.2 Procédure de service

Le serveur MMS doit:

7.3.8.1 Structure

The structure of the component service primitives is shown in table 16.

Table 16 – AddToUnitControl service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
List of Domains	M	M(=)			
List of Program Invocations	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.8.1.1 Argument

This parameter shall convey the parameters of the AddToUnitControl service request.

7.3.8.1.1.1 Unit Control Name

This parameter, of type identifier, shall identify the Unit Control object in the VMD whose list of constituents is to be altered.

7.3.8.1.1.2 List of Domains

This parameter, of type list of identifier, shall specify zero or more Domains that are to be added to the List of Domains attribute of the Unit Control object.

7.3.8.1.1.3 List of Program Invocations

This parameter, of type list of identifier, shall specify zero or more Program Invocations that are to be added to List of Program Invocations attribute of the Unit Control object.

7.3.8.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded.

7.3.8.1.3 Result(-)

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.8.2 Service procedure

The MMS server shall:

- a) Pour chaque élément du paramètre Liste de Domaines (le cas échéant), ajouter une référence à ce Domaine dans l'attribut Liste de Domaines d'objet Contrôle-Commande d'Unité.
- b) Pour chaque élément du paramètre Liste d'Invocations de Programme (le cas échéant), ajouter une référence à cette Invocation de Programme dans l'attribut Liste d'Invocations de Programme d'objet Contrôle-Commande d'Unité.
- c) Renvoyer un Résultat(+).

7.3.8.3 *Protocole AddToUnitControl*

La syntaxe abstraite de la sélection addToUC de AdditionalService-Request et AdditionalService-Response est spécifiée par AddToUnitControl-Request et AddToUnitControl-Response, respectivement. Ces types sont précisés ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
AddToUnitControl-Request ::= SEQUENCE {
    unitControl          [0] IMPLICIT Identifier, -- Unit Control Name
    domains              [1] IMPLICIT SEQUENCE OF Identifier,
    programInvocations   [2] IMPLICIT SEQUENCE OF Identifier,
}
```

```
AddToUnitControl-Response ::= NULL
```

7.3.8.3.1 *AddToUnitControl-Request (Demande-Ajouter à Unité de Contrôle-Commande)*

La syntaxe abstraite de la sélection addToUC du type AdditionalService-Request doit être AddToUnitControl-Request.

7.3.8.3.2 *AddToUnitControl-Response (Réponse-Ajouter à Unité de Contrôle-Commande)*

La syntaxe abstraite de la sélection addToUC du type AdditionalService-Response doit être AddToUnitControl-Response.

7.3.9 *Service RemoveFromUnitControl (Retirer de Contrôle-Commande d'Unité)*

Le service RemoveFromUnitControl est utilisé par un client MMS pour retirer des Domaines ou des Invocations de Programme ou les deux à la fois d'objet Contrôle-Commande d'Unité.

7.3.9.1 *Structure*

La structure des primitives qui constituent le service est définie dans le tableau 17.

- a) For each element of the List of Domains parameter (if any), add a reference to that Domain to the List of Domains attribute of the Unit Control object.
- b) For each element of the List of Program Invocations parameter (if any), add a reference to that Program Invocation to the List of Program Invocations attribute of the Unit Control object.
- c) Return a Result(+)

7.3.8.3 *AddToUnitControl protocol*

The abstract syntax of the addToUC choice of the AdditionalService-Request and the AdditionalService-Response is specified by the AddToUnitControl-Request and the AddToUnitControl-Response, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
AddToUnitControl-Request ::= SEQUENCE {
    unitControl          [0] IMPLICIT Identifier, -- Unit Control Name
    domains              [1] IMPLICIT SEQUENCE OF Identifier,
    programInvocations   [2] IMPLICIT SEQUENCE OF Identifier
}
```

```
AddToUnitControl-Response ::= NULL
```

7.3.8.3.1 *AddToUnitControl-Request*

The abstract syntax of the addToUC choice of the AdditionalService-Request type shall be the AddToUnitControl-Request.

7.3.8.3.2 *AddToUnitControl-Response*

The abstract syntax of the addToUC choice of the AdditionalService-Response type shall be the AddToUnitControl-Response.

7.3.9 *RemoveFromUnitControl service*

The RemoveFromUnitControl service is used by an MMS client to remove Domains, or Program Invocations, or both from the Unit Control object.

7.3.9.1 *Structure*

The structure of the component service primitives is shown in table 17.

Tableau 17 – Service RemoveFromUnitControl

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle-Commande d'Unité	M	M(=)			
Liste de Domaines	M	M(=)			
Liste d'Invocation de Programme	M	M(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.9.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service RemoveFromUnitControl.

7.3.9.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, doit identifier l'objet Contrôle-Commande d'Unité dans le VMD dont la liste des constituants est à modifier.

7.3.9.1.1.2 Liste de Domaines

Ce paramètre, du type liste d'identificateurs, doit spécifier zéro ou plusieurs Domaines qui sont à retirer de l'attribut Liste des références de Domaines d'objet Contrôle-Commande d'Unité.

7.3.9.1.1.3 Liste d'Invocations de Programme

Ce paramètre, du type liste d'identificateurs, doit spécifier zéro ou plusieurs Invocations de Programme qui sont à retirer de l'attribut Liste des références d'Invocations de Programme d'objet Contrôle-Commande d'Unité.

7.3.9.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi.

7.3.9.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.9.2 Procédure de service

Le serveur MMS doit:

Table 17 – RemoveFromUnitControl service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
List of Domains	M	M(=)			
List of Program Invocations	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.9.1.1 Argument

This parameter shall convey the parameters of the RemoveFromUnitControl service request.

7.3.9.1.1.1 Unit Control Name

This parameter, of type identifier, shall identify the Unit Control object in the VMD whose list of constituents is to be altered.

7.3.9.1.1.2 List of Domains

This parameter, of type list of identifier, shall specify zero or more Domains that are to be removed from the List of Domains reference attribute of the Unit Control object.

7.3.9.1.1.3 List of Program Invocations

This parameter, of type list of identifier, shall specify zero or more Program Invocations that are to be removed from the List of Program Invocations reference attribute of the Unit Control object.

7.3.9.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded.

7.3.9.1.3 Result(-)

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.9.2 Service procedure

The MMS server shall:

- a) Pour chaque élément du paramètre Liste de Domaines (le cas échéant), retirer la référence à ce domaine de l'attribut Liste des références de Domaines d'objet Contrôle-Commande d'Unité.
- b) Pour chaque élément du paramètre Liste d'Invocations de Programme (le cas échéant), retirer la référence à cette Invocation de Programme de l'attribut Liste des références d'Invocations de Programme d'objet Contrôle-Commande d'Unité.
- c) Renvoyer un Résultat(+).

7.3.9.3 *Protocole RemoveFromUnitControl*

La syntaxe abstraite de la sélection removeFromUC de AdditionalService-Request et AdditionalService-Response est spécifiée par RemoveFromUnitControl-Request et RemoveFromUnitControl-Response, respectivement. Ces types sont définis ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
RemoveFromUnitControl-Request ::= SEQUENCE {
    unitControl          [0] IMPLICIT Identifier, -- Unit Control Name
    domains              [1] IMPLICIT SEQUENCE OF Identifier,
    programInvocations   [2] IMPLICIT SEQUENCE OF Identifier
}
```

```
RemoveFromUnitControl-Response ::= NULL
```

7.3.9.3.1 *RemoveFromUnitControl-Request* (Demande-Retirer de Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection removeFromUC du type AdditionalService-Request doit être RemoveFromUnitControl-Request.

7.3.9.3.2 *RemoveFromUnitControl-Response* (Réponse-Retirer de Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection removeFromUC du type AdditionalService-Response doit être RemoveFromUnitControl-Response.

7.3.10 *Service GetUnitControlAttributes (Obtenir Attributs Contrôle-Commande d'Unité)*

Ce service permet au client MMS d'obtenir la liste de Domaines et d'Invocations de Programme qui constituent un Contrôle-Commande d'Unité.

7.3.10.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 18.

- a) For each element of the List of Domains parameter (if any), remove the reference to that Domain from the List of Domains reference attribute of the Unit Control object.
- b) For each element of the List of Program Invocations parameter (if any), remove the reference to that Program Invocation from the List of Program Invocation references attribute of the Unit Control object.
- c) Return a Result(+)

7.3.9.3 *RemoveFromUnitControl protocol*

The abstract syntax of the *removeFromUC* choice of the *AdditionalService-Request* and the *AdditionalService-Response* is specified by the *RemoveFromUnitControl-Request* and the *RemoveFromUnitControl-Response*, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
RemoveFromUnitControl-Request ::= SEQUENCE {
    unitControl          [0] IMPLICIT Identifier, -- Unit Control Name
    domains              [1] IMPLICIT SEQUENCE OF Identifier,
    programInvocations   [2] IMPLICIT SEQUENCE OF Identifier
}
```

```
RemoveFromUnitControl-Response ::= NULL
```

7.3.9.3.1 *RemoveFromUnitControl-Request*

The abstract syntax of the *removeFromUC* choice of the *AdditionalService-Request* type shall be the *RemoveFromUnitControl-Request*.

7.3.9.3.2 *RemoveFromUnitControl-Response*

The abstract syntax of the *removeFromUC* choice of the *AdditionalService-Response* type shall be the *RemoveFromUnitControl-Response*.

7.3.10 *GetUnitControlAttributes service*

This service allows the MMS client get the list of constituent Domains and Program Invocations of a Unit Control object.

7.3.10.1 *Structure*

The structure of the component service primitives is shown in table 18.

Tableau 18 – Service GetUnitControlAttributes

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle-Commande d'Unité	M	M(=)			
Résultat(+)			S	S(=)	
Liste de Domaines			M	M(=)	
Liste d'Invocation de Programme			M	M(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.10.1.1 Argument

Ce paramètre doit véhiculer le paramètre de la demande de service GetUnitControlAttributes.

7.3.10.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre, du type identificateur, doit identifier l'objet Contrôle-Commande d'Unité dont les attributs sont à obtenir.

7.3.10.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que le service a réussi. Dans ce cas, les paramètres suivants doivent apparaître.

7.3.10.1.2.1 Liste de Domaines

Ce paramètre, du type liste d'identificateurs, doit indiquer les noms de Domaines qui constituent l'objet Contrôle-Commande d'Unité.

7.3.10.1.2.2 Liste d'Invocations de Programme

Ce paramètre, du type liste d'identificateurs, doit spécifier les noms d'Invocations de Programme qui constituent l'objet Contrôle-Commande d'Unité.

7.3.10.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.10.2 Procédure de service

Le serveur MMS doit:

- Pour chaque entrée de l'attribut Liste des références d'Invocations de Programme d'objet Contrôle-Commande d'Unité, vérifier que l'Invocation de Programme existe.

Table 18 – GetUnitControlAttributes service

Parameter name	Req	Ind	Resp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Result(+)			S	S(=)	
List of Domains			M	M(=)	
List of Program Invocations			M	M(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.10.1.1 Argument

This parameter shall convey the parameter of the GetUnitControlAttributes service request.

7.3.10.1.1.1 Unit Control Name

This parameter, of type identifier, shall identify the Unit Control object for which the attributes are to be obtained.

7.3.10.1.2 Result(+)

The Result(+) parameter shall indicate that the service has succeeded. If success is indicated, the following parameters shall appear.

7.3.10.1.2.1 List of Domains

This parameter, of type list of identifier, shall specify the names of the Domains that are constituents of the Unit Control object.

7.3.10.1.2.2 List of Program Invocations

This parameter, of type list of identifier, shall specify the names of the Program Invocations that are constituents of the Unit Control object.

7.3.10.1.3 Result(-)

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.10.2 Service procedure

The MMS server shall:

- a) For each entry in the List of Program Invocation references attribute of the Unit Control object, verify that the Program Invocation exists. If the Program Invocation

Si l'Invocation de Programme n'existe pas, retirer sa référence de l'attribut Liste des références d'Invocations de Programme d'objet Contrôle-Commande d'Unité.

- b) Pour chaque entrée dans l'attribut Liste des références de Domaines d'objet Contrôle-Commande d'Unité, vérifier que le Domaine existe. Si le Domaine n'existe pas, retirer sa référence de l'attribut Liste des références de Domaines d'objet Contrôle-Commande d'Unité.
- c) Renvoyer un Résultat(+) avec la liste des noms de Domaines et d'Invocations de Programme tels que spécifiés dans l'attribut Liste de Domaines et dans l'attribut Liste d'Invocations de Programme d'objet Contrôle-Commande d'Unité.

NOTE – Du fait du modèle d'objet Contrôle-Commande d'Unité fourni dans la présente partie de l'ISO/IEC 9506, il est possible que la liste des éléments qui constituent un objet Contrôle-Commande d'Unité ne soit plus cohérente avec l'ensemble proprement dit de Domaines et d'Invocations de Programme, comme par exemple, par suite de l'annulation explicite d'un Domaine. La procédure de service de la présente partie est destinée à rétablir la cohérence de cet objet Contrôle-Commande d'Unité avant de terminer le service. Il est admis qu'une mise en oeuvre réelle choisisse de maintenir la cohérence à tout moment en utilisant un ensemble de références inverses dans chaque Domaine et Invocation de Programme. Cependant, ceci n'est pas exigé. Une autre technique de mise en oeuvre pourrait consister à utiliser les références par nom au sein d'objet Contrôle-Commande d'Unité, en ne rétablissant la cohérence que lorsque cela est exigé.

7.3.10.3 Protocole *GetUnitControlAttributes*

La syntaxe abstraite de la sélection *getUCAAttributes* de *AdditionalService-Request* et *AdditionalService-Response* est spécifiée par *GetUnitControlAttributes-Request* et *GetUnitControlAttributes-Response*, respectivement. Ces types sont définis ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit les dérivations de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
GetUnitControlAttributes-Request ::= Identifier -- Unit Control Name
GetUnitControlAttributes-Response ::= SEQUENCE {
    domains                [0] IMPLICIT SEQUENCE OF Identifier,
    programInvocations     [1] IMPLICIT SEQUENCE OF Identifier
}
```

7.3.10.3.1 *GetUnitControlAttributes-Request* (Demande-Obtenir Attributs de Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection *getUCAAttributes* du type *AdditionalService-Request* doit être *GetUnitControlAttributes-Request*.

7.3.10.3.2 *GetUnitControlAttributes-Response* (Réponse-Obtenir Attributs de Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection *getUCAAttributes* du type *AdditionalService-Response* doit être *GetUnitControlAttributes-Réponse*.

7.3.11 Service *LoadUnitControlFromFile* (Charger Contrôle-Commande d'Unité du Fichier)

Le service *LoadUnitControlFromFile* peut être utilisé par un client MMS pour demander à un serveur MMS de créer l'objet Contrôle-Commande d'Unité et de charger l'objet Contrôle-Commande d'Unité au moyen des informations disponibles localement ou en provenance d'un site tiers.

does not exist, remove its reference from the List of Program Invocation references attribute of the Unit Control object.

- b) For each entry in the List of Domain references attribute of the Unit Control object, verify that the Domain exists. If the Domain does not exist, remove its reference from the List of Domain references attribute of the Unit Control object.
- c) Return a Result(+) with the list of names of the Domains and Program Invocations as specified in the List of Domains attribute and List of Program Invocations attribute of the Unit Control object.

NOTE – Following the model of Unit Control object given in this part of ISO/IEC 9506, it is possible that the list of constituents of a Unit Control object may become inconsistent with the actual set of Domains and Program Invocations, e.g. following the explicit deletion of a Domain. The service procedure of this part is intended to re-establish consistency for this Unit Control object prior to completion of the service. A real implementation may choose to maintain consistency at all times by employing a set of inverse references in each Domain and Program Invocation. However, this is not required. An alternate implementation technique could be to implement the references within the Unit Control object by name, re-establishing consistency only when required.

7.3.10.3 *GetUnitControlAttributes protocol*

The abstract syntax of the *getUCAAttributes* choice of the *AdditionalService-Request* and the *AdditionalService-Response* is specified by the *GetUnitControlAttributes-Request* and the *GetUnitControlAttributes-Response*, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
GetUnitControlAttributes-Request ::= Identifier -- Unit Control Name
GetUnitControlAttributes-Response ::= SEQUENCE {
    domains                [0] IMPLICIT SEQUENCE OF Identifier,
    programInvocations     [1] IMPLICIT SEQUENCE OF Identifier
}
```

7.3.10.3.1 *GetUnitControlAttributes-Request*

The abstract syntax of the *getUCAAttributes* choice of the *AdditionalService-Request* type shall be the *GetUnitControlAttributes-Request*.

7.3.10.3.2 *GetUnitControlAttributes-Response*

The abstract syntax of the *getUCAAttributes* choice of the *AdditionalService-Response* type shall be the *GetUnitControlAttributes-Response*.

7.3.11 *LoadUnitControlFromFile service*

The *LoadUnitControlFromFile* service may be used by a MMS client to request a MMS server to create a Unit Control object and load the Unit Control object using information available locally or from a third party.

7.3.11.1 Structure

La Structure des primitives qui constituent ce service est définie dans le tableau 19.

Tableau 19 – Service LoadUnitControlFromFile

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			TPY
Nom de Contrôle-Commande d'Unité	M	M(=)			
Nom de Fichier	M	M(=)			
Tiers	U	U(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Erreur Déclenchement Contrôle-Commande d'Unité			M	M(=)	
Nom du Domaine			S	S(=)	
Nom d'Invocation de Programme			S	S(=)	

7.3.11.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de Service LoadUnitControl FromFile.

7.3.11.1.1.1 Nom de Contrôle-Commande d'Unité

Ce paramètre doit spécifier le nom d'objet Contrôle-Commande d'Unité à charger au niveau du VMD.

7.3.11.1.1.2 Nom du Fichier

Ce paramètre, du type Nom du Fichier, doit spécifier le nom du fichier contenant les informations à charger.

7.3.11.1.1.3 Tiers

Ce paramètre, du type Référence d'Application, doit spécifier la référence d'application du Processus d'Application par lequel on peut accéder au fichier nommé. Le support de traitement de ce paramètre est une option de mise en oeuvre qui doit être appliquée si une prise en charge du bloc de conformité de paramètre TPY est demandée. S'il est mis en oeuvre, son utilisation est facultative. Si ce paramètre est absent, le serveur MMS doit tenter d'accéder directement au fichier demandé.

7.3.11.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Ce résultat n'apporte pas de paramètres spécifiques au service.

7.3.11.1 Structure

The Structure of the component service primitives is shown in table 19.

Table 19 – LoadUnitControlFromFile service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
File Name	M	M(=)			
Third Party	U	U(=)			TPY
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Initiate Unit Control Error			M	M(=)	
Domain Name			S	S(=)	
Program Invocation Name			S	S(=)	

7.3.11.1.1 Argument

This parameter shall convey the parameters of the LoadUnitControlFromFile service request.

7.3.11.1.1.1 Unit Control Name

This parameter shall specify the name of the Unit Control object at the VMD to be loaded.

7.3.11.1.1.2 File Name

This parameter, of type FileName, shall specify the name of the file containing the information to be loaded.

7.3.11.1.1.3 Third Party

This parameter, of type ApplicationReference, shall specify the application reference of the Application Process through which the named file may be accessed. Support of processing for this parameter is an implementation option that shall be implemented if support for the TPY parameter conformance building block is claimed. If it is implemented, its use is a user option. If this parameter is absent, the MMS server shall attempt to access the requested file directly.

7.3.11.1.2 Result(+)

The Result(+) parameter shall indicate that the service request succeeded. A successful result does not supply service specific parameters.

7.3.11.1.3 *Résultat(-)*

Le paramètre *Résultat(-)* doit indiquer que la demande de service a échoué. Le paramètre *Type d'Erreur*, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec.

7.3.11.1.3.1 *Nom de Domaine*

Ce paramètre doit indiquer le *Domaine* en cours de création lorsque l'erreur a été détectée. Il est admis que ce paramètre ou le *Nom d'Invocation de Programme* soit présent, mais non les deux à la fois.

7.3.11.1.3.2 *Nom d'Invocation de Programme*

Ce paramètre doit indiquer l'*Invocation de Programme* en cours de création lorsque l'erreur a été détectée. Il est admis que ce paramètre ou le *nom de Domaine* soit présent mais non les deux à la fois.

7.3.11.2 *Procédure de service*

Le VMD doit vérifier que l'objet *Contrôle-Commande d'Unité* de nom spécifié n'existe pas. Lorsqu'un tiers est spécifié, il faut établir une association avec cette application s'il n'en existe aucune; il faut ensuite prendre les mesures nécessaires pour que l'objet *Contrôle-Commande d'Unité* nommée soit chargée. Si aucun tiers n'est spécifié, prendre les mesures nécessaires pour obtenir le fichier par des moyens définis localement et le charger dans l'objet *Contrôle-Commande d'Unité* spécifiée. Si le chargement est réussi, renvoyer un *Résultat(+)*; autrement, renvoyer un *Résultat(-)* en indiquant la raison de l'échec dans le paramètre *Erreur-Déclenchement de Contrôle-Commande d'Unité*.

7.3.11.3 *Protocole LoadUnitControlFromFile*

La syntaxe abstraite de la sélection *loadUCFromFile* de *ConfirmedServiceRequest* et *ConfirmedServiceResponse* est spécifiée par *LoadUnitControlFromFile-Request* et *LoadUnitControlFromFile-Response* définis ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
LoadUnitControlFromFile-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier,
    fileName             [1] IMPLICIT FileName,
    thirdParty           [2] IMPLICIT ApplicationReference OPTIONAL
}

LoadUnitControlFromFile-Response ::= NULL

LoadUnitControlFromFile-Error ::= CHOICE {
    none                [0] IMPLICIT NULL,
    domain              [1] IMPLICIT Identifier
    programInvocation   [2] IMPLICIT Identifier
}
```

7.3.11.1.3 *Result (-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1, shall provide the reason for failure.

7.3.11.1.3.1 *Domain Name*

This parameter shall indicate the Domain that was being created when the error was detected. Either this parameter or the Program Invocation Name parameter may be present, but not both.

7.3.11.1.3.2 *Program Invocation Name*

This parameter shall indicate the Program Invocation that was being created when the error was detected. Either this parameter or the Domain Name parameter may be present but not both.

7.3.11.2 *Service procedure*

The VMD shall verify that the Unit Control object of the specified name does not exist. If a third party is specified, establish an association with that application if none exists; thereafter take appropriate action to cause the named Unit Control object to be loaded. If no third party is specified, perform the necessary steps to obtain the file through local means and load it into the specified Unit Control object. If the loading is successful, return a Result(+); otherwise return a Result(-) indicating in the Initiate Unit Control Error parameter the reason for failure.

7.3.11.3 *LoadUnitControlFromFile protocol*

The abstract syntax of the loadUCFromFile choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified by the LoadUnitControlFromFile-Request and LoadUnitControlFromFile-Response specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
LoadUnitControlFromFile-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier,
    fileName             [1] IMPLICIT FileName,
    thirdParty           [2] IMPLICIT ApplicationReference OPTIONAL
}
```

```
LoadUnitControlFromFile-Response ::= NULL
```

```
LoadUnitControlFromFile-Error ::= CHOICE {
    none                [0] IMPLICIT NULL,
    domain              [1] IMPLICIT Identifier,
    programInvocation   [2] IMPLICIT Identifier
}
```

7.3.11.3.1 *LoadUnitControlFromFile-Request* (Demande-Charger Contrôle-Commande d'Unité du Fichier)

La syntaxe abstraite de la sélection loadUCFromFile du type AdditionalService-Request doit être LoadUnitControlFromFile-Request.

7.3.11.3.2 *LoadUnitControlFromFile-Response* (Réponse-Charger Contrôle-Commande d'Unité du Fichier)

La syntaxe abstraite de la sélection loadUCFromFile du type AdditionalService-Response doit être LoadUnitControlFromFile-Response.

7.3.11.3.3 *LoadUnitControlFromFile-Error* (Erreur-Charger Unité de Contrôle-Commande du Fichier)

La syntaxe abstraite de la sélection loadUCFromFile du type AdditionalService-Error doit être LoadUnitControlFromFile-Error.

7.3.12 *Service StoreUnitControlToFile* (Archiver Contrôle-Commande d'Unité dans Fichier)

Il est admis que le service StoreUnitControlToFile soit utilisé par un client MMS pour demander à un serveur MMS d'archiver les Domaines et Invocations de Programme d'objet Contrôle-Commande d'Unité, soit sur un site tiers, soit localement.

7.3.12.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 20.

Tableau 20 – Service StoreUnitControlToFile

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			TPY
Nom de Contrôle-Commande d'Unité	M	M(=)			
Nom de fichier	M	M(=)			
Tiers	U	U(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.12.1.1 *Argument*

Ce paramètre doit véhiculer les paramètres de la demande de service StoreUnitControlToFile.

7.3.12.1.1.1 *Nom de Contrôle-Commande d'Unité*

Ce paramètre doit spécifier le nom d'objet Contrôle-Commande d'Unité, au niveau du VMD, dont le contenu est à archiver.

7.3.11.3.1 *LoadUnitControlFromFile-Request*

The abstract syntax of the loadUCFromFile choice of the AdditionalService-Request type shall be the LoadUnitControlFromFile-Request.

7.3.11.3.2 *LoadUnitControlFromFile-Response*

The abstract syntax of the loadUCFromFile choice of the AdditionalService-Response type shall be the LoadUnitControlFromFile-Response.

7.3.11.3.3 *LoadUnitControlFromFile-Error*

The abstract syntax of the loadUCFromFile choice of the AdditionalService-Error type shall be the LoadUnitControlFromFile-Error.

7.3.12 *StoreUnitControlToFile service*

The StoreUnitControlToFile service may be used by a MMS client to request a MMS server to store the Domains and Program Invocations of a Unit Control object either at a third party site or locally.

7.3.12.1 *Structure*

The structure of the component service primitives is shown in table 20.

Table 20 – StoreUnitControlToFile service

Parameter name	Req	Ind	Rsp	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
File Name	M	M(=)			
Third Party	U	U(=)			TPY
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.12.1.1 *Argument*

This parameter shall convey the parameters of the StoreUnitControlToFile service request.

7.3.12.1.1.1 *Unit Control Name*

This parameter shall specify the name of the Unit Control object at the VMD for which the content is to be stored.

7.3.12.1.1.2 *Nom de Fichier*

Ce paramètre, du type *NomdeFichier*, doit spécifier le nom du fichier dont les informations sont à archiver.

7.3.12.1.1.3 *Tiers*

Ce paramètre facultatif, du type *Référenced'Application*, doit spécifier la référence de l'application du Processus d'Application sur lequel réside la mémoire d'enregistrement de fichiers qui doit recevoir le contenu d'objet Contrôle-Commande d'Unité spécifiée. La prise en charge du traitement de ce paramètre est une option de mise en oeuvre qui sera exécutée si la prise en charge du bloc de conformité du paramètre TPY est demandée. S'il est mis en oeuvre, son utilisation est facultative. Si ce paramètre est absent, le serveur MMS doit tenter d'archiver directement le contenu d'objet Contrôle-Commande d'Unité dans le fichier spécifié.

7.3.12.1.2 *Résultat(+)*

Le paramètre *Résultat(+)* doit indiquer que la demande de service a réussi. Ce résultat n'apporte pas de paramètres spécifiques au service.

7.3.12.1.3 *Résultat(-)*

Le paramètre *Résultat(-)* doit indiquer que la demande de service a échoué. Le paramètre *Type d'Erreur*, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec.

7.3.12.2 *Procédure de service*

Le serveur MMS doit vérifier que l'objet Contrôle-Commande d'Unité de nom spécifié existe. Lorsque le paramètre tiers a été fourni, le serveur MMS doit établir une association avec cette application s'il n'en existe aucune; ensuite, il doit prendre les mesures nécessaires pour que l'objet Contrôle-Commande d'Unité nommée soit archivée sur le site Tiers. Si aucun tiers n'est spécifié, le serveur MMS doit prendre les mesures nécessaires pour archiver l'objet Contrôle-Commande d'Unité dans le fichier spécifié en utilisant des moyens disponibles localement. Si la procédure est réussie, renvoyer un *Résultat(+)*; autrement, renvoyer un *Résultat(-)*.

7.3.12.3 *Protocole StoreUnitControlToFile*

La syntaxe abstraite de la sélection *storeUCToFile* de *ConfirmedServiceRequest* et *ConfirmedServiceResponse* est spécifiée par *StoreUnitControlToFile-Request* et *StoreUnitControlToFile-Response* définis ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
StoreUnitControlToFile-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier,
    filename             [1] IMPLICIT FileName,
    thirdParty           [2] IMPLICIT ApplicationReference OPTIONAL
}

StoreUnitControlToFile-Response ::= NULL
```

7.3.12.1.1.2 *File Name*

This parameter, of type *FileName*, shall specify the name of the file to which the information is to be stored.

7.3.12.1.1.3 *Third Party*

This optional parameter, of type *ApplicationReference*, shall specify the application reference of the Application Process on which the file store resides that is to receive the contents of the specified Unit Control object. Support of processing for this parameter is an implementation option that shall be implemented if support for the *TPY* parameter conformance building block is claimed. If it is implemented, its use is a user option. If this parameter is absent, the MMS server shall attempt to store the content of the Unit Control object directly in the specified file.

7.3.12.1.2 *Result(+)*

The *Result(+)* parameter shall indicate that the service request succeeded. A successful result does not supply service specific parameters.

7.3.12.1.3 *Result(-)*

The *Result(-)* parameter shall indicate that the service request failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1, shall provide the reason for failure.

7.3.12.2 *Service procedure*

The MMS server shall verify that the Unit Control object of the specified name exists. If the Third Party parameter has been provided, the MMS server shall establish an association with that application if none exists; thereafter it shall take appropriate action to cause the named Unit Control object to be stored at the Third Party site. If no Third Party is specified, the MMS server shall perform the necessary steps to store the Unit Control object in the file specified through local means. If the process is successful, return a *Result(+)*; otherwise return a *Result(-)*.

7.3.12.3 *StoreUnitControlToFile protocol*

The abstract syntax of the *storeUCToFile* choice of the *ConfirmedService-Request* and *ConfirmedService-Response* is specified by the *StoreUnitControlToFile-Request* and *StoreUnitControlToFile-Response* specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
StoreUnitControlToFile-Request ::= SEQUENCE {
    unitControlName      [0] IMPLICIT Identifier,
    fileName             [1] IMPLICIT FileName,
    thirdParty           [2] IMPLICIT ApplicationReference OPTIONAL
}

StoreUnitControlToFile-Response ::= NULL
```

7.3.12.3.1 *StoreUnitControlToFile-Request* (Demande-Archiver Contrôle-Commande d'Unité dans Fichier)

La syntaxe abstraite de la sélection storeUCToFile du type AdditionalService-Request doit être StoreUnitControlToFile-Request.

7.3.12.3.2 *StoreUnitControlToFile-Response* (Réponse-Archiver Contrôle-Commande d'Unité dans Fichier)

La syntaxe abstraite de la sélection storeUCToFile du type AdditionalService-Response doit être StoreUnitControlToFile-Response.

7.3.13 *Service DeleteUnitControl* (Détruire Contrôle-Commande d'Unité)

Ce service permet au client MMS de détruire l'objet Contrôle-Commande d'Unité ainsi que tous les éléments qui la constituent.

7.3.13.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 21.

Tableau 21 – Service DeleteUnitControl

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Contrôle-Commande d'Unité	M	M(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Erreur Détruire Contrôle-Commande d'Unité			M	M(=)	
Nom de Domaine			S	S(=)	
Nom d'Invocation de Programme			S	S(=)	

7.3.13.1.1 *Argument*

Ce paramètre véhicule le paramètre du service DeleteUnitControl.

7.3.13.1.1.1 *Nom de Contrôle-Commande d'Unité*

Ce paramètre, du type identificateur, doit identifier l'objet Contrôle-Commande d'Unité qui est à détruire en même temps que ses éléments constituants.

7.3.13.1.2 *Résultat(+)*

Le paramètre Résultat(+) doit indiquer que le service a réussi.

7.3.12.3.1 *StoreUnitControlToFile-Request*

The abstract syntax of the storeUCToFile choice of the AdditionalService-Request type shall be the StoreUnitControlToFile-Request.

7.3.12.3.2 *StoreUnitControlToFile-Response*

The abstract syntax of the storeUCToFile choice of the AdditionalService-Response type shall be the StoreUnitControlToFile-Response.

7.3.13 *DeleteUnitControl service*

This service allows the MMS client to delete the Unit Control object and all its constituent elements.

7.3.13.1 *Structure*

The structure of the component service primitives is shown in table 21.

Table 21 – DeleteUnitControl service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Unit Control Name	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Delete Unit Control Error			M	M(=)	
Domain Name			S	S(=)	
Program Invocation Name			S	S(=)	

7.3.13.1.1 *Argument*

This parameter conveys the parameter of the DeleteUnitControl service.

7.3.13.1.1.1 *Unit Control Name*

This parameter, of type identifier, shall identify the Unit Control object that is to be deleted with its constituent elements.

7.3.13.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service has succeeded.

7.3.13.1.3 *Résultat(-)*

Le paramètre *Résultat(-)* doit indiquer que la demande de service a échoué. Le paramètre *Type d'Erreur*, qui est défini plus en détail dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec.

7.3.13.1.3.1 *Nom du Domaine*

Ce paramètre doit indiquer le Domaine dont la destruction était en cours lorsque l'erreur a été détectée. Ce paramètre ou le Nom d'Invocation de Programme doit être sélectionné.

7.3.13.1.3.2 *Nom d'Invocation de Programme*

Ce paramètre doit indiquer l'Invocation de Programme dont la destruction était en cours lorsque l'erreur a été détectée. Ce paramètre ou le paramètre *Nom de Domaine* doit être sélectionné.

7.3.13.2 *Procédure de service*

Le serveur MMS doit:

- a) Pour chaque entrée de l'attribut *Liste des références d'Invocation de Programme d'objet Contrôle-Commande d'Unité*:
 - 1) Vérifier que l'Invocation de Programme existe. Si l'Invocation de Programme n'existe pas, retirer sa référence de l'attribut *Liste des références d'Invocation de Programme d'objet Contrôle-Commande d'Unité* et omettre le reste de la présente procédure pour ce qui concerne cette entrée.
 - 2) Vérifier que l'Invocation de Programme n'est pas en l'état **EN EXECUTION**. Si cette condition n'est pas remplie, renvoyer un *Résultat(-)* et omettre le reste de la procédure.
 - 3) Exécuter une procédure de service *DeleteProgramInvocation* (Détruire Invocation de Programme) comme spécifié dans 11.3.2 de l'ISO/IEC 9506-1 et retirer la référence à cette Invocation de Programme de l'attribut *Liste de références d'Invocation de Programme d'objet Contrôle-Commande d'Unité*.
- b) Pour chaque entrée de l'attribut *Liste des références de Domaine d'objet Contrôle-Commande d'Unité*:
 - 1) Vérifier que le Domaine existe. Si le Domaine n'existe pas, retirer sa référence de l'attribut *Liste des références de Domaine d'objet Contrôle-Commande d'Unité* et omettre le reste de cette procédure pour ce qui concerne cette entrée.
 - 2) Vérifier que le Domaine n'est pas en l'état **EN UTILISATION**. Si cette condition n'est pas remplie, renvoyer un *Résultat(-)* et omettre le reste de la procédure.
 - 3) Exécuter une procédure de service *DeleteDomain* (Détruire Domaine) comme spécifié dans 10.12.2 de l'ISO/IEC 9506-1 et retirer la référence à ce Domaine de l'attribut *Liste des références de Domaine d'objet Contrôle-Commande d'Unité*.
- c) Détruire l'objet *Contrôle-Commande d'Unité* du VMD.
- d) Renvoyer un *Résultat(+)*.

7.3.13.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request has failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1, shall provide the reason for failure.

7.3.13.1.3.1 *Domain Name*

This parameter shall indicate the Domain whose deletion was being attempted when the error was detected. Either this parameter or the Program Invocation Name parameter shall be selected.

7.3.13.1.3.2 *Program Invocation Name*

This parameter shall indicate the Program Invocation whose deletion was being attempted when the error was detected. Either this parameter or the Domain Name parameter shall be selected.

7.3.13.2 *Service procedure*

The MMS server shall:

- a) For each entry in the List of Program Invocation references attribute of the Unit Control object:
 - 1) Verify that the Program Invocation exists. If the Program Invocation does not exist, remove its reference from the List of Program Invocation references attribute of the Unit Control object and skip the remainder of this step for this entry.
 - 2) Verify that the Program Invocation is not in the RUNNING state. If this condition is not satisfied, return a Result(-) and skip the remainder of the procedure.
 - 3) Perform a DeleteProgramInvocation service procedure as specified in 11.3.2 of ISO/IEC 9506-1 and remove the reference to this Program Invocation from the List of Program Invocation references attribute of the Unit Control object.
- b) For each entry in the List of Domain references attribute of the Unit Control object:
 - 1) Verify that the Domain exists. If the Domain does not exist, remove its reference from the List of Domain references attribute of the Unit Control object and skip the remainder of this step for this entry.
 - 2) Verify that the Domain is not in the IN-USE state. If this condition is not satisfied, return a Result(-) and skip the remainder of the procedure.
 - 3) Perform a DeleteDomain service procedure as specified in 10.12.2 of ISO/IEC 9506-1 and remove the reference to this Domain from the List of Domain references attribute of the Unit Control object.
- c) Delete the Unit Control object from the VMD.
- d) Return a Result(+).

Si cette procédure renvoie à un Résultat(-), il est admis que seulement certains des Domaines et Invocations de Programme de l'objet Contrôle-Commande d'Unité aient été détruits. Dans ce cas, le paramètre DeleteErrorControl doit indiquer le Domaine ou l'Invocation de Programme sur laquelle la procédure s'est arrêtée et le contenu courant de l'objet Contrôle-Commande d'Unité doit indiquer les Domaines et Invocations de Programme qui restent encore sur l'objet Contrôle-Commande d'Unité.

NOTE – Du fait du modèle de Contrôle-Commande d'Unité fourni dans la présente partie de l'ISO/IEC 9506, il est possible que la liste des éléments qui constituent un Contrôle-Commande d'Unité ne soit plus cohérente avec l'ensemble proprement dit de Domaines et d'Invocations de Programme, comme par exemple, par suite de l'annulation explicite d'un Domaine. La procédure de service de la présente partie est destinée à rétablir la cohérence de ce Contrôle-Commande d'Unité avant de terminer le service. Il est admis qu'une mise en oeuvre réelle choisisse de maintenir la cohérence à tout moment en utilisant un ensemble de références inverses dans chaque Domaine et Invocation de Programme. Cependant, ceci n'est pas exigé. Une autre technique de mise en oeuvre pourrait consister à utiliser les références par nom au sein d'objet Contrôle-Commande d'Unité, en ne rétablissant la cohérence que lorsque cela est exigé.

7.3.13.3 Protocole DeleteUnitControl

La syntaxe abstraite de la sélection deleteUC de AdditionalService-Request et AdditionalService-Response est spécifiée par DeleteUnitControl-Request et DeleteUnitControl-Response, respectivement. Ces types sont définis ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
DeleteUnitControl-Request ::= Identifier -- Unit Control Name
DeleteUnitControl-Response ::= NULL
DeleteUnitControl-Error ::= CHOICE {
    domain [0] IMPLICIT Identifier,
    programInvocation [1] IMPLICIT Identifier
}
```

7.3.13.3.1 DeleteUnitControl-Request (Demande-Détruire Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection deleteUC du type AdditionalService-Request doit être DeleteUnitControl-Request.

7.3.13.3.2 DeleteUnitControl-Response (Réponse-Détruire Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection deleteUC du type AdditionalService-Response doit être DeleteUnitControl-Response.

7.3.13.3.3 DeleteUnitControl-Error (Erreur-Détruire Contrôle-Commande d'Unité)

La syntaxe abstraite de la sélection deleteUC du type AdditionalService-Error doit être DeleteUnitControl-Error.

7.3.14 Service DefineEventConditionList (Définir Liste de Conditions Événementielles)

Le service DefineEventConditionList doit être utilisé pour obtenir la création d'un objet Liste de Conditions Événementielles au niveau du VMD.

If the procedure returns a Result(-), the Unit Control object may have had some of its Domains and Program Invocations deleted. In this case, the Delete Unit Control Error parameter shall indicate the Domain or Program Invocation on which the procedure stopped, and the current contents of the Unit Control object shall indicate the Domains and Program Invocations that still remain on the Unit Control object.

NOTE – Following the model of Unit Control object given in this part of ISO/IEC 9506, it is possible that the list of constituents of a Unit Control object may become inconsistent with the actual set of Domains and Program Invocations, e.g. following the explicit deletion of a Domain. The service procedure of this section is intended to reestablish consistency for this Unit Control object prior to completion of the service. A real implementation may choose to maintain consistency at all times by employing a set of inverse references in each Domain and Program Invocation. However, this is not required. An alternate implementation technique could be to implement the references within the Unit Control object by name, reestablishing consistency only when required.

7.3.13.3 *DeleteUnitControl protocol*

The abstract syntax of the deleteUC choice of the AdditionalService-Request and the AdditionalService-Response is specified by the DeleteUnitControl-Request and the DeleteUnitControl-Response, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```

DeleteUnitControl-Request ::= Identifier -- Unit Control Name
DeleteUnitControl-Response ::= NULL
DeleteUnitControl-Error ::= CHOICE {
    domain                [0] IMPLICIT Identifier,
    programInvocation     [1] IMPLICIT Identifier
}

```

7.3.13.3.1 *DeleteUnitControl-Request*

The abstract syntax of the deleteUC choice of the AdditionalService-Request type shall be the DeleteUnitControl-Request.

7.3.13.3.2 *DeleteUnitControl-Response*

The abstract syntax of the deleteUC choice of the AdditionalService-Response type shall be the DeleteUnitControl-Response.

7.3.13.3.3 *DeleteUnitControl-Error*

The abstract syntax of the deleteUC choice of the AdditionalService-Error type shall be the DeleteUnitControl-Error.

7.3.14 *DefineEventConditionList service*

The DefineEventConditionList service shall be used to cause the creation of an Event Condition List object at the VMD.

7.3.14.1 Structure

La structure des primitives qui constituent ce service est définie dans le tableau 22.

Tableau 22 – Service DefineEventConditionList

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			RECL
Nom de Liste de Conditions Événementielles	M	M(=)			
Liste de noms de Conditions Événementielles	M	M(=)			
Liste de noms de Listes de Conditions Événementielles	C	C(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Objet erroné			C	C(=)	

7.3.14.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service DefineEventConditionList.

7.3.14.1.1.1 Nom de Liste de Conditions Événementielles

Ce paramètre, du type Nom d'Objet, doit spécifier le nom de l'objet Liste de Conditions Événementielles à créer au niveau du VMD.

7.3.14.1.1.2 Liste de noms de Conditions Événementielles

Ce paramètre doit contenir une liste des attributs nom d'objet Condition Événementielle qui doit être incluse dans la liste de Conditions Événementielles spécifiée. Cette liste ne doit pas être vide si RECL n'a pas été pris en compte. S'il n'y a pas de nom de Conditions Événementielles dans la demande de service, ce paramètre doit être une liste vide. Si le paramètre nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, ce paramètre ne doit pas contenir un nom de Conditions Événementielles de portée AA.

7.3.14.1.1.3 Liste de noms de Listes de Conditions Événementielles

Ce paramètre doit contenir une liste des attributs nom d'objets Liste de Conditions Événementielles qui doit être incluse, par référence à chaque objet inclus, dans la Liste de Conditions Événementielles. Ce paramètre ne doit pas être présent si le CBB RECL n'a pas été pris en compte. Si RECL a été pris en compte mais qu'il n'y a aucun nom de Liste de Conditions Événementielles dans cette demande de service, le paramètre doit être une liste vide. Si le paramètre nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, ce paramètre ne doit pas contenir un nom de Liste de Conditions Événementielles à portée AA.

7.3.14.1 Structure

The structure of the component service primitives is shown in table 22.

Table 22 – DefineEventConditionList service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Event Condition List name	M	M(=)			
List of Event Condition names	M	M(=)			
List of Event Condition List names	C	C(=)			RECL
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Object in error			C	C(=)	

7.3.14.1.1 Argument

This parameter shall convey the parameters of the DefineEventConditionList service request.

7.3.14.1.1.1 Event Condition List name

This parameter, of type Object Name, shall specify the name of the Event Condition List object that is to be created at the VMD.

7.3.14.1.1.2 List of Event Condition names

This parameter shall contain a list of name attributes of Event Condition objects that shall be included in the specified Event Condition List. This list shall not be empty if RECL has not been negotiated. If there are no Event Condition names in this service request, this parameter shall be an empty list. If the scope of the Event Condition List name parameter is VMD-specific or Domain-specific, this parameter shall not contain an Event Condition name whose scope is AA-specific.

7.3.14.1.1.3 List of Event Condition List names

This parameter shall contain a list of name attributes of Event Condition List objects that shall be included, by reference to each included object, in the Event Condition List. This parameter shall not be present if the RECL CBB has not been negotiated. If RECL has been negotiated but there are no Event Condition List names in this service request, this parameter shall be an empty list. If the scope of the Event Condition List name parameter is VMD-specific or Domain-specific, this parameter shall not contain an Event Condition List name whose scope is AA-specific.

7.3.14.1.2 *Résultat(+)*

Le paramètre *Résultat(+)* doit indiquer que la demande de service a réussi. Dans ce cas, le résultat ne renvoie aucun paramètre spécifique au service.

7.3.14.1.3 *Résultat(-)*

Le paramètre *Résultat(-)* doit indiquer que la demande de service a échoué. Le paramètre *Type d'Erreur*, qui est défini dans l'article 17 de l'ISO/IEC 9506-1 doit donner la raison de l'échec. En outre, le paramètre suivant peut apparaître.

7.3.14.1.3.1 *Objet erroné*

Ce paramètre, du type *Nom d'Objet*, doit être présent lorsque l'erreur concerne la non existence ou l'incohérence de l'objet *Condition Événementielle* spécifié dans le paramètre *Liste de noms de Conditions Événementielles*, ou un objet *Liste de Conditions Événementielles* spécifié dans le paramètre *Liste de noms de Listes de Conditions Événementielles*. Il doit fournir le nom de l'objet dont l'erreur a été déterminée au niveau du VMD. Ce paramètre ne doit pas être présent lorsque l'échec de ce service n'est pas dû à la non existence ou à l'incohérence d'un objet *Condition Événementielle* ou d'un objet *Liste de Conditions Événementielles*.

7.3.14.2 *Procédure de service*

Le VMD doit vérifier qu'au niveau du VMD il n'existe pas des objets *Liste de Conditions Événementielles* ayant un nom égal à la valeur du paramètre *nom de Liste de Conditions Événementielles*.

S'il est déterminé que l'un des objets *Condition Événementielle* spécifié dans le paramètre *Liste de noms de Conditions Événementielles* n'existe pas au niveau du VMD ou que l'un des objets *Liste de Conditions Événementielles* spécifié dans le paramètre *Liste de noms de Listes de Conditions Événementielles* n'existe pas au niveau du VMD, le VMD doit émettre la primitive de service *Résultat(-)* avec une *Classe d'Erreur d'ACCES*, un *Code d'Erreur d'OBJET-NON-EXISTANT*, ainsi que le paramètre *Objet erroné*.

Si le paramètre *nom de Liste de Conditions Événementielles* a une portée VMD ou *Domaine*, et si le nom de l'une des *Conditions Événementielles* dans le paramètre *Liste de noms de Conditions Événementielles* ou de l'une des *Conditions Événementielles* du paramètre *Liste de noms de Listes de Conditions Événementielles* a une portée AA, le VMD doit émettre une primitive de service *Résultat(-)* avec une *Classe d'Erreur DEFINITION*, un *Code d'Erreur ATTRIBUT-D'OBJET-INCOHERENT*, ainsi que le paramètre *Objet erroné*.

Si la définition demandée est acceptable, un nouvel objet *Liste de Conditions Événementielles* doit être créé et initialisé comme indiqué ci-après. Cet objet doit alors faire partie de l'état d'association sur lequel la définition demandée a été reçue, il doit faire partie du VMD ou d'un *Domaine* du VMD, selon la portée du paramètre *nom de Listes de Conditions Événementielles*.

Si le paramètre *Liste de noms de Conditions Événementielles* a été fourni, pour chaque objet *Condition Événementielle* spécifié dans le paramètre *Liste de noms de Conditions Événementielles*, le VMD doit inclure une référence à l'objet *Liste de Conditions Événementielles* nouvellement créé dans l'attribut *Liste des références de Liste de Conditions Événementielles de référence de l'objet Condition Événementielle*.

7.3.14.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service request succeeded. A successful result shall return no service-specific parameters.

7.3.14.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure. In addition, the following parameter may appear.

7.3.14.1.3.1 *Object in error*

This parameter, of type Object Name, shall be present when the error concerns the non-existence or inconsistency of an Event Condition object specified in the List of Event Condition names parameter, or an Event Condition List object specified in the List of Event Condition List names parameter. It shall provide the name of the object determined in error at the VMD. This parameter shall not be present when the failure of this service is not due to the nonexistence or inconsistency of an Event Condition object or a Event Condition List object.

7.3.14.2 *Service procedure*

The VMD shall verify that no other Event Condition List objects exist at the VMD with a name equal to the value of the Event Condition List name parameter.

If one of the Event Condition objects specified in the List of Event Condition names parameter is determined to not exist at the VMD, or if one of the Event Condition List objects specified in the List of Event Condition List names parameter is determined not to exist at the VMD, the VMD shall issue the Result(-) service primitive with an Error Class of ACCESS, Error Code of OBJECT-NON-EXISTENT, and the Object in error parameter.

If the scope of the Event Condition List name parameter is VMD-specific or Domain-specific, and the scope of the name of any of the Event Condition in the List of Event Condition names parameter or of any of the List of Event Condition List names parameter is AA-specific, the VMD shall issue a Result(-) service primitive with an Error Class of DEFINITION, Error Code of OBJECT-ATTRIBUTE-INCONSISTENT, and the Object in error parameter.

If the requested definition is acceptable, a new Event Condition List object shall be created and initialized as below. This object shall then become part of the state of the association over which the requested definition was received, of the VMD or of a Domain of the VMD, depending on the scope of the Event Condition List name parameter.

If the List of Event Condition names parameter has been provided, for every Event Condition object specified in the List of Event Condition names parameter, the VMD shall place a reference to the newly created Event Condition List object in the Event Condition object's List of referencing Event Condition List references attribute.

Si le paramètre Listes de noms de Listes de Conditions Événementielles a été fourni, pour chaque objet Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles, le VMD doit faire référence à l'objet Liste de Conditions Événementielles nouvellement créé dans l'attribut Liste des références de Listes de Conditions Événementielles de référence de l'objet Liste de Conditions Événementielles référencées.

Finalement, un Résultat(+) doit être émis, indiquant que l'objet Liste de Conditions Événementielles a été créé et que les références ont été mises à jour.

Les valeurs initiales des attributs de l'objet Liste de Conditions Événementielles sont définies ci-après:

- a) Nom de Liste de Conditions Événementielles – Initialisé à la valeur du paramètre nom de Liste de Conditions Événementielles.
- b) Destructible par MMS – Initialisé sur VRAI.
- c) Liste des références de Conditions Événementielles – Initialisée de manière à faire référence aux objets Condition Événementielle spécifiés par la valeur du paramètre Liste de noms de Conditions Événementielles.
- d) Liste des références de Listes de Conditions Événementielles – Initialisée de manière à faire référence aux objets Liste de Conditions Événementielles spécifiés par la valeur du paramètre Liste de noms de Listes de Conditions Événementielles.
- e) Liste des références de Listes de Conditions Événementielles de référence – Initialisée sur une liste vide.

7.3.14.3 *Protocole DefineEventConditionList* (Définir Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection defineECL de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
DefineEventConditionList-Request ::= SEQUENCE {
    eventConditionListName          [0] ObjectName,
    listOfEventConditionName        [1] IMPLICIT SEQUENCE OF Object
                                     Name,
    listOfEventConditionListName    [2] IMPLICIT SEQUENCE OF Object
                                     Name OPTIONAL
    -- shall appear if and only if RECL has been negotiated.
}

DefineEventConditionList-Response ::= NULL
DefineEventConditionList-Error ::= ObjectName
```

If the List of Event Condition List names parameter has been provided, for every Event Condition List object specified in the List of Event Condition List names parameter, the VMD shall place a reference to the newly created Event Condition List object in the referenced Event Condition List object's List of referencing Event Condition List references attribute.

Finally, a Result(+) shall be issued, indicating that the Event Condition List object was created and references updated.

The initial value for the attributes of the Event Condition List object are specified below:

- a) Event Condition List name - Initialized to the value of the Event Condition List name parameter.
- b) MMS Deletable - Initialized to TRUE.
- c) List of Event Condition references - Initialized to refer to the Event Condition objects specified by the value of the List of Event Condition names parameter.
- d) List of Event Condition List references - Initialized to refer to the Event Condition List objects specified by the value of the List of Event Condition List names parameter.
- e) List of referencing Event Condition List references - Initialized to an empty list.

7.3.14.3 *DefineEventConditionList protocol*

The abstract syntax of the defineECL choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
DefineEventConditionList-Request ::= SEQUENCE {
    eventConditionListName      [0] ObjectName,
    listOfEventConditionName    [1] IMPLICIT SEQUENCE OF ObjectName,
    listOfEventConditionListName [2] IMPLICIT SEQUENCE OF ObjectName
                                OPTIONAL
    -- shall appear if and only if RECL has been negotiated.
}

DefineEventConditionList-Response ::= NULL
DefineEventConditionList-Error ::= ObjectName
```

7.3.14.3.1 *DefineEventConditionList-Request* (Demande-Définir Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection defineECL du type ConfirmedServiceRequest doit être DefineEventConditionList-Request.

7.3.14.3.2 *DefineEventConditionList-Response* (Réponse-Définir Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection defineECL du type ConfirmedServiceResponse doit être DefineEventConditionList-Response.

7.3.14.3.3 *DefineEventConditionList-Error* (Erreur-Définir Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection defineECL de AdditionalService-Error doit être DefineEventConditionList-Error, qui doit être le paramètre Objet erroné du paramètre Résultat(-) de la primitive DefineEventConditionList-Response, et doit apparaître comme étant le paramètre Objet erroné de la primitive DefineEventConditionList.confirm, s'il est émis.

7.3.15 *Service DeleteEventConditionList* (Détruire Liste de Conditions Événementielles)

Le service DeleteEventConditionList doit être utilisé pour obtenir la destruction d'un objet Liste de Conditions Événementielles au niveau du VMD.

7.3.15.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 23.

Tableau 23 – Service DeleteEventConditionList

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de liste de Conditions Événementielles	M	M(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.15.1.1 *Argument*

Ce paramètre doit véhiculer les paramètres de la demande de service DeleteEventConditionList.

7.3.15.1.1.1 *Nom de Liste de Conditions Événementielles*

Ce paramètre, du type Nom d'Objet, doit spécifier le nom de l'objet Liste de Conditions Événementielles à détruire au niveau du VMD.

7.3.14.3.1 *DefineEventConditionList-Request*

The abstract syntax of the defineECL choice of the ConfirmedServiceRequest type shall be the DefineEventConditionList-Request.

7.3.14.3.2 *DefineEventConditionList-Response*

The abstract syntax of the defineECL choice of the ConfirmedServiceResponse type shall be the DefineEventConditionList-Response.

7.3.14.3.3 *DefineEventConditionList-Error*

The abstract syntax of the defineECL choice of the AdditionalService-Error shall be the DefineEventConditionList-Error, which shall be the Object in error parameter of the Result(-) parameter of the DefineEventConditionList-Response primitive, and shall appear as the Object in error parameter of the DefineEventConditionList.confirm primitive, if issued.

7.3.15 *DeleteEventConditionList service*

The DeleteEventConditionList service shall be used to cause the deletion of an Event Condition List object at the VMD.

7.3.15.1 *Structure*

The structure of the component service primitives is shown in table 23.

Table 23 – DeleteEventConditionList service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Event Condition List name	M	M(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.15.1.1 *Argument*

This parameter shall convey the parameters of the DeleteEventConditionList service request.

7.3.15.1.1.1 *Event Condition List name*

This parameter, of type Object Name, shall specify the name of the Event Condition List object that is to be deleted at the VMD.

7.3.15.1.2 *Résultat(+)*

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Dans ce cas, le résultat ne doit renvoyer aucun paramètre spécifique au service.

7.3.15.1.3 *Résultat(-)*

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec.

7.3.15.2 *Procédure de service*

Si la demande est acceptable, le VMD doit s'assurer que la valeur de l'attribut Liste des références de Listes de Conditions Événementielles de référence de l'objet Liste de Conditions Événementielles spécifié est égale à une liste vide. Si la valeur de cet attribut n'est pas égale à une liste vide, la demande de service doit échouer et un Résultat(-) doit être renvoyé.

Le VMD doit détruire la référence à l'objet Liste de Conditions Événementielles spécifié, contenue dans l'attribut Liste des références de Listes de Conditions Événementielles de référence, de tous les autres objets Liste de Conditions Événementielles identifiés par la valeur de l'attribut Liste des références de Listes de Conditions Événementielles.

Le VMD doit détruire les références à l'objet Liste de Conditions Événementielles spécifié, contenu dans l'attribut Liste des références de Listes de Conditions Événementielles de référence, à partir des objets Condition Événementielle spécifiés et référencés par la valeur de l'attribut Liste des références de Conditions Événementielles de la Liste de Conditions Événementielles.

Le VMD doit annuler l'objet Liste de Conditions Événementielles spécifié, et un Résultat(+) doit être renvoyé.

7.3.15.3 *Protocole DeleteEventConditionList*

La syntaxe abstraite de la sélection deleteECL de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
DeleteEventConditionList-Request ::= ObjectName -- EventConditionListName
DeleteEventConditionList-Response ::= NULL
```

7.3.15.3.1 *DeleteEventConditionList-Request* (Demande-Détruire Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection deleteECL du type Confirmed Service Request doit être DeleteEventConditionList-Request.

7.3.15.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service request succeeded. A successful result shall return no service-specific parameters.

7.3.15.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.15.2 *Service procedure*

If the request is acceptable, the VMD shall ensure that the value of the List of referencing Event Condition List references attribute of the specified Event Condition List object is equal to an empty list. If the value of this attribute is not equal to an empty list, the service request shall fail and a Result(-) shall be returned.

The VMD shall delete the reference to the specified Event Condition List object, contained in the List of referencing Event Condition List references attribute, from all other Event Condition List objects identified by the value of the List of Event Condition List references attribute.

The VMD shall delete references to the specified Event Condition List object, contained in the List of referencing Event Condition List references attribute, from Event Condition objects specified in and referenced by the value of the specified Event Condition List's List of Event Condition references attribute.

The VMD shall delete the specified Event Condition List object, and a Result(+) shall be returned.

7.3.15.3 *DeleteEventConditionList protocol*

The abstract syntax of the deleteECL choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
DeleteEventConditionList-Request ::= ObjectName -- EventConditionListName
DeleteEventConditionList-Response ::= NULL
```

7.3.15.3.1 *DeleteEventConditionList-Request*

The abstract syntax of the deleteECL choice of the ConfirmedServiceRequest type shall be the DeleteEventConditionList-Request.

7.3.15.3.2 *DeleteEventConditionList-Response* (Réponse-Détruire Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection deleteECL du type ConfirmedServiceResponse doit être DeleteEventConditionList-Response.

7.3.16 *Service AddEventConditionListReference* (Ajouter Référence de Liste de Conditions Événementielles)

Le service AddEventConditionListReference doit être utilisé pour l'ajout de références d'objet Condition Événementielle ou de références d'objet Liste de Conditions Événementielles, ou l'ajout des deux à la fois, à un objet Liste de Conditions Événementielles au niveau du VMD.

7.3.16.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 24.

Tableau 24 – Service AddEventConditionListReference

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Liste de Conditions Événementielles	M	M(=)			
Liste de noms de Conditions Événementielles	M	M(=)			
Liste de noms de Listes de Conditions Événementielles	C	C(=)			RECL
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Objet erroné			C	C(=)	

7.3.16.1.1 *Argument*

Ce paramètre doit véhiculer les paramètres de la demande de service AddEventConditionListReference.

7.3.16.1.1.1 *Nom de Liste de Conditions Événementielles*

Ce paramètre, du type Nom d'Objet, doit spécifier le nom de l'objet Liste de Conditions Événementielles à modifier au niveau du VMD.

7.3.16.1.1.2 *Liste de noms de Conditions Événementielles*

Ce paramètre doit contenir une liste des attributs nom d'objet Condition Événementielle qui doit être ajoutée à la Liste de Conditions Événementielles spécifiée. Cette liste ne doit

7.3.15.3.2 DeleteEventConditionList-Response

The abstract syntax of the deleteECL choice of the ConfirmedServiceResponse type shall be the DeleteEventConditionList-Response.

7.3.16 AddEventConditionListReference service

The AddEventConditionListReference service shall be used to cause the addition of Event Condition object references, or Event Condition List object references, or both to an Event Condition List object at the VMD.

7.3.16.1 Structure

The structure of the component service primitives is shown in table 24.

Table 24 – AddEventConditionListReference service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Event Condition List name	M	M(=)			
List of Event Condition names	M	M(=)			
List of Event Condition List names	C	C(=)			RECL
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Object in error			C	C(=)	

7.3.16.1.1 Argument

This parameter shall convey the parameters of the AddEventConditionListReference service request.

7.3.16.1.1.1 Event Condition List name

This parameter, of type Object Name, shall specify the name of the Event Condition List object that is to be modified at the VMD.

7.3.16.1.1.2 List of Event Condition names

This parameter shall contain a list of name attributes of Event Condition objects that shall be added to the specified Event Condition List. This list shall not be empty if RECL has not

pas être vide si RECL n'a pas été pris en compte. S'il n'y a pas de nom de Condition Événementielle dans la demande de service, ce paramètre doit être une liste vide. Si le paramètre nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, ce paramètre ne doit pas contenir un nom de Condition Événementielle de portée AA.

7.3.16.1.1.3 *Liste de noms de Listes de Conditions Événementielles*

Ce paramètre doit contenir une liste des attributs nom d'objet Liste de Conditions Événementielles qui doit être ajoutée à la liste de Conditions Événementielles. Ce paramètre ne doit pas être présent si le CBB RECL n'a pas été pris en compte. Si RECL a été pris en compte, alors qu'il n'y a pas de nom de Liste de Conditions Événementielles dans cette demande de service, ce paramètre doit être une liste vide. Si le paramètre nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, ce paramètre ne doit pas contenir un nom de Condition Événementielle de portée AA.

7.3.16.1.2 *Résultat(+)*

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Dans ce cas, le résultat ne doit renvoyer aucun paramètre spécifique au service.

7.3.16.1.3 *Résultat(-)*

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec. En outre, le paramètre suivant peut apparaître.

7.3.16.1.3.1 *Objet erroné*

Ce paramètre, du type Nom d'Objet, doit être présent lorsque l'erreur concerne la non-existence ou l'incohérence d'attribut d'un objet Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles, ou d'un objet Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles. Il doit fournir le nom de l'objet déterminé comme ayant causé l'erreur au niveau du VMD. Ce paramètre ne doit pas être présent lorsque l'échec du service n'est pas dû à la non-existence ou à l'incohérence d'un objet Condition Événementielle ou d'un objet Liste de Conditions Événementielles.

7.3.16.2 *Procédure de service*

Le VMD doit s'assurer que l'objet Liste de Conditions Événementielles spécifié par le paramètre nom de Liste de Conditions Événementielles existe au niveau du VMD. Si l'objet Liste de Conditions Événementielles spécifié est déterminé comme non existant, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES et Code d'Erreur OBJET-NON-EXISTANT.

S'il est déterminé que l'un des objets Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles n'existe pas ou s'il est déterminé que l'un des objets Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles n'existe pas, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES, Code d'Erreur OBJET-NON-EXISTANT, ainsi que le paramètre Objet erroné.

been negotiated. If there are no Event Condition names in this service request, this parameter shall be an empty list. If the scope of the Event Condition List name parameter is VMD-specific or Domain-specific, this parameter shall not contain an Event Condition name whose scope is AA-specific.

7.3.16.1.1.3 *List of Event Condition List names*

This parameter shall contain a list of name attributes of Event Condition List objects that shall be added to the Event Condition List. This parameter shall not be present if the RECL CBB has not been negotiated. If RECL has been negotiated but there are no Event Condition List names in this service request, this parameter shall be an empty list. If the scope of the Event Condition List name parameter is VMD-specific or Domain-specific, this parameter shall not contain an Event Condition List name whose scope is AA-specific.

7.3.16.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service request succeeded. A successful result shall return no service-specific parameters.

7.3.16.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure. In addition, the following parameter may appear.

7.3.16.1.3.1 *Object in error*

This parameter, of type Object Name, shall be present when the error concerns the non-existence or attribute inconsistency of an Event Condition object specified in the List of Event Condition names parameter, or an Event Condition List object specified in the List of Event Condition List names parameter. It shall provide the name of the object determined which caused the error at the VMD. This parameter shall not be present when the failure of this service is not due to the nonexistence or inconsistency of an Event Condition object or an Event Condition List object.

7.3.16.2 *Service procedure*

The VMD shall determine that the Event Condition List object specified by the Event Condition List name parameter exists at the VMD. If the specified Event Condition List object is determined to not exist, the Result(-) service primitive shall be issued with Error Class ACCESS and Error Code OBJECT-NON-EXISTENT.

If it is determined that one of the Event Condition objects specified in the List of Event Condition names parameter does not exist, or if it is determined that one of the Event Condition List objects specified in the List of Event Condition List names parameter does not exist, the Result(-) service primitive shall be issued with Error Class ACCESS, Error Code OBJECT-NON-EXISTENT, and the Object in error parameter.

Si le paramètre nom de Liste de Conditions Événementielles a une portée VMD ou Domaine, et si le nom de l'une des Conditions Événementielles dans le paramètre Liste de noms de Conditions Événementielles ou si l'un des noms contenus dans le paramètre Liste de noms de Listes de Conditions Événementielles a une portée AA, le VMD doit émettre une primitive de service Résultat(-) avec une Classe d'Erreur DEFINITION, un Code d'Erreur ATTRIBUT-OBJET-INCOHERENT, ainsi que le paramètre Objet erroné.

Si la demande est acceptable, et si le paramètre Liste de noms de Conditions Événementielles a été fourni, pour chaque objet Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles, le VMD doit:

- a) vérifier que l'objet Condition Événementielle n'est pas déjà référencé dans la Liste des Conditions Événementielles; s'il est référencé, omettre le reste de cette procédure pour cette Condition Événementielle;
- b) faire référence à l'objet Liste de Conditions Événementielles spécifié dans l'attribut Liste des références des Listes de Conditions Événementielles de référence de l'objet Condition Événementielle;
- c) ajouter la référence objet Condition Événementielle identifié à l'attribut Liste des références de Conditions Événementielles de l'objet Liste de Conditions Événementielles.

Si la demande est acceptable et si le paramètre Liste de noms de Listes de Conditions Événementielles a été fourni, pour chaque objet Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles, le VMD doit:

- a) vérifier que l'objet Liste de Conditions Événementielles n'est pas déjà référencé dans la Liste de Conditions Événementielles; s'il est déjà référencé, omettre le reste de cette procédure pour cette Liste de Conditions Événementielles;
- b) faire référence à l'objet Liste de Conditions Événementielles spécifié dans l'attribut Liste des références de Listes de Conditions Événementielles de référence de l'objet Liste de Conditions Événementielles;
- c) ajouter la référence à l'objet Liste de Conditions Événementielles identifié à l'attribut Liste des références de Listes de Conditions Événementielles de l'objet Liste de Conditions Événementielles.

Finalement, un Résultat(+) doit être émis, indiquant que l'objet Liste de Conditions Événementielles a été modifié et que les références ont été mises à jour.

7.3.16.3 Protocole AddEventConditionListReference

La syntaxe abstraite de la sélection addECLReference de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

If the scope of the Event Condition List name parameter is VMD-specific or Domain-specific, and the scope of the name of any of the Event Condition in the List of Event Condition names parameter or of any of the List of Event Condition List names parameter is AA-specific, the VMD shall issue a Result(-) service primitive with an Error Class of DEFINITION, Error Code of OBJECT-ATTRIBUTE-INCONSISTENT, and the Object in error parameter.

If the request is acceptable, and if the List of Event Condition names parameter has been provided, for every Event Condition object specified in the List of Event Condition names parameter, the VMD shall:

- a) verify that the Event Condition object is not already referenced in this Event Condition List; if it is referenced, skip the remainder of this step for this Event Condition.
- b) place a reference to the specified Event Condition List object in the Event Condition object's List of referencing Event Condition List references attribute.
- c) add the identified Event Condition object reference to the specified Event Condition List object's List of Event Condition references attribute.

If the request is acceptable, and if the List of Event Condition List names parameter has been provided, for every Event Condition List object specified in the List of Event Condition List names parameter, the VMD shall:

- a) verify that the Event Condition List object is not already referenced in this Event Condition List; if it is referenced, skip the remainder of this step for this Event Condition List.
- b) place a reference to the specified Event Condition List object in the referenced Event Condition List object's List of referencing Event Condition List references attribute.
- c) add the identified Event Condition List object reference to the specified Event Condition List object's List of Event Condition List references attribute.

Finally, a Result(+) shall be issued, indicating that the Event Condition List object was modified and references updated.

7.3.16.3 *AddEventConditionListReference protocol*

The abstract syntax of the addECLReference choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```

AddEventConditionListReference-Request ::= SEQUENCE {
    eventConditionListName      [0] ObjectName,
    listOfEventConditionName    [1] IMPLICIT SEQUENCE OF Object Name,
    listOfEventConditionListName [2] IMPLICIT SEQUENCE OF Object Name
                                OPTIONAL
    -- shall appear if and only if RECL has been negotiated
}

AddEventConditionListReference-Response ::= NULL
AddEventConditionListReference-Error ::= ObjectName
    
```

7.3.16.3.1 *AddEventConditionListReference-Request* (Demande-Ajouter Référence de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection addECLReference du type ConfirmedServiceRequest doit être AddEventConditionListReference-Request.

7.3.16.3.2 *AddEventConditionListReference-Response* (Réponse-Ajouter Référence de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection addECLReference du type ConfirmedServiceResponse doit être AddEventConditionListReference-Response.

7.3.16.3.3 *AddEventConditionListReference-Error* (Erreur-Ajouter Référence de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection addECLReference de AdditionalService-Error doit être AddEventConditionListReference-Error, qui doit être le paramètre Objet erroné du paramètre Résultat(-) de la primitive AddEventConditionListReference-Response, et doit apparaître comme étant le paramètre Objet erroné de la primitive AddEventConditionListReference.confirm, s'il est émis.

7.3.17 *Service RemoveEventConditionListReference* (Retirer-Référence de Liste de Conditions Événementielles)

Le service RemoveEventConditionListReference doit être utilisé pour retirer des références d'objet Condition Événementielle, ou pour retirer des références de Liste de Conditions Événementielles, ou les deux à la fois, d'un objet Liste de Conditions Événementielles spécifié au niveau du VMD.

7.3.17.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 25.

```

AddEventConditionListReference-Request ::= SEQUENCE {
    eventConditionListName      [0] ObjectName,
    listOfEventConditionName    [1] IMPLICIT SEQUENCE OF ObjectName,
    listOfEventConditionListNam [2] IMPLICIT SEQUENCE OF ObjectName
                                OPTIONAL
    -- shall appear if and only if RECL has been negotiated
}

AddEventConditionListReference-Response ::= NULL
AddEventConditionListReference-Error ::= ObjectName

```

7.3.16.3.1 *AddEventConditionListReference-Request*

The abstract syntax of the addECLReference choice of the ConfirmedServiceRequest type shall be the AddEventConditionListReference-Request.

7.3.16.3.2 *AddEventConditionListReference-Response*

The abstract syntax of the addECLReference choice of the ConfirmedServiceResponse type shall be the AddEventConditionListReference-Response.

7.3.16.3.3 *AddEventConditionListReference-Error*

The abstract syntax of the addECLReference choice of the AdditionalService-Error shall be the AddEventConditionListReference-Error, which shall be the Object in error parameter of the Result(-) parameter of the AddEventConditionListReference-Response primitive, and shall appear as the Object in error parameter of the AddEventConditionListReference.confirm primitive, if issued.

7.3.17 *RemoveEventConditionListReference service*

The RemoveEventConditionListReference service shall be used to cause the removal of Event Condition object references, or removal of Event Condition List references, or both, from a specified Event Condition List object at the VMD.

7.3.17.1 *Structure*

The structure of the component service primitives is shown in table 25.

Tableau 25 – Service RemoveEventConditionListReference

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Liste de Conditions Événementielles	M	M(=)			
Liste de noms de Conditions Événementielles	M	M(=)			
Liste de noms de Listes de Conditions Événementielles	C	C(=)			RECL
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	
Objet erroné			C	C(=)	

7.3.17.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service RemoveEventConditionListReference.

7.3.17.1.1.1 Nom de Liste de Conditions Événementielles

Ce paramètre, du type Nom d'Objet, doit spécifier le nom de l'objet Liste de Conditions Événementielles à modifier au niveau du VMD.

7.3.17.1.1.2 Liste de noms de Conditions Événementielles

Ce paramètre doit contenir une liste des attributs nom d'objets Condition Événementielle qui doivent être retirés de la Liste de Conditions Événementielles spécifiée. Cette liste ne doit pas être vide si RECL n'a pas été pris en compte. S'il n'y a pas de noms de Conditions Événementielles dans cette demande de service, ce paramètre doit être une liste vide.

7.3.17.1.1.3 Liste de noms de Listes de Conditions Événementielles

Ce paramètre doit contenir une liste des attributs nom des objets Liste de Conditions Événementielles qui doivent être retirés de la Liste de Conditions Événementielles. Ce paramètre ne doit pas être présent si le CBB RECL n'a pas été pris en compte. Si RECL a été pris en compte mais qu'il n'y a pas de noms de Listes de Conditions Événementielles dans la demande de service, ce paramètre doit être une liste vide.

7.3.17.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Dans ce cas, le résultat ne doit renvoyer aucun paramètre spécifique au service.

7.3.17.1.3 Résultat(-)

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec. Il est en outre admis que le paramètre suivant apparaisse.

Table 25 – RemoveEventConditionListReference service

Parameter name	Req	Ind	Resp	Cnf	CBB
Argument	M	M(=)			
Event Condition List name	M	M(=)			
List of Event Condition names	M	M(=)			
List of Event Condition List names	C	C(=)			RECL
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	
Object in error			C	C(=)	

7.3.17.1.1 Argument

This parameter shall convey the parameters of the RemoveEventConditionListReference service request.

7.3.17.1.1.1 Event Condition List name

This parameter, of type Object Name, shall specify the name of the Event Condition List object that is to be modified at the VMD.

7.3.17.1.1.2 List of Event Condition names

This parameter shall contain a list of name attributes of Event Condition objects that shall be removed from the specified Event Condition List. This list shall not be empty if RECL has not been negotiated. If there are no Event Condition names in this service request, this parameter shall be an empty list.

7.3.17.1.1.3 List of Event Condition List names

This parameter shall contain a list of name attributes of Event Condition List objects that shall be removed from the Event Condition List. This parameter shall not be present if the RECL CBB has not been negotiated. If RECL has been negotiated but there are no Event Condition List names in this service request, this parameter shall be an empty list.

7.3.17.1.2 Result(+)

The Result(+) parameter shall indicate that the service request succeeded. A successful result shall return no service-specific parameters.

7.3.17.1.3 Result(-)

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in clause 17 of ISO/IEC 9506-1, shall provide the reason for failure. In addition, the following parameter may appear.

7.3.17.1.3.1 *Objet erroné*

Ce paramètre, du type Nom d'Objet, doit indiquer l'objet Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles, ou l'objet Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles, qui a entraîné l'erreur. Cette erreur indique soit (1) que l'objet spécifié n'existe pas, soit 2) qu'il n'est pas référencé par l'attribut Liste de références de Conditions Événementielles ou de Liste de références de Listes de Conditions Événementielles de l'objet Liste de Conditions Événementielles spécifié par le paramètre Nom de Liste de Conditions Événementielles.

7.3.17.2 *Procédure de service*

Le VMD doit déterminer que l'objet Liste de Conditions Événementielles spécifié par le paramètre nom de Liste de Conditions Événementielles existe au niveau du VMD. S'il est déterminé que l'objet Liste de Conditions Événementielles n'existe pas, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES et Code d'Erreur OBJET-NON-EXISTANT.

S'il est déterminé que l'un des objets Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles n'existe pas, ou s'il est déterminé que l'un des objets Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles n'existe pas, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES, Code d'Erreur OBJET-NON-EXISTANT, ainsi que le paramètre Objet erroné.

S'il est déterminé que l'un des objets Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles n'est pas référencé par l'attribut Liste des références de Conditions Événementielles de l'objet Liste de Conditions Événementielles ou s'il est déterminé que l'un des objets Liste de Conditions Événementielles spécifié dans le paramètre Liste de noms de Listes de Conditions Événementielles n'est pas référencé par l'attribut Liste des références de Listes de Conditions Événementielles de l'objet Liste de Conditions Événementielles, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES, un Code d'Erreur OBJET-NON EXISTANT, ainsi que le paramètre Objet erroné.

Si la demande est acceptable, et si le paramètre Liste de noms de Conditions Événementielles a été fourni, pour chaque objet Condition Événementielle spécifié dans le paramètre Liste de noms de Conditions Événementielles, le VMD doit retirer la référence à l'objet Liste de Conditions Événementielles spécifié dans l'attribut Liste des références de Listes de Conditions Événementielles de référence de l'objet Condition Événementielle. Le VMD doit retirer la référence à chaque objet Condition Événementielle identifié de l'attribut Liste des références de Conditions Événementielles de l'objet Liste de Conditions Événementielles spécifié.

Si le paramètre Liste de noms de Listes de Conditions Événementielles a été fourni, pour chaque objet Liste de Conditions Événementielles spécifié dans le paramètre Liste de Noms de Listes de Conditions Événementielles, le VMD doit retirer la référence à l'objet Liste de Conditions Événementielles spécifié dans l'attribut Liste des références de Liste de Conditions Événementielles de référence de l'objet Liste de Conditions Événementielles. Le VMD doit retirer la référence à chaque objet Liste de Conditions Événementielles identifié de l'attribut Liste des références de Liste de Conditions Événementielles de l'objet Liste de Conditions Événementielles spécifié.

7.3.17.1.3.1 *Object in error*

This parameter, of type Object Name, shall indicate the Event Condition object specified in the List of Event Condition names parameter, or the Event Condition List object specified in the List of Event Condition List names parameter, which caused the error. This error indicates that either (1) the specified object does not exist, or (2) that it is not referenced by the List of Event Condition reference or List of Event Condition List reference attribute of the Event Condition List object specified by the Event Condition List Name parameter.

7.3.17.2 *Service procedure*

The VMD shall determine that the Event Condition List object specified by the Event Condition List name parameter exists at the VMD. If the specified Event Condition List object is determined to not exist, the Result(-) service primitive shall be issued with Error Class ACCESS and Error Code OBJECT-NON-EXISTENT.

If it is determined that one of the Event Condition objects specified in the List of Event Condition names parameter does not exist, or if it is determined that one of the Event Condition List objects specified in the List of Event Condition List names parameter does not exist, the Result(-) service primitive shall be issued with Error Class ACCESS, Error Code OBJECT-NON-EXISTENT, and the Object in error parameter.

If it is determined that one of the Event Condition objects specified in the List of Event Condition names parameter is not referenced by the List of Event Condition reference attribute of the Event Condition List object, or if it is determined that one of the Event Condition List objects specified in the List of Event Condition List names parameter is not referenced by the List of Event Condition List reference attribute of the Event Condition List object, the Result(-) service primitive shall be issued with Error Class ACCESS, Error Code OBJECT-NON-EXISTENT, and the Object in error parameter.

If the request is acceptable, and if the List of Event Condition names parameter has been provided, for every Event Condition object specified in the List of Event Condition names parameter, the VMD shall remove the reference to the specified Event Condition List object in the Event Condition object's List of referencing Event Condition List references attribute. The VMD shall remove the reference to each identified Event Condition object from the specified Event Condition List object's List of Event Condition references attribute.

If the List of Event Condition List names parameter has been provided, for every Event Condition List object specified in the List of Event Condition List names parameter, the VMD shall remove the reference to the specified Event Condition List object in the Event Condition List object's List of referencing Event Condition List references attribute. The VMD shall remove the reference to each identified Event Condition List object from the specified Event Condition List object's List of Event Condition List references attribute.

7.3.17.3 *Protocole RemoveEventConditionListReference*

La syntaxe abstraite de la sélection removeECLReference de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
RemoveEventConditionListReference-Request ::= SEQUENCE {
    eventConditionListName      [0] ObjectName,
    listOfEventConditionName    [1] IMPLICIT SEQUENCE OF Object
                                Name,
    listOfEventConditionListName [2] IMPLICIT SEQUENCE OF Object
                                Name OPTIONAL
    -- shall appear if and only if RECL has been negotiated
}

RemoveEventConditionListReference-Response ::= NULL
RemoveEventConditionListReference-Error ::= CHOICE {
    eventCondition      [0] ObjectName,
    eventConditionList [1] ObjectName
}
```

7.3.17.3.1 *RemoveEventConditionListReference-Request* (Demande-Retirer Référence de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection removeECLReference du type ConfirmedServiceRequest doit être RemoveEventConditionListReference-Request.

7.3.17.3.2 *RemoveEventConditionListReference-Response* (Réponse-Retirer Référence de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection removeECLReference du type ConfirmedServiceResponse doit être RemoveEventConditionListReference-Response.

7.3.17.3.3 *RemoveEventConditionListReference-Error* (Erreur-Retirer Référence de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection removeECLReference de AdditionalService-Error doit être RemoveEventConditionListReference-Error, qui doit être le paramètre Objet erroné du paramètre Résultat(-) de la primitive RemoveEventConditionListReference.response, et doit apparaître comme étant le paramètre Objet erroné de la primitive RemoveEventConditionListReference.confirm, s'il est émis.

7.3.18 *Service GetEventConditionListAttributes* (Obtenir Attributs de Liste de Conditions Événementielles)

Le service GetEventConditionListAttributes doit être utilisé pour déterminer les valeurs d'attributs d'un objet Liste de Conditions Événementielles spécifié au niveau du VMD.

7.3.18.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 26.

7.3.17.3 *RemoveEventConditionListReference protocol*

The abstract syntax of the removeECLReference choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of part 2 of ISO 9506 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```

RemoveEventConditionListReference-Request ::= SEQUENCE {
    eventConditionListName      [0] ObjectName,
    listOfEventConditionName    [1] IMPLICIT SEQUENCE OF ObjectName,
    listOfEventConditionListName [2] IMPLICIT SEQUENCE OF ObjectName
                                OPTIONAL
    -- shall appear if and only if RECL has been negotiated
}

RemoveEventConditionListReference-Response ::= NULL
RemoveEventConditionListReference-Error ::= CHOICE {
    eventCondition      [0] ObjectName,
    eventConditionList  [1] ObjectName
}

```

7.3.17.3.1 *RemoveEventConditionListReference-Request*

The abstract syntax of the removeECLReference choice of the ConfirmedServiceRequest type shall be the RemoveEventConditionListReference-Request.

7.3.17.3.2 *RemoveEventConditionListReference-Response*

The abstract syntax of the removeECLReference choice of the ConfirmedServiceResponse type shall be the RemoveEventConditionListReference-Response.

7.3.17.3.3 *RemoveEventConditionListReference-Error*

The abstract syntax of the removeECLReference choice of the AdditionalService-Error shall be the RemoveEventConditionListReference-Error, which shall be the Object in error parameter of the Result(-) parameter of the RemoveEventConditionListReference-Response primitive, and shall appear as the Object in error parameter of the RemoveEventConditionListReference.confirm primitive, if issued.

7.3.18 *GetEventConditionListAttributes service*

The GetEventConditionListAttributes service shall be used to determine the attribute values of a specified Event Condition List object at the VMD.

7.3.18.1 *Structure*

The structure of the component service primitives is shown in table 26.

Tableau 26 – Service GetEventConditionListAttributes

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Liste de Conditions Événementielles	M	M(=)			
Résultat(+)			S	S(=)	
Liste de noms de Conditions Événementielles			M	M(=)	
Liste de noms de Listes de Conditions Événementielles			C	C(=)	RECL
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.18.1.1 Argument

Ce paramètre doit véhiculer les paramètres de la demande de service GetEventConditionListAttributes.

7.3.18.1.1.1 Nom de Liste de Conditions Événementielles

Ce paramètre, du type Nom d'Objet, doit spécifier le nom de l'objet Liste de Conditions Événementielles au niveau du VMD, à partir duquel les valeurs d'attributs doivent être obtenues.

7.3.18.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Dans ce cas, le résultat doit renvoyer des paramètres Liste de noms de Conditions Événementielles et Liste de noms de Listes de Conditions Événementielles.

7.3.18.1.2.1 Liste de noms de Conditions Événementielles

Ce paramètre doit contenir une liste des attributs nom d'objets Condition Événementielle qui ont été extraits de l'attribut Liste des références de Conditions Événementielles de l'objet Liste de Conditions Événementielles spécifié. Cette liste ne doit pas être vide si RECL n'a pas été pris en compte. S'il n'y a pas de noms de Conditions Événementielles dans la demande de service, ce paramètre doit être une liste vide.

7.3.18.1.2.2 Liste de noms de Listes de Conditions Événementielles

Ce paramètre doit contenir une liste des attributs nom d'objets Liste de Conditions Événementielles qui ont été induits de l'attribut Liste des références de Listes de Conditions Événementielles de l'objet Liste de Conditions Événementielles spécifié. Ce paramètre ne doit pas être présent si le CBB RECL n'a pas été pris en compte. Si RECL a été pris en compte mais qu'il n'y a pas de nom de Listes de Conditions Événementielles dans cette demande de service, ce paramètre doit être une liste vide.

Table 26 – GetEventConditionListAttributes service

Parameter name	Req	Ind	Rep	Cnf	CBB
Argument	M	M(=)			
Event Condition List name	M	M(=)			
Result(+)			S	S(=)	
List of Event Condition names			M	M(=)	
List of Event Condition List names			C	C(=)	RECL
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.18.1.1 Argument

This parameter shall convey the parameters of the GetEventConditionListAttributes service request.

7.3.18.1.1.1 Event Condition List name

This parameter, of type Object Name, shall specify the name of the Event Condition List object at the VMD from which the attribute values are to be obtained.

7.3.18.1.2 Result(+)

The Result(+) parameter shall indicate that the service request succeeded. A successful result shall return the List of Event Condition names and the List of Event Condition List names parameters.

7.3.18.1.2.1 List of Event Condition names

This parameter shall contain a list of name attributes of Event Condition objects that have been derived from the List of Event Condition references attribute of the specified Event Condition List object. This list shall not be empty if RECL has not been negotiated. If there are no Event Condition names in this service request, this parameter shall be an empty list.

7.3.18.1.2.2 List of Event Condition List names

This parameter shall contain a list of name attributes of Event Condition List objects that have been derived from the List of Event Condition List references attribute of the specified Event Condition List object. This parameter shall not be present if the RECL CBB has not been negotiated. If RECL has been negotiated but there are no Event Condition List names in this service request, this parameter shall be an empty list.

7.3.18.1.3 *Résultat(-)*

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini dans l'article 17 de l'ISO/IEC 9506-1 doit indiquer la raison de l'échec.

7.3.18.2 *Procédure de service*

Le VMD doit déterminer que l'objet Liste de Conditions Événementielles spécifié par le paramètre nom de Liste de Conditions Événementielles existe au niveau du VMD. S'il est déterminé que l'objet Liste de Conditions Événementielles spécifié n'existe pas, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES et Code d'Erreur OBJET-NON-EXISTANT.

Si la demande est acceptable, le VMD doit obtenir les valeurs des attributs nom des objets Condition Événementielle et des objets Liste de Conditions Événementielles référencés, les mettre respectivement sur les valeurs du paramètre Liste de noms de Conditions Événementielles et du paramètre Liste de noms de Listes de Conditions Événementielles, et renvoyer le Résultat(+).

7.3.18.3 *Protocole GetEventConditionListAttributes*

La syntaxe abstraite de la sélection getECLAttributes de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

GetEventConditionListAttributes-Request::=ObjectName eventConditionList-Name

GetEventConditionListAttributes-Response::= SEQUENCE {
 listOfEventConditionName [1] IMPLICIT SEQUENCE OF ObjectName,
 listOfEventConditionName [2] IMPLICIT SEQUENCE OF ObjectName
 OPTIONAL
 -- shall appear if and only if RECL has been negotiated
 }

7.3.18.3.1 *GetEventConditionListAttributes-Request* *(Demande-Obtenir Attributs de Liste de Conditions Événementielles)*

La syntaxe abstraite de la sélection getECLAttributes du type ConfirmedServiceRequest doit être GetEventConditionListAttributes-Request.

7.3.18.3.2 *Get EventConditionListAttributes-Response* *(Réponse-Obtenir Attributs de Liste de Conditions Événementielles)*

La syntaxe abstraite de la sélection getECLAttributes du type ConfirmedServiceResponse doit être GetEventConditionListAttributes-Response.

7.3.19 *Service ReportEventConditionListStatus* *(Rendre Compte de l'Etat de Liste de Conditions Événementielles)*

Le service ReportEventConditionListStatus est utilisé pour véhiculer l'état d'une Liste de Conditions Événementielles.

7.3.18.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in clause 17 of ISO/IEC 9506-1 shall provide the reason for failure.

7.3.18.2 *Service procedure*

The VMD shall determine that the Event Condition List object specified by the Event Condition List name parameter exists at the VMD. If the specified Event Condition List object is determined to not exist, the Result(-) service primitive shall be issued with Error Class ACCESS and Error Code OBJECT-NON-EXISTENT.

If the request is acceptable, the VMD shall obtain the values of the name attributes of the referenced Event Condition objects and Event Condition List objects, place them in the values of the List of Event Condition names parameter and List of Event Condition List names parameter respectively, and return the Result(+).

7.3.18.3 *GetEventConditionListAttributes protocol*

The abstract syntax of the getECLAttributes choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
GetEventConditionListAttributes-Request ::= ObjectName --
eventConditionListName
GetEventConditionListAttributes-Response ::= SEQUENCE {
  listOfEventConditionName [1] IMPLICIT SEQUENCE OF ObjectName,
  listOfEventConditionListName [2] IMPLICIT SEQUENCE OF ObjectName
  OPTIONAL
  -- shall appear if and only if RECL has been negotiated
}
```

7.3.18.3.1 *GetEventConditionListAttributes-Request*

The abstract syntax of the getECLAttributes choice of the ConfirmedServiceRequest type shall be the GetEventConditionListAttributes-Request.

7.3.18.3.2 *GetEventConditionListAttributes-Response*

The abstract syntax of the getECLAttributes choice of the ConfirmedServiceResponse type shall be the GetEventConditionListAttributes-Response.

7.3.19 *ReportEventConditionListStatus service*

The ReportEventConditionListStatus service is used to convey the status of an Event Condition List.

7.3.19.1 Structure

La structure des primitives qui constituent ce service est définie dans le tableau 27.

Tableau 27 – Service ReportEventConditionListStatus

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Liste de Conditions Événementielles	M	M(=)			
A la Suite de	U	U(=)			
Résultat(+)			S	S(=)	
Liste d'Etat de Conditions Événementielles			M	M(=)	
Etat en Cours			M	M(=)	
Nombre d'Enveloppes Événementielles			M	M(=)	
Démasqué			C	C(=)	
Heure de Dernière Transition sur Actif			C	C(=)	
Heure de Dernière Transition sur Inactif			C	C(=)	
A Suivre			C	C(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.19.1.1 Argument

Ce paramètre doit véhiculer le paramètre de demande de service ReportEventCondition List Status.

7.3.19.1.1.1 Nom de Liste de Conditions Événementielles

Ce paramètre, du type Nom d'Objet, doit contenir le nom de la Liste de Conditions Événementielles dont le compte-rendu d'état est demandé.

7.3.19.1.1.2 A la Suite de

Ce paramètre, du type Nom d'Objet, indique que l'utilisateur MMS requérant demande la liste d'Etat de Conditions Événementielles renvoyée par l'utilisateur MMS appelé en commençant par un objet Condition Événementielle autre que le premier objet dans la liste. L'ordre de classement spécifié par l'ISO 646 doit être utilisé par l'utilisateur MMS appelé pour déterminer l'objet Condition Événementielle dans la liste spécifiée, après lequel il doit commencer.

7.3.19.1.2 Résultat(+)

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Dans ce cas, les paramètres suivants sont renvoyés dans la primitive de réponse.

7.3.19.1 Structure

The structure of the component service primitives is shown in table 27.

Table 27 – ReportEventConditionListStatus service

Parameter name	Req	Ind	Resp	Cnf	CBB
Argument	M	M(=)			
Event Condition List name	M	M(=)			
Continue After	U	U(=)			
Result(+)			S	S(=)	
List of Event Condition Status			M	M(=)	
Current State			M	M(=)	
Number of Event Enrollments			M	M(=)	
Enabled			C	C(=)	
Time of Last Transition To Active			C	C(=)	
Time of Last Transition To Idle			C	C(=)	
More Follows			C	C(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.19.1.1 Argument

This parameter shall convey the parameter of the ReportEventConditionListStatus service request.

7.3.19.1.1.1 Event Condition List name

This parameter, of type Object Name, shall contain the name of the Event Condition List from which the status report is requested.

7.3.19.1.1.2 Continue After

This parameter, of type Object Name, indicates that the requesting MMS-user requests the list of Event Condition Status returned by the responding MMS-user to begin with an Event Condition object other than the first object in the list. The collating sequence specified by ISO 646 shall be used by the responding MMS-user to determine the Event Condition object in the specified list to start after.

7.3.19.1.2 Result(+)

The Result(+) parameter shall indicate that the service request succeeded. When success is indicated the following parameters are returned in the response primitive.

7.3.19.1.2.1 *Liste d'Etat de Conditions Evénementielles*

Ce paramètre, du type Nom d'Objet, doit contenir zéro ou plusieurs entrées décrivant l'état des objets Condition Evénementielle hiérarchiquement liés à l'objet Liste de Conditions Evénementielles.

7.3.19.1.2.1.1 *Etat en Cours*

Ce paramètre, du type EC-enCours doit contenir la valeur de l'attribut Etat de l'objet Condition Evénementielle.

7.3.19.1.2.1.2 *Nombre d'Enveloppes Evénementielles*

Ce paramètre, du type entier, doit contenir le décompte du nombre d'entrées dans l'attribut Liste des références d'Enveloppes Evénementielles de l'objet Condition Evénementielle.

7.3.19.1.2.1.3 *Démasqué*

Ce paramètre, du type booléen, doit véhiculer le contenu de l'attribut Démasqué de l'objet Condition Evénementielle, pour un objet Condition Evénementielle surveillé. Si l'attribut Classe de Conditions Evénementielles contient la valeur DECLENCHEE PAR LE RESEAU, ce paramètre doit être omis.

7.3.19.1.2.1.4 *Heure de Dernière Transition sur Actif*

Si l'objet Condition Evénementielle est surveillé et a une valeur d'attribut Heure de Dernière Transition sur Actif qui n'est pas égale à INDEFINIE, ce paramètre doit contenir la valeur courante de l'attribut Heure de Dernière Transition sur Actif. Autrement, ce paramètre doit être omis.

7.3.19.1.2.1.5 *Heure de Dernière Transition sur Inactif*

Si l'objet Condition Evénementielle est surveillé et a une valeur d'attribut Heure de Dernière Transition sur Inactif qui n'est pas égale à INDEFINIE, ce paramètre doit contenir la valeur courante de l'attribut Heure de Dernière Transition sur Inactif. Autrement, ce paramètre doit être omis.

7.3.19.1.2.2 *A Suivre*

Ce paramètre, du type booléen, doit indiquer si des demandes ReportEventConditionList-Status supplémentaires sont nécessaires pour récupérer d'autres informations demandées. S'il est VRAI, d'autres demandes sont nécessaires (si l'utilisateur MMS requérant désire récupérer plus de données). S'il est FAUX, soit la liste d'Etat de Conditions Evénementielles contient le dernier état dans la liste soit elle est vide. Le paramètre A Suivre doit être FAUX lorsque le paramètre Liste d'Etat de Conditions Evénementielles est vide.

7.3.19.1.3 *Résultat(-)*

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini plus en détail dans l'article 17 de l'ISO/CEI 9506-1 doit indiquer la raison de l'échec.

7.3.19.1.2.1 *List of Event Condition Status*

This parameter, of type Object Name, shall contain zero or more entries describing the status of Event Condition objects hierarchically related to the Event Condition List object.

7.3.19.1.2.1.1 *Current State*

This parameter, of type EC-State, shall contain the value of the Event Condition object's State attribute.

7.3.19.1.2.1.2 *Number of Event Enrollments*

This parameter, of type integer, shall contain the count of the number of entries in the Event Condition object's List of Event Enrollment references attribute.

7.3.19.1.2.1.3 *Enabled*

This parameter, of type boolean, shall convey the contents of the Enabled attribute of the Event Condition object, for a monitored Event Condition object. If the Event Condition Class attribute contains the value NETWORK-TRIGGERED, this parameter shall be omitted.

7.3.19.1.2.1.4 *Time Of Last Transition To Active*

If the Event Condition object is monitored and has a Time Of Last Transition To Active attribute value not equal to UNDEFINED, this parameter shall contain the current value of the Time Of Last Transition To Active attribute. Otherwise, this parameter shall be omitted.

7.3.19.1.2.1.5 *Time Of Last Transition To Idle*

If the Event Condition object is monitored and has a Time Of Last Transition To Idle attribute with value not equal to UNDEFINED, this parameter shall contain the current value of the Time of Last Transition To Idle attribute. Otherwise, this parameter shall be omitted.

7.3.19.1.2.2 *More Follows*

This parameter, of type boolean, shall indicate whether additional ReportEventConditionList-Status requests are necessary to retrieve more of the requested information. If TRUE, more requests are necessary (if the requesting MMS-user wishes to retrieve more data). If FALSE, then either the List of Event Condition Status contains the last status in the list, or the List of Event Condition Status is empty. The More Follows parameter shall be FALSE when the List of Event Condition Status parameter is empty.

7.3.19.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in detail in clause 17 of ISO/IEC 9506-1, shall provide the reason for failure.

7.3.19.1.4 Procédure de service

Le serveur MMS doit construire une liste d'objets Condition Événementielle, soit en faisant directement référence à l'objet Liste de Conditions Événementielles nommée, soit (si RECL a été pris en compte) en faisant indirectement référence aux objets Liste de Conditions Événementielles référencée par l'objet Liste de Conditions Événementielles nommée. Le serveur MMS doit ordonner la liste selon l'ordre de classement de l'ISO 646, sur la base des attributs nom des objets Condition Événementielle. Il doit renvoyer les informations d'état pour autant d'objets Condition Événementielle qu'il peut en traiter en une seule réponse, en commençant soit au début de la liste si le paramètre A la Suite de n'a pas été spécifié, soit en commençant au niveau ou immédiatement après l'objet Condition Événementielle spécifié par le paramètre A la Suite de.

7.3.19.1.5 Protocole ReportEventConditionListStatus

La syntaxe abstraite de la sélection reportECLStatus de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

```
ReportEventConditionListStatus-Request ::= SEQUENCE {
    eventConditionListName    [0] ObjectName -- Event Condition List
                                Name,
    continueAfter             [1] IMPLICIT Identifier OPTIONAL
}

ReportEventConditionListStatus-Response ::= SEQUENCE {
    listOfEventConditionStatus [1] IMPLICIT SEQUENCE OF EventCondi-
                                tionStatus,
    moreFollows               [2] IMPLICIT BOOLEAN DEFAULT TRUE
}

EventConditionStatus ::= SEQUENCE {
    eventConditionName        [0] ObjectName,
    currentState              [1] IMPLICIT EC-State,
    numberOfEventEnrollments  [2] IMPLICIT Unsigned32,
    enabled                   [3] IMPLICIT BOOLEAN OPTIONAL,
    timeOfLastTransitionToActive [4] EventTime OPTIONAL,
    timeOfLastTransitionToIdle  [5] EventTime OPTIONAL
}
```

7.3.19.1.5.1 ReportEventConditionListStatus-Request (Demande-Rendre Compte de l'Etat de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection reportECLStatus du type ConfirmedServiceRequest doit être ReportEventConditionListStatus-Request.

7.3.19.1.5.2 ReportEventConditionListStatus-Response (Réponse-Rendre Compte de l'Etat de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection reportECLStatus du type ConfirmedServiceResponse doit être ReportEventConditionListStatus-Response.

7.3.19.1.4 *Service procedure*

The MMS server shall construct a list of Event Condition objects, either directly referenced by the named Event Condition List object, or (if RECL has been negotiated) through indirect references of Event Condition List objects referenced by the named Event Condition List object. The MMS server shall order this list using the collating sequence of ISO 646 on the name attributes of the Event Condition objects. It shall return the status information for as many of these Event Condition objects as it can accommodate in a single response, beginning either at the beginning of the list if the Continue After parameter has not been specified, or beginning at or immediately following the Event Condition object specified by the Continue After parameter.

7.3.19.1.5 *ReportEventConditionListStatus protocol*

The abstract syntax of the reportECLStatus choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```
ReportEventConditionListStatus-Request ::= SEQUENCE {
    eventConditionListName [0] ObjectName -- Event Condition List Name,
    continueAfter          [1] IMPLICIT Identifier OPTIONAL
}

ReportEventConditionListStatus-Response ::= SEQUENCE {
    listOfEventConditionStatus [1] IMPLICIT SEQUENCE OF EventCondition
    Status,
    moreFollows                [2] IMPLICIT BOOLEAN DEFAULT TRUE
}

EventConditionStatus ::= SEQUENCE {
    eventConditionName [0] ObjectName,
    currentState       [1] IMPLICIT EC-State,
    numberOfEventEnrollments [2] IMPLICIT Unsigned32,
    enabled             [3] IMPLICIT BOOLEAN OPTIONAL,
    timeOfLastTransitionToActive [4] EventTime OPTIONAL,
    timeOfLastTransitionToIdle  [5] EventTime OPTIONAL
}
```

7.3.19.1.5.1 *ReportEventConditionListStatus-Request*

The abstract syntax of the reportECLStatus choice of the ConfirmedServiceRequest type shall be the ReportEventConditionListStatus-Request.

7.3.19.1.5.2 *ReportEventConditionListStatus-Response*

The abstract syntax of the reportECLStatus choice of the ConfirmedServiceResponse type shall be the ReportEventConditionListStatus-Response.

7.3.20 *Service AlterEventConditionListMonitoring* (Modifier Surveillance de Liste de Conditions Événementielles)

Le service *AlterEventConditionListMonitoring* doit être utilisé pour modifier la valeur des attributs des objets Condition Événementielle référencés par un objet Liste de Conditions Événementielles au niveau du VMD. Les attributs susceptibles d'être modifiés par ce service comprennent l'attribut Démasqué et l'attribut Intervention sur Priorité de Groupe.

7.3.20.1 *Structure*

La structure des primitives qui constituent ce service est définie dans le tableau 28.

Tableau 28 – Service *AlterEventConditionListMonitoring*

Nom du paramètre	Dem	Ind	Rép	Cnf	CBB
Argument	M	M(=)			
Nom de Liste de Conditions Événementielles	M	M(=)			
Démasqué	U	U(=)			
Modification de Priorité	U	U(=)			
Valeur de Priorité	S	S(=)			
Réinitialisation de Priorité	S	S(=)			
Résultat(+)			S	S(=)	
Résultat(-)			S	S(=)	
Type d'Erreur			M	M(=)	

7.3.20.1.1 *Argument*

Ce paramètre doit véhiculer les paramètres de la demande de service *AlterEventConditionListMonitoring*.

7.3.20.1.1.1 *Nom de Liste de Conditions Événementielles*

Ce paramètre, du type Nom d'Objet, doit spécifier le nom de l'objet Liste de Conditions Événementielles au niveau du VMD dont les valeurs d'attribut doivent être modifiées.

7.3.20.1.1.2 *Démasqué*

Ce paramètre facultatif, du type booléen, doit être la valeur de remplacement du contenu de l'attribut Démasqué de tous les objets Condition Événementielle directement ou indirectement référencés par l'objet Liste de Conditions Événementielles. Ce paramètre, ou le paramètre de modification de Priorité, ou les deux à la fois, doivent être fournis.

7.3.20.1.1.3 *Modification de Priorité*

Ce paramètre facultatif, doit modifier l'attribut Intervention sur Priorité de Groupe des objets Condition Événementielle référencés. Ce paramètre, ou le paramètre Démasqué, ou les deux doivent être fournis. Si ce paramètre est fourni, le paramètre valeur de Priorité ou le paramètre rétablissement de Priorité doit être fourni.

7.3.20 *AlterEventConditionListMonitoring service*

The *AlterEventConditionListMonitoring* service shall be used to alter the value of attributes of Event Condition objects referenced by a Event Condition List object at the VMD. Attributes subject to modification by this service include the Enabled attribute and the Group Priority Override attribute.

7.3.20.1 *Structure*

The structure of the component service primitives is shown in table 28.

Table 28 – *AlterEventConditionListMonitoring* service

Parameter name	Req	Ind	Rep	Cnf	CRR
Argument	M	M(=)			
Event Condition List name	M	M(=)			
Enabled	U	U(=)			
Priority change	U	U(=)			
Priority value	S	S(=)			
Priority reset	S	S(=)			
Result(+)			S	S(=)	
Result(-)			S	S(=)	
Error Type			M	M(=)	

7.3.20.1.1 *Argument*

This parameter shall convey the parameters of the *AlterEventConditionListMonitoring* service request.

7.3.20.1.1.1 *Event Condition List name*

This parameter, of type Object Name, shall specify the name of the Event Condition List object at the VMD for which attribute values are to be altered.

7.3.20.1.1.2 *Enabled*

This optional parameter, of type boolean, shall be the replacement value for the contents of the Enabled attribute of all Event Condition objects directly or indirectly referenced by the Event Condition List object. Either this parameter, or the Priority change parameter, or both, shall be provided.

7.3.20.1.1.3 *Priority change*

This optional parameter shall alter the Group Priority Override attribute of the referenced Event Condition objects. Either this parameter, or the Enabled parameter, or both shall be provided. If this parameter is provided, either the Priority value parameter or the Priority restore parameter shall be provided.

7.3.20.1.1.3.1 *Valeur de Priorité*

Ce paramètre, du type entier, doit constituer une valeur de remplacement pour la valeur de l'attribut Intervention sur Priorité de Groupe de tous les objets Condition Événementielle référencés.

7.3.20.1.1.3.2 *Réinitialisation de Priorité*

Ce paramètre, du type à valeur NULLE, doit indiquer que la valeur de l'attribut Intervention sur Priorité de Groupe doit être réinitialisée sur INDEFINIE.

7.3.20.1.2 *Résultat(+)*

Le paramètre Résultat(+) doit indiquer que la demande de service a réussi. Dans ce cas, le résultat ne doit renvoyer aucun paramètre spécifique au service.

7.3.20.1.3 *Résultat(-)*

Le paramètre Résultat(-) doit indiquer que la demande de service a échoué. Le paramètre Type d'Erreur, qui est défini dans l'article 17 de l'ISO/IEC 9506-1, doit indiquer la raison de l'échec.

7.3.20.2 *Procédure de service*

Le VMD doit déterminer que l'objet Liste de Conditions Événementielles spécifié par le paramètre nom de Liste de Conditions Événementielles existe au niveau du VMD. S'il est déterminé que l'objet Liste de Conditions Événementielles spécifié n'existe pas, la primitive de service Résultat(-) doit être émise avec Classe d'Erreur ACCES et Code d'Erreur OBJET-NON-EXISTANT.

Si la demande est acceptable, pour chaque objet Condition Événementielle référencé par la valeur de l'attribut Liste des références de Conditions Événementielles de la Liste de Conditions Événementielles et pour chaque objet Condition Événementielle indirectement référencé par la valeur de l'attribut Liste des références de Listes de Conditions Événementielles, le VMD doit:

- a) Si le paramètre Démasqué a été fourni, établir la valeur de l'attribut Démasqué comme étant égale à la valeur du paramètre Démasqué.
- b) Si le paramètre modification de Priorité a été fourni, le VMD doit:
 - 1) Si la valeur de Priorité a été fournie, remplacer la valeur de l'attribut Intervention sur Priorité de Groupe par la valeur fournie dans le paramètre Valeur de Priorité.
 - 2) Si le paramètre réinitialisation de Priorité a été fourni, modifier la valeur de l'attribut Intervention sur Priorité de Groupe sur INDEFINIE.
- c) Finalement, le Résultat(+) doit être renvoyé.

7.3.20.3 *Protocole AlterEventConditionListMonitoring*

La syntaxe abstraite de la sélection alterECLMonitoring de ConfirmedServiceRequest et ConfirmedServiceResponse est définie ci-après et décrite dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent article.

7.3.20.1.1.3.1 *Priority value*

This parameter, of type integer, shall constitute a replacement value for the value of the Group Priority Override attribute of all referenced Event Condition objects.

7.3.20.1.1.3.2 *Priority reset*

This parameter, of type NULL, shall indicate that the value of the Group Priority Override attribute shall be reset to UNDEFINED.

7.3.20.1.2 *Result(+)*

The Result(+) parameter shall indicate that the service request succeeded. A successful result shall return no service-specific parameters.

7.3.20.1.3 *Result(-)*

The Result(-) parameter shall indicate that the service request failed. The Error Type parameter, which is defined in clause 17 of ISO/IEC 9506-1, shall provide the reason for failure.

7.3.20.2 *Service procedure*

The VMD shall determine that the Event Condition List object specified by the Event Condition List name parameter exists at the VMD. If the specified Event Condition List object is determined to not exist, the Result(-) service primitive shall be issued with Error Class ACCESS and Error Code OBJECT-NON-EXISTENT.

If the request is acceptable, for every Event Condition object referenced by the value of the specified Event Condition List object's List of Event Condition references attribute, and for every Event Condition object indirectly referenced through the value of the List of Event Condition List references attribute, the VMD shall:

- a) If the Enabled parameter has been provided, the value of the Enabled attribute shall be set equal to the value of the Enabled parameter.
- b) If the Priority change parameter has been provided the VMD shall:
 - 1) if the Priority value has been provided, replace the value of the Group Priority Override attribute with the value provided in the Priority Value parameter;
 - 2) if the Priority reset parameter has been provided, change the value of the Group Priority Override attribute to UNDEFINED.
- c) Finally, the Result(+) shall be returned.

7.3.20.3 *AlterEventConditionListMonitoring protocol*

The abstract syntax of the alterECLMonitoring choice of the ConfirmedServiceRequest and ConfirmedServiceResponse is specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this clause.

```

AlterEventConditionListMonitoring-Request ::= SEQUENCE {
    eventConditionListName      [0] ObjectName,
    enabled                     [1] IMPLICIT BOOLEAN DEFAULT FALSE,
    priorityChange               [2] CHOICE {
        priorityValue           [0] IMPLICIT INTEGER,
        priorityReset           [1] IMPLICIT NULL
    } OPTIONAL
}

AlterEventConditionListMonitoring-Response ::= NULL
    
```

7.3.20.3.1 *AlterEventConditionList Monitoring-Request* (Demande-Modifier Surveillance de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection *alterECLMonitoring* du type *ConfirmedServiceRequest* doit être *AlterEventConditionListMonitoring-Request*.

7.3.20.3.2 *Alter EventConditionListMonitoring-Response* (Réponse-Modifier Surveillance de Liste de Conditions Événementielles)

La syntaxe abstraite de la sélection *alterECLMonitoring* du type *ConfirmedServiceResponse* doit être *AlterEventConditionListMonitoring-Response*.

7.4 *Le Service et le Protocole Déclencher*

Le présent paragraphe prescrit l'utilisation spécifique au Contrôle-Commande de procédé, du service et du protocole Déclencher.

7.4.1 *Paramètre Supplémentaire de Demande de Déclenchement*

La MMS prévoit un paramètre Supplémentaire de Demande de Déclenchement à définir par les normes d'accompagnement. La structure du paramètre Supplémentaire Demande de Déclenchement est définie dans le tableau 29.

Tableau 29 – Paramètre Supplémentaire Demande de Déclenchement

Nom du paramètre	Dem	Ind
Numéro de Version Proposé	M	M(=)
CBB de Paramètre Proposé	M	M(=)
Appel aux Services Pris en charge	M	M(=)
Appel aux Services Supplémentaires Pris en charge	M	M(=)
Appel à CBB Supplémentaire Pris en charge	M	M(=)

Le paramètre Supplémentaire Demande de Déclenchement doit contenir des paramètres supplémentaires relatifs à la communication dans le contexte de présentation, résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506. Les paramètres qui le constituent sont définis comme suit:

```

AlterEventConditionListMonitoring-Request ::= SEQUENCE {
    eventConditionListName  [0] ObjectName,
    enabled                 [1] IMPLICIT BOOLEAN DEFAULT FALSE,
    priorityChange          [2] CHOICE {
        priorityValue       [0] IMPLICIT INTEGER,
        priorityReset       [1] IMPLICIT NULL
    } OPTIONAL
}

AlterEventConditionListMonitoring-Response ::= NULL

```

7.3.20.3.1 *AlterEventConditionListMonitoring-Request*

The abstract syntax of the alterECLMonitoring choice of the ConfirmedServiceRequest type shall be the AlterEventConditionListMonitoring-Request.

7.3.20.3.2 *AlterEventConditionListMonitoring-Response*

The abstract syntax of the alterECLMonitoring choice of the ConfirmedServiceResponse type shall be the AlterEventConditionListMonitoring-Response.

7.4 *The Initiate Service and Protocol*

This subclause specifies the process control specific use of the Initiate service and protocol.

7.4.1 *Init Request Detail parameter*

MMS provides for a Init Request Detail parameter to be defined by companion standards. The structure of the Init Request Detail parameter is shown in table 29.

Table 29 – Init Request Detail parameter

Parameter name	Req	Ind
Proposed Version Number	M	M(=)
Proposed Parameter CBB	M	M(=)
Services Supported Calling	M	M(=)
Additional Services Supported Calling	M	M(=)
Additional CBB Supported Calling	M	M(=)

The Init Request Detail shall contain additional parameters relating to communication in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506. The component parameters are specified as follows:

7.4.1.1 *Numéro de Version Proposé*

Ce paramètre, du type entier, doit contenir un nombre qui représente un numéro de version mineure de la présente partie de l'ISO/IEC 9506. Ce numéro est le numéro de version mineure proposé qui sera utilisé dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 pour ce type de communication. Un numéro supérieur à un indique la prise en compte de toutes les versions mineures entre le numéro un et le numéro proposé.

NOTE – Les révisions majeures de la présente partie de l'ISO/IEC 9506 sont reflétées par la définition et l'enregistrement de syntaxe abstraite distincte (voir l'article 5 de l'ISO/IEC 9506-2). Les révisions mineures sont reflétées par le paramètre numéro de version mineure. Les versions mineures de la présente partie de l'ISO/IEC 9506 ayant le même niveau qu'une révision majeure sont compatibles avec les versions de la présente partie de l'ISO/IEC 9506 ayant des numéros de version mineure inférieurs.

Il est admis que la valeur de ce paramètre soit réduite par le fournisseur MMS s'il ne peut prendre en charge la valeur requise. La valeur dans la primitive d'indication doit être inférieure ou égale à la valeur dans la primitive de demande, mais ne peut être inférieure à un.

7.4.1.2 *CBB de Paramètre Proposé*

Ce paramètre est défini dans 8.2.3.2 de l'ISO/IEC 9506-1.

7.4.1.3 *Appel aux Services Pris en charge*

Ce paramètre est défini dans 8.2.3.3 de l'ISO/IEC 9506-1.

7.4.1.4 *Appel aux Services Supplémentaires Pris en Charge*

Ce paramètre, du type chaîne binaire, doit spécifier la Prise en Charge par l'utilisateur MMS Appelant d'un ensemble de services supplémentaires qui sont définis par la présente partie de l'ISO/IEC 9506 pour utilisation dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 pour l'association d'application considérée.

La valeur du paramètre dans la primitive d'indication doit préciser l'intersection entre l'ensemble de services supplémentaires Prise en Charge par l'utilisateur MMS Appelant et l'ensemble de services pris en charge par le fournisseur MMS.

L'affectation d'un service à un élément binaire particulier d'une chaîne de bits type est définie en 7.4.3. Une valeur un dans l'élément binaire affecté au service doit indiquer la prise en charge du service correspondant. Une valeur zéro doit indiquer la non prise en charge. Tout élément binaire supplémentaire doit être ignoré.

La prise en charge des services confirmés doit être définie comme étant la capacité de recevoir une indication de demande et d'exécuter correctement la procédure de service définie pour assurer le rôle de répondeur.

Si un service confirmé est pris en charge, une PDU de rejet ne doit pas être émise sur réception d'une demande de ce service, sauf du fait d'une erreur de protocole. Si un service confirmé n'est pas pris en charge, une PDU de rejet doit être émise sur réception d'une demande dudit service, avec code de rejet du type «SERVICE NON RECONNU».

7.4.1.1 *Proposed Version Number*

This parameter, of type integer, shall contain a number that represents a minor version number of this part of ISO/IEC 9506. This number is the proposed minor version number which will be used in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506 for this instance of communication. Proposal of a number greater than one indicates support for all minor versions between one and the number proposed.

NOTE – Major revisions of this part of ISO/IEC 9506 are reflected through the definition and registration of distinct abstract syntaxes (see clause 5 of ISO/IEC 9506-2). Minor revisions are reflected in the minor version number parameter. Minor versions of this part of ISO/IEC 9506 at the same major revision level are compatible with versions of this part of ISO/IEC 9506 with smaller minor version numbers.

The value of this parameter may be reduced by the MMS-provider if it cannot support the requested value. The value in the indication primitive shall be less than or equal to the value in the request primitive, but not less than one.

7.4.1.2 *Proposed Parameter CBB*

This parameter is specified in 8.2.3.2 of ISO/IEC 9506-1.

7.4.1.3 *Services Supported Calling*

This parameter is specified in 8.2.3.3 of ISO/IEC 9506-1.

7.4.1.4 *Additional Services Supported Calling*

This parameter, of type bitstring, shall specify support by the Calling MMS-user of a set of additional services that are defined by this part of ISO/IEC 9506 for use in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506 on the application association.

The value of the parameter in the indication primitive shall specify the intersection of the set of additional services supported by the Calling MMS-user and the set of services supported by the MMS-provider.

The assignment of a service to an individual bit of the bitstring type is specified in 7.4.3. A value of one in the assigned bit shall indicate support for the corresponding service. A value of zero shall indicate non-support. Any additional bits shall be ignored.

Support for confirmed services shall be defined as the ability to receive a request indication and properly execute the service procedure defined for the responder role.

If a confirmed service is supported, then a reject PDU shall not be issued on receipt of a request for that service, except for a protocol error. If a confirmed service is not supported, then a reject PDU shall be issued on receipt of a request for that service with a reject code of "UNRECOGNIZED SERVICE".

7.4.1.5 Appel à CBB Supplémentaire Pris en Charge

Ce paramètre, du type chaîne de bits, doit indiquer la prise en charge par l'utilisateur MMS Appelant d'un ensemble de paramètres CBB supplémentaires qui sont définis par la présente partie de l'ISO/IEC 9506 pour utilisation dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 sur l'association d'application.

La valeur du paramètre dans la primitive d'indication doit spécifier l'intersection entre l'ensemble de CBB supplémentaires pris en charge par l'utilisateur MMS Appelant et l'ensemble de CBB pris en charge par le fournisseur MMS.

Les définitions des CBB supplémentaires dans la présente partie de l'ISO/IEC 9506 sont fournies en 6.6. L'affectation d'un CBB à un élément binaire particulier du type chaîne binaire est définie en 7.4.3. Une valeur un de l'élément binaire affecté doit indiquer la prise en charge du CBB correspondant. Une valeur zéro doit indiquer la non prise en charge. Tout autre élément binaire supplémentaire doit être ignoré.

7.4.2 Paramètre Supplémentaire de Réponse d'Initialisation

La MMS prévoit un Paramètre Supplémentaire de Réponse d'Initialisation à définir par les normes d'accompagnement. La structure du paramètre Supplémentaire de Réponse d'Initialisation est définie dans le tableau 30.

Tableau 30 – Paramètre Supplémentaire de Réponse d'Initialisation

Nom du paramètre	Dem	Ind
Numéro de Version Pris en Charge	M	M(=)
CBB de Paramètre Pris en Charge	M	M(=)
Services Pris en Charge Appelés	M	M(=)
Services Supplémentaires Pris en Charge Appelés	M	M(=)
CBB Supplémentaire Pris en Charge Appelé	M	M(=)

Le Paramètre Supplémentaire de Réponse d'Initialisation doit contenir des paramètres relatifs à la communication dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506. Les paramètres qui le constituent sont définis comme suit:

7.4.2.1 Numéro de Version Pris en Charge

Ce paramètre, du type entier, doit contenir un nombre qui représente un numéro de version mineure de la présente partie de l'ISO/IEC 9506. Ce numéro est le numéro de version mineure de la présente partie de l'ISO/IEC 9506 qui sera utilisé dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 pour ce type de communication. Ce numéro doit être inférieur ou égal au paramètre Numéro de Version Proposé dans la primitive de demande. Il ne doit pas être inférieur à un.

NOTE – Les révisions majeures de la présente partie de l'ISO/IEC 9506 sont reflétées par la définition et l'inscription de syntaxe abstraite distincte (voir l'article 5 de l'ISO/IEC 9506-2). Les révisions mineures sont

7.4.1.5 Additional CBB Supported Calling

This parameter, of type bitstring, shall specify support by the Calling MMS-user of a set of additional parameter CBBs that are defined by this part of ISO/IEC 9506 for use in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506 on the application association.

The value of the parameter in the indication primitive shall specify the intersection of the set of additional CBBs supported by the Calling MMS-user and the set of CBBs supported by the MMS-provider.

The definition of additional CBBs in this part of ISO/IEC 9506 are specified in 6.6. The assignment of a CBB to an individual bit of the bitstring type is specified in 7.4.3. A value of one in the assigned bit shall indicate support for the corresponding CBB. A value of zero shall indicate non-support. Any additional bits shall be ignored.

7.4.2 Init Response Detail parameter

MMS provides for an Init Response Detail parameter to be defined by companion standards. The structure of the Init Response Detail parameter is shown in table 30.

Table 30 – Init Response Detail parameter

Parameter name	Req	Ind
Negotiated Version Number	M	M(=)
Negotiated Parameter CBB	M	M(=)
Services Supported Called	M	M(=)
Additional Services Supported Called	M	M(=)
Additional CBB Supported Called	M	M(=)

The Init Response Detail parameter shall contain parameters relating to communication in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506. The component parameters are specified as follows:

7.4.2.1 Negotiated Version Number

This parameter, of type integer, shall contain a number that represents a minor version number of this part of ISO/IEC 9506. This number is the minor version number of this part of ISO/IEC 9506 which will be used in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506 for this instance of communication. This number shall be less than or equal to the Proposed Version Number parameter in the request primitive. It shall not be reduced to less than one.

NOTE – Major revisions of this part of ISO/IEC 9506 are reflected through the definition and registration of distinct abstract syntaxes (see clause 5 of ISO/IEC 9506-2). Minor revisions are reflected in the minor version

reflétées par le paramètre numéro de version mineure. Les versions mineures de la présente partie de l'ISO/IEC 9506 ayant le même niveau qu'une révision majeure sont compatibles avec les versions de la présente partie de l'ISO/IEC 9506 ayant des numéros de version mineure inférieurs.

7.4.2.2 *CBB de Paramètre Pris en Charge*

Ce paramètre est défini dans 8.2.4.2 de l'ISO/IEC 9506-1.

7.4.2.3 *Services Appelés Pris en Charge*

Ce paramètre est spécifié dans 8.2.4.3 de l'ISO/IEC 9506-1.

7.4.2.4 *Services Appelés Supplémentaires Pris en Charge*

Ce paramètre, du type chaîne binaire, doit spécifier la prise en charge par l'utilisateur MMS Appelé d'un ensemble de services supplémentaires qui sont définis dans la présente partie de l'ISO/IEC 9506 pour utilisation dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 sur l'association d'application.

La valeur du paramètre dans la primitive d'indication doit spécifier l'intersection entre l'ensemble de services supplémentaires pris en charge par l'utilisateur MMS Appelé et l'ensemble de services supplémentaires pris en charge par le fournisseur MMS.

L'affectation d'un service à un élément binaire particulier du type chaîne binaire est spécifiée dans 7.4.3. Une valeur un de l'élément binaire affecté doit indiquer la prise en charge du service correspondant. Une valeur zéro doit indiquer la non prise en charge. Tout autre élément binaire supplémentaire doit être ignoré.

La prise en charge de services confirmés doit être définie comme étant la capacité de recevoir une indication de demande et d'exécuter correctement la procédure de service définie pour le rôle répondant.

Si un service confirmé est pris en charge, une PDU de rejet ne doit pas être émise sur réception d'une demande de service, sauf pour une erreur de protocole. Si un service confirmé n'est pas pris en charge, une PDU de Rejet doit être, dans ce cas, émise sur réception d'une demande pour ce service, avec code de rejet «SERVICE NON RECONNU».

7.4.2.5 *CBB Supplémentaire Pris en Charge Appelé*

Ce paramètre, du type chaîne binaire, doit spécifier la prise en charge par l'utilisateur MMS Appelé d'un ensemble de CBB de paramètre supplémentaires qui sont définis dans la présente partie de l'ISO/IEC 9506 pour utilisation dans le contexte de présentation résultant de la syntaxe abstraite définie dans la présente partie de l'ISO/IEC 9506 sur l'association d'application.

La valeur du paramètre dans la primitive d'indication doit spécifier l'intersection entre l'ensemble de CBB supplémentaires pris en charge par l'utilisateur MMS Appelé et l'ensemble de CBB supplémentaires pris en charge par le fournisseur MMS.

L'affectation d'un CBB à un élément binaire particulier du type chaîne binaire est spécifiée dans 7.4.3. Une valeur un dans le bit affecté doit indiquer la prise en charge du CBB correspondant. Une valeur zéro doit indiquer la non prise en charge. Tout élément binaire

number parameter. Minor versions of this part of ISO/IEC 9506 at the same major revision level are compatible with versions of this part of ISO/IEC 9506 with smaller minor version numbers.

7.4.2.2 *Negotiated Parameter CBB*

This parameter is specified in 8.2.4.2 of ISO/IEC 9506-1.

7.4.2.3 *Services Supported Called*

This parameter is specified in 8.2.4.3 of ISO/IEC 9506-1.

7.4.2.4 *Additional Services Supported Called*

This parameter, of type bitstring, shall specify support by the Called MMS-user of a set of additional services that are defined in this part of ISO/IEC 9506 for use in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506 on the application association.

The value of the parameter in the indication primitive shall specify the intersection of the set of additional services supported by the Called MMS-user and the set of additional services supported by the MMS-provider.

The assignment of a service to an individual bit of the bitstring type is specified in 7.4.3. A value of one in the assigned bit shall indicate support for the corresponding service. A value of zero shall indicate non-support. Any additional bits shall be ignored.

Support for confirmed services shall be defined as the ability to receive a request indication and properly execute the service procedure defined for the responder role.

If a confirmed service is supported, then a Reject PDU shall not be issued on receipt of a request for that service, except for a protocol error. If a confirmed service is not supported, then a Reject PDU shall be issued on receipt of a request for that service with a reject code of "UNRECOGNIZED SERVICE".

7.4.2.5 *Additional CBB Supported Called*

This parameter, of type bitstring, shall specify support by the Called MMS-user of a set of additional parameter CBBs that are defined in this part of ISO/IEC 9506 for use in the presentation context derived from the abstract syntax defined in this part of ISO/IEC 9506 on the application association.

The value of the parameter in the indication primitive shall specify the intersection of the set of additional CBBs supported by the Called MMS-user and the set of additional CBBs supported by the MMS-provider.

The assignment of a CBB to an individual bit of the bitstring type is specified in 7.4.3. A value of one in the assigned bit shall indicate support for the corresponding CBB. A value of zero shall indicate non-support. Any additional bits shall be ignored. Additional CBBs

supplémentaire doit être ignoré. Des CBB supplémentaires sont définis dans la présente partie de l'ISO/IEC 9506, en 6.6

7.4.3 Protocole Initiate (Déclenchement)

La syntaxe abstraite des paramètres Demande et Réponse Supplémentaires de Déclenchement doit être spécifiée par les types InitRequestDetail et InitResponseDetail, respectivement. Ces types sont définis ci-après et décrits dans les paragraphes suivants. Le paragraphe 5.5 de l'ISO/IEC 9506-2 décrit la dérivation de tous les paramètres pour lesquels des dérivations explicites ne sont pas fournies dans le présent paragraphe.

```

InitRequestDetail ::= SEQUENCE {
    proposedVersionNumber      [0] IMPLICIT Integer16,
    proposedParameterCBB       [1] IMPLICIT ParameterSupportOptions,
    servicesSupportedCalling    [2] IMPLICIT ServiceSupportOptions,
    additionalSupportedCalling  [3] IMPLICIT AdditionalSupportOptions,
    additionalCbbsupportedCalling [4] IMPLICIT AdditionalCbbOptions,
    privilegeClassIdentityCalling [5] IMPLICIT VisibleString
}

InitResponseDetail ::= SEQUENCE {
    negotiatedVersionNumber     [0] IMPLICIT Integer16,
    negotiatedParameterCBB      [1] IMPLICIT ParameterSupportOptions,
    servicesSupportedCalled      [2] IMPLICIT ServiceSupportOptions,
    additionalSupportedCalled    [3] IMPLICIT AdditionalSupportOptions,
    additionalCbbSupportedCalled [4] IMPLICIT AdditionalCbbOptions,
    privilegeClassIdentityCalled [5] IMPLICIT VisibleString
}

AdditionalSupportOptions ::= BITSTRING {
    -- Bits 0 - 3 are reserved
    initiateUnitControlLoad      (4),
    unitControlLoadSegment       (5),
    unitControlUpload            (6),
    startUnitControl             (7),
    stopUnitControl              (8),
    createUnitControl            (9),
    addToUnitControl             (10),
    removeFromUnitControl        (11),
    getUnitControlAttributes     (12),
    loadUnitControlFromFile      (13),
    storeUnitControlToFile       (14),
    deleteUnitControl           (15),
    defineEventConditionList     (16),
    deleteEventConditionList    (17),
    addEventConditionListReference (18),
    removeEventConditionListReference (19),
    getEventConditionListAttributes (20),

```

are defined in this part of ISO/IEC 9506 in 6.6.

7.4.3 Initiate protocol

The abstract syntax of the Init Request Detail and Init Response Detail parameters shall be specified by the InitRequestDetail and InitResponseDetail types, respectively. These types are specified below and described in the paragraphs that follow. Subclause 5.5 of ISO/IEC 9506-2 describes the derivation of all parameters for which explicit derivations are not provided in this subclause.

```

InitRequestDetail ::= SEQUENCE {
    proposedVersionNumber      [0] IMPLICIT Integer16,
    proposedParameterCBB       [1] IMPLICIT ParameterSupportOptions,
    servicesSupportedCalling    [2] IMPLICIT ServiceSupportOptions,
    additionalSupportedCalling  [3] IMPLICIT AdditionalSupportOptions,
    additionalCbbSupportedCalling [4] IMPLICIT AdditionalCbbOptions,
    privilegeClassIdentityCalling [5] IMPLICIT VisibleString
}

InitResponseDetail ::= SEQUENCE {
    negotiatedVersionNumber      [0] IMPLICIT Integer16,
    negotiatedParameterCBB       [1] IMPLICIT ParameterSupportOptions,
    servicesSupportedCalled       [2] IMPLICIT ServiceSupportOptions,
    additionalSupportedCalled     [3] IMPLICIT AdditionalSupportOptions,
    additionalCbbSupportedCalled  [4] IMPLICIT AdditionalCbbOptions,
    privilegeClassIdentityCalled  [5] IMPLICIT VisibleString
}

AdditionalSupportOptions ::= BITSTRING {
    -- Bits 0 - 3 are reserved
    initiateUnitControlLoad      (4),
    unitControlLoadSegment      (5),
    unitControlUpload            (6),
    startUnitControl             (7),
    stopUnitControl              (8),
    createUnitControl            (9),
    addToUnitControl             (10),
    removeFromUnitControl        (11),
    getUnitControlAttributes     (12),
    loadUnitControlFromFile      (13),
    storeUnitControlToFile       (14),
    deleteUnitControl           (15),
    defineEventConditionList     (16),
    deleteEventConditionList     (17),
    addEventConditionListReference (18),
    removeEventConditionListReference (19),
    getEventConditionListAttributes (20),

```

```

reportEventConditionListStatus      (21),
alterEventConditionListMonitoring    (22)
}
AdditionalCbbOptions ::= BITSTRING {
    DES      (0),
    DEI      (1),
    RECL     (2)
}

```

7.5 Extensions de protocole généralisées

La syntaxe abstraite de l'ISO/IEC 9506-2 doit, outre les extensions spécifiées ci-dessus, être étendue comme définie ci-après afin d'accueillir les extensions PDU requises pour la prise en charge de nouveaux services et erreurs définis dans le présent article.

7.5.1 Extensions ConfirmedServiceRequest (Demande de Service Confirmée)

La sélection AdditionalService-Request (Demande-Service Supplémentaire) de Confirmed Service-Request doit être définie comme décrit ci-après:

```

AdditionalService-Request ::= CHOICE {
    initiateUCLoad      [4] IMPLICIT InitiateUnitControlLoad-Request,
    uCLoad              [5] IMPLICIT UnitControlLoadSegment-Request,
    uCUpload            [6] IMPLICIT UnitControlUpload-Request,
    startUC             [7] IMPLICIT StartUnitControl-Request,
    stopUC              [8] IMPLICIT StopUnitControl-Request,
    createUC            [9] IMPLICIT CreateUnitControl-Request,
    addToUC             [10] IMPLICIT AddToUnitControl-Request,
    removeFromUC        [11] IMPLICIT RemoveFromUnitControl-Request,
    getUCAttributes     [12] IMPLICIT GetUnitControlAttributes-Request,
    loadUCFromFile      [13] IMPLICIT LoadUnitControlFromFile-Request,
    storeUCToFile       [14] IMPLICIT StoreUnitControlToFile-Request,
    deleteUC           [15] IMPLICIT DeleteUnitControl-Request,
    defineECL           [16] IMPLICIT DefineEventConditionList-Request,
    deleteECL          [17] IMPLICIT DeleteEventConditionList-Request,
    addECLReference     [18] IMPLICIT AddEventConditionListReference-Request,
    removeECLReference [19] IMPLICIT RemoveEventConditionListReference-Request,
    getECLAttributes   [20] IMPLICIT GetEventConditionListReference-Request,
    reportECLStatus    [21] IMPLICIT ReportEventConditionListStatus-Request,
    alterECLMonitoring [22] IMPLICIT AlterEventConditionListMonitoring-Request
}

```

```

reportEventConditionListStatus      (21),
alterEventConditionListMonitoring    (22)
}

```

```

AdditionalCbbOptions ::= BITSTRING {
    DES      (0),
    DEI      (1),
    RECL     (2)
}

```

7.5 Generalized protocol extensions

The abstract syntax of ISO/IEC 9506-2 shall, in addition to the extensions specified above, be extended as specified below in order to accommodate the PDU extensions required in support of new services and errors defined in this clause.

7.5.1 ConfirmedServiceRequest extensions

The AdditionalService-Request choice of the ConfirmedService-Request shall be defined as specified below:

```

AdditionalService-Request ::= CHOICE {
    initiateUCLoad      [4] IMPLICIT InitiateUnitControlLoad-Request,
    uCLoad              [5] IMPLICIT UnitControlLoadSegment-Request,
    uCUpload            [6] IMPLICIT UnitControlUpload-Request,
    startUC             [7] IMPLICIT StartUnitControl-Request,
    stopUC              [8] IMPLICIT StopUnitControl-Request,
    createUC            [9] IMPLICIT CreateUnitControl-Request,
    addToUC             [10] IMPLICIT AddToUnitControl-Request,
    removeFromUC        [11] IMPLICIT RemoveFromUnitControl-Request,
    getUCAttributes     [12] IMPLICIT GetUnitControlAttributes-Request,
    loadUCFromFile      [13] IMPLICIT LoadUnitControlFromFile-Request,
    storeUCToFile       [14] IMPLICIT StoreUnitControlToFile-Request,
    deleteUC           [15] IMPLICIT DeleteUnitControl-Request,
    defineECL           [16] IMPLICIT DefineEventConditionList-Request,
    deleteECL          [17] IMPLICIT DeleteEventConditionList-Request,
    addECLReference     [18] IMPLICIT AddEventConditionListReference-
        Request,
    removeECLReference  [19] IMPLICIT RemoveEventConditionListReference-
        Request,
    getECLAttributes    [20] IMPLICIT GetEventConditionListAttributes-
        Request,
    reportECLStatus     [21] IMPLICIT ReportEventConditionListStatus-
        Request,
    alterECLMonitoring  [22] IMPLICIT AlterEventConditionListMonitoring-
        Request
}

```

7.5.2 Extensions ConfirmedServiceResponse (Réponse de Service Confirmée)

La sélection ConfirmedService-Response du ConfirmedService-Response doit être définie comme prescrit ci-après:

```
AdditionalService-Response ::= CHOICE {
    initiateUCLoad      [4] IMPLICIT InitiateUnitControlLoad-Response,
    uCLoad              [5] IMPLICIT UnitControlLoadSegment-Response,
    uCUpload            [6] IMPLICIT UnitControlUpload-Response,
    startUC             [7] IMPLICIT StartUnitControl-Response,
    stopUC              [8] IMPLICIT StopUnitControl-Response,
    createUC            [9] IMPLICIT CreateUnitControl-Response,
    addToUC             [10] IMPLICIT AddToUnitControl-Response,
    removeFromUC        [11] IMPLICIT RemoveFromUnitControl-Response,
    getUCAAttributes    [12] IMPLICIT GetUnitControlFromFile-Response,
    loadIUCFromFile     [13] IMPLICIT LoadUnitControlFromFile-Response,
    storeUCToFile       [14] IMPLICIT StoreUnitControlToFile-Response,
    deleteUC           [15] IMPLICIT DeleteUnitControl-Response,
    defineECL           [16] IMPLICIT DefineEventConditionList-Response,
    deleteECL          [17] IMPLICIT DeleteEventConditionList-Response,
    addECLReference     [18] IMPLICIT AddEventConditionListReference
                        Response,
    removeECLReference [19] IMPLICIT RemoveEventConditionListReference
                        Response,
    getECLAttributes    [20] IMPLICIT GetEventConditionListAttributes
                        Response,
    reportECLStatus     [21] IMPLICIT ReportEventConditionListStatus
                        Response,
    alterECLMonitoring [22] IMPLICIT AlterEventConditionListMonitoring
                        Response
}
```

7.5.3 Extensions erreur spécifique au service

La sélection Servicesupplémentaire de la sélection InformationSpécifiqueauservice de la Classe d'erreur qui fait partie d'ErreurService doit être définie comme prescrit ci-après:

```
AdditionalService-Error ::= CHOICE {
    defineEcl          [0] DefineEventConditionList-Error,
    addECLReference    [1] AddEventConditionListReference-Error,
    removeECLReference [2] RemoveEventConditionListReference-Error,
    initiateUC         [3] InitiatedUnitControl-Error,
    startUC            [4] IMPLICIT StartUnitControl-Error,
    stopUC             [5] IMPLICIT StopUnitControl-Error,
    deleteUC          [6] IMPLICIT DeleteUnitControl-Error,
    loadUCFromFile     [7] IMPLICIT LoadUnitControlFromFile-Error
}
```

7.5.2 ConfirmedServiceResponse extensions

The ConfirmedService-Response choice of the ConfirmedService-Response shall be defined as specified below:

```
AdditionalService-Response ::= CHOICE {
  initiateUCLoad      [4] IMPLICIT InitiateUnitControlLoad-Response,
  uCLoad              [5] IMPLICIT UnitControlLoadSegment-Response,
  uCUpload            [6] IMPLICIT UnitControlUpload-Response,
  startUC              [7] IMPLICIT StartUnitControl-Response,
  stopUC              [8] IMPLICIT StopUnitControl-Response,
  createUC            [9] IMPLICIT CreateUnitControl-Response,
  addToUC             [10] IMPLICIT AddToUnitControl-Response,
  removeFromUC         [11] IMPLICIT RemoveFromUnitControl-Response,
  getUCAAttributes     [12] IMPLICIT GetUnitControlAttributes-Response,
  loadUCFromFile       [13] IMPLICIT LoadUnitControlFromFile-Response,
  storeUCToFile        [14] IMPLICIT StoreUnitControlToFile-Response,
  deleteUC             [15] IMPLICIT DeleteUnitControl-Response,
  defineECL            [16] IMPLICIT DefineEventConditionList-Response,
  deleteECL            [17] IMPLICIT DeleteEventConditionList-Response,
  addECLReference      [18] IMPLICIT AddEventConditionListReference-
    Response,
  removeECLReference   [19] IMPLICIT RemoveEventConditionListReference-
    Response,
  getECLAttributes     [20] IMPLICIT GetEventConditionListAttributes-
    Response,
  reportECLStatus      [21] IMPLICIT ReportEventConditionListStatus-
    Response,
  alterECLMonitoring   [22] IMPLICIT AlterEventConditionListMonitoring-
    Response
}
```

7.5.3 Service specific error extensions

The additionalService choice of the serviceSpecificInformation choice of the errorClass component of ServiceError shall be defined as specified below:

```
AdditionalService-Error ::= CHOICE {
  defineEcl           [0] DefineEventConditionList-Error,
  addECLReference      [1] AddEventConditionListReference-Error,
  removeECLReference   [2] RemoveEventConditionListReference-Error,
  initiateUC           [3] InitiateUnitControl-Error,
  startUC              [4] IMPLICIT StartUnitControl-Error,
  stopUC              [5] IMPLICIT StopUnitControl-Error,
  deleteUC             [6] IMPLICIT DeleteUnitControl-Error,
  loadUCFromFile       [7] IMPLICIT LoadUnitControlFromFile-Error
}
```

7.6 Fin de Module

L'instruction END (FIN) ci-après ferme le module.

END

8 Objets normalisés

Il n'y a pas d'objets normalisés spécifiés par la présente partie de l'ISO/IEC 9506. Les recommandations relatives aux noms de Variables sont fournies dans l'annexe D.

9 Conformité

La conformité aux prescriptions de la présente norme s'exprime au niveau de la prise en charge, ou de la non prise en charge de services, paramètres de service et procédures de service particuliers étendus par la présente partie de l'ISO/IEC 9506. Les mises en oeuvre conformes doivent rendre compte dans la PICS des services et options de service pris en charge.

9.1 Classes de conformité

9.1.1 Définition des classes de conformité

Les classes de conformité suivantes sont définies pour la présente partie de l'ISO/IEC 9506. Le tableau 31 fournit une présentation générale des classes de conformité.

Tableau 31 – Classes de conformité

Classe de Conformité	Fonctions de Contrôle-Commande de Procédé
1	Acquisition de Données Commande Paramétrique
2	Gestion du Programme
3	Acquisition de Données non Sollicitée Commande Verrouillée
4	Acquisition de Données Configurées Alarme Semaphore
5	Objet MMS Distant Instantiation
6	Consignateur
NOTE - Ces classes de conformité sont établies afin d'être compatibles avec des classes semblables dans les autres normes d'accompagnement.	

7.6 End of module

The following END statement closes the module.

END

8 Standardized objects

There are no standardized objects specified by this part of ISO/IEC 9506. Recommendations for Variable names are specified in annex D.

9 Conformance

Conformance to requirements of this standard shall be at the level of support, or non-support, of individual services, service parameters and service procedures extended by this part of ISO/IEC 9506. Conforming implementations shall report in the PICS which services and service options are supported.

9.1 Conformance classes

9.1.1 Definition of conformance classes

The following conformance classes are defined for this part of ISO/IEC 9056. An overview of the conformance classes is provided by table 31.

Table 31 – Conformance classes

Conformance Class	Process Control Functions
1	Data Acquisition Parametric Control
2	Program Management
3	Unsolicited Data Acquisition Interlocked Control
4	Configured Data Acquisition Alarming Semaphore
5	Remote MMS Object Instantiation
6	Historian
NOTE - These conformance classes are established to be compatible with similar classes in other companion standards.	

9.1.2 *Services exigés pour les classes de conformité*

Le tableau 32 fournit les services exigés pour chaque classe de conformité. Les services sont regroupés selon l'article pertinent de l'ISO/IEC 9506-1 ou selon l'article 7 de la présente partie de l'ISO/IEC 9506.

Un «s» dans le tableau indique qu'un système conforme doit prendre en charge ce service en tant que serveur. Un «x» indique qu'un système conforme doit prendre en charge ce service à la fois en tant que client et en tant que serveur.

IECNORM.COM : Click to view the full PDF of ISO/IEC 9506-6:1994

Withdrawn

9.1.2 *Services required for conformance classes*

The required services for each conformance class are given in table 32. Services are grouped by the relevant clause of ISO/IEC 9506-1 or by clause 7 of this part of ISO/IEC 9506.

An "s" in the table indicates that a conforming system shall support this service as a server. An "x" indicates that a conforming system shall support this service as both a client and as a server.

Withdrawing
IECNORM.COM : Click to view the full PDF of ISO/IEC 9506-6:1994

Tableau 32 – Exigences de service pour les classes de conformité

Service	Classe					
	1	2	3	4	5	6
Initiate	S	S	X	X	X	S
Conclude	S	S	X	X	X	S
Abort	S	S	S	S	S	S
Cancel				S	S	
Reject	S	S	S	S	S	S
Status	S	S	X	X	X	S
UnsolicitedStatus			X	X	X	
GetNameList		S	S	S	S	S
Identify	S	S	S	S	S	S
Rename						
GetCapabilityList						
InitiateDownloadSequence		S	S	S	S	
DownloadSegment		S	S	S	S	
TerminateDownloadSequence		S	S	S	S	
InitiateUploadSequence		S	S	S	S	
UploadSegment		S	S	S	S	
TerminateUploadSequence		S	S	S	S	
RequestDomainDownload						
RequestDomainUpload						
LoadDomainContent						
StoreDomainContent						
DeleteDomain		S	S	S	S	
GetDomainAttributes		S	S	S	S	
CreateProgramInvocation		S	S	S	S	
DeleteProgramInvocation		S	S	S	S	
Start		S	S	S	S	
Stop		S	S	S	S	
Resume		S	S	S	S	
Reset		S	S	S	S	
Kill						
GetProgramInvocationAttributes		S	S	S	S	
Read	S	S	X	X	X	
Write	S	S	X	X	X	
InformationReport			X	X	X	
GetVariableAccessAttributes			S	S	S	
DefineNamedVariable					S	
DefineScatteredAccess						
GetScatteredAccessAttributes						
DeleteVariableAccess					S	
DefineNamedVariableList					S	
GetNamedVariableListAttributes					S	
DeleteNamedVariableList					S	
DefineNamedType						
GetNamedTypeAttributes						
DeleteNamedType						

Table 32 – Service requirements for conformance classes

Service	Class					
	1	2	3	4	5	6
Initiate	S	S	X	X	X	S
Conclude	S	S	X	X	X	S
Abort	S	S	S	S	S	S
Cancel				S	S	
Reject	S	S	S	S	S	S
Status	S	S	X	X	X	S
UnsolicitedStatus			X	X	X	
GetNameList		S	S	S	S	S
Identify	S	S	S	S	S	S
Rename						
GetCapabilityList						
InitiateDownloadSequence		S	S	S	S	
DownloadSegment		S	S	S	S	
TerminateDownloadSequence		S	S	S	S	
InitiateUploadSequence		S	S	S	S	
UploadSegment		S	S	S	S	
TerminateUploadSequence		S	S	S	S	
RequestDomainDownload						
RequestDomainUpload						
LoadDomainContent						
StoreDomainContent						
DeleteDomain		S	S	S	S	
GetDomainAttributes		S	S	S	S	
CreateProgramInvocation		S	S	S	S	
DeleteProgramInvocation		S	S	S	S	
Start		S	S	S	S	
Stop		S	S	S	S	
Resume		S	S	S	S	
Reset		S	S	S	S	
Kill						
GetProgramInvocationAttributes		S	S	S	S	
Read	S	S	X	X	X	
Write	S	S	X	X	X	
InformationReport			X	X	X	
GetVariableAccessAttributes			S	S	S	
DefineNamedVariable					S	
DefineScatteredAccess						
GetScatteredAccessAttributes						
DeleteVariableAccess					S	
DefineNamedVariableList					S	
GetNamedVariableListAttributes					S	
DeleteNamedVariableList					S	
DefineNamedType						
GetNamedTypeAttributes						
DeleteNamedType						

Tableau 32 (suite)

Service	Classe					
	1	2	3	4	5	6
TakeControl				s	s	
RelinquishControl				s	s	
DefineSemaphore					s	
DeleteSemaphore					s	
ReportSemaphoreStatus				s	s	
ReportPoolSemaphoreStatus						
ReportSemaphoreEntryStatus				s	s	
AttachToSemaphore						
Input						
Output						
DefineEventCondition					s	
DeleteEventCondition					s	
GetEventConditionAttributes					s	
ReportEventConditionStatus				s	s	
AlterEventConditionMonitoring				s	s	
TriggerEvent						
DefineEventAction					s	
DeleteEventAction					s	
GetEventActionAttributes					s	
ReportEventActionStatus				s	s	
DefineEventEnrollment					s	
DeleteEventEnrollment					s	
GetEventEnrollmentAttributes					s	
ReportEventEnrollmentStatus				s	s	
AlterEventEnrollment					s	
EventNotification				s	s	
AcknowledgeEventNotification				s	s	
GetAlarmSummary				s	s	
GetAlarmEnrollmentSummary						
AttachToEventCondition						
ReadJournal						s
WriteJournal						s
InitializeJournal						s
ReportJournalStatus						s
CreateJournal						
DeleteJournal						
ObtainFile						
DataExchange			x	x	x	
GetDataExchangeAttributes					x	
InitiateUnitControlLoad						
UnitControlLoadSegment						
UnitControlUpload						
StartUnitControl						
StopUnitControl						
CreateUnitControl						
AddToUnitControl						

Table 32 (continued)

Service	Class					
	1	2	3	4	5	6
TakeControl				S	S	
RelinquishControl				S	S	
DefineSemaphore					S	
DeleteSemaphore					S	
ReportSemaphoreStatus				S	S	
ReportPoolSemaphoreStatus						
ReportSemaphoreEntryStatus				S	S	
AttachToSemaphore						
Input						
Output						
DefineEventCondition					S	
DeleteEventCondition					S	
GetEventConditionAttributes					S	
ReportEventConditionStatus				S	S	
AlterEventConditionMonitoring				S	S	
TriggerEvent						
DefineEventAction					S	
DeleteEventAction					S	
GetEventActionAttributes					S	
ReportEventActionStatus				S	S	
DefineEventEnrollment					S	
DeleteEventEnrollment					S	
GetEventEnrollmentAttributes					S	
ReportEventEnrollmentStatus				S	S	
AlterEventEnrollment					S	
EventNotification				S	S	
AcknowledgeEventNotification				S	S	
GetAlarmSummary				S	S	
GetAlarmEnrollmentSummary						
AttachToEventCondition						
ReadJournal						S
WriteJournal						S
InitializeJournal						S
ReportJournalStatus						S
CreateJournal						
DeleteJournal						
ObtainFile						
DataExchange			X	X	x	
GetDataExchangeAttributes					x	
InitiateUnitControlLoad						
UnitControlLoadSegment						
UnitControlUpload						
StartUnitControl						
StopUnitControl						
CreateUnitControl						
AddToUnitControl						

Tableau 32 (fin)

Service	Classe					
	1	2	3	4	5	6
RemoveFromUnitControl GetUnitControlAttributes LoadUnitControlFromFile StoreUnitControlToFile DeleteUnitControl						
DefineEventConditionList DeleteEventConditionList AddEventConditionListReference RemoveEventConditionListReference GetEventConditionListAttributes ReportEventConditionListStatus AlterEventConditionListMonitoring						

9.1.3 CBB de paramètre exigés pour les classes de conformité

Le tableau 33 fournit les CBB de paramètre exigés pour chaque classe de conformité. Les CBB de paramètre sont définis dans l'ISO/IEC 9506-1 ou dans l'article 7 de la présente partie de l'ISO/IEC 9506.

Un «x» indique qu'un système conforme doit prendre en charge ce CBB de paramètre. Pour ce qui concerne le paramètre NEST, la valeur minimale acceptable pour un système conforme est indiquée.

Tableau 33 – Exigences de paramètre pour les classes de conformité

CBB de Paramètre	Classe					
	1	2	3	4	5	6
STR1			x	x	x	
STR2			x	x	x	
VNAM	x	x	x	x	x	
VADR						
VALT				x	x	
VSCA						
TPY						
VLIS						
REAL						
AKEC						
CEI						
DES						
DEI						
RECL						
NEST	0	0	1	1	1	0

Table 32 (concluded)

Service	Class					
	1	2	3	4	5	6
RemoveFromUnitControl						
GetUnitControlAttributes						
LoadUnitControlFromFile						
StoreUnitControlToFile						
DeleteUnitControl						
DefineEventConditionList						
DeleteEventConditionList						
AddEventConditionListReference						
RemoveEventConditionListReference						
GetEventConditionListAttributes						
ReportEventConditionListStatus						
AlterEventConditionListMonitoring						

9.1.3 Parameter CBBs required for conformance classes

The required parameter CBBs for each conformance class are given in table 33. Parameter CBBs are defined in ISO/IEC 9506-1 or in clause 7 of this part of ISO/IEC 9506.

An "x" indicates that a conforming system shall support this parameter CBB. For the NEST parameter, the minimum acceptable value for a conforming system is shown.

Table 33 - Parameter requirements for conformance classes

Parameter CBB	Class					
	1	2	3	4	5	6
STR1			x	x	x	
STR2			x	x	x	
VNAM	x	x	x	x	x	
VADR						
VALT				x	x	
VSCA						
TPY						
VLIS						
REAL						
AKEC						
CEI						
DES						
DEI						
RECL						
NEST	0	0	1	1	1	0

9.2 Partie Un de la PICS: Information de Mise en oeuvre

Les dispositions de 18.2 de l'ISO/IEC 9506-2 doivent s'appliquer sans modification. Une mise en oeuvre conforme doit compléter la partie Un de la PICS de l'ISO/IEC 9506-2.

9.3 Partie Deux de la PICS: CBB de Service

Les dispositions de 18.3 de l'ISO/IEC 9506-2 doivent s'appliquer sans modification. Une mise en oeuvre conforme doit compléter la partie Deux de la PICS de l'ISO/IEC 9506-2. En outre, le tableau 34 ci-dessous doit être complété afin de fournir des informations PICS supplémentaires concernant la présente partie de l'ISO/IEC 9506.

Tableau 34 – CBB de Services supplémentaires

Bloc de Conformité de Services Supplémentaires	Serveur, Client, ou les deux
InitiateUnitControlLoad UnitControlLoadSegment UnitControlUpload StartUnitControl StopUnitControl CreateUnitControl AddToUnitControl RemoveFromUnitControl GetUnitControlAttributes LoadUnitControlFromFile StoreUnitControlToFile DeleteUnitControl DefineEventConditionList DeleteEventConditionList AddEventConditionListReference RemoveEventConditionListReference GetEventConditionListAttributes ReportEventConditionListStatus AlterEventConditionListMonitoring	

9.4 Partie Trois de la PICS: CBB de paramètre

Les dispositions du paragraphe 18.4 de l'ISO/IEC 9506-2 doivent s'appliquer sans modification. Une mise en oeuvre conforme doit compléter la partie Trois de la PICS de l'ISO/IEC 9506-2. En outre, une mise en oeuvre conforme doit compléter le tableau 35 ci-dessous pour indiquer la prise en charge des CBB de paramètre supplémentaires.

9.2 PICS Part One: Implementation information

The provisions of 18.2 of ISO/IEC 9506-2 shall apply without alteration. A conforming implementation shall complete the PICS part One of ISO/IEC 9506-2.

9.3 PICS part Two: Service CBBs

The provisions of 18.3 of ISO/IEC 9506-2 shall apply without alteration. A conforming implementation shall complete the PICS part Two of ISO/IEC 9506-2. In addition, table 34 shall be completed to provide additional PICS information relevant to this part of ISO/IEC 9506.

Table 34 – Additional Service CBBs

Additional Service Conformance Building Blocks	Server, Client or Both
InitiateUnitControlLoad UnitControlLoadSegment UnitControlUpload StartUnitControl StopUnitControl CreateUnitControl AddToUnitControl RemoveFromUnitControl GetUnitControlAttributes LoadUnitControlFromFile StoreUnitControlToFile DeleteUnitControl DefineEventConditionList DeleteEventConditionList AddEventConditionListReference RemoveEventConditionListReference GetEventConditionListAttributes ReportEventConditionListStatus AlterEventConditionListMonitoring	

9.4 PICS Part Three: Parameter CBBs

The provisions of subclause 18.4 of ISO/IEC 9506-2 shall apply without alteration. A conforming implementation shall complete the PICS part Three of ISO/IEC 9506-2. In addition, a conforming implementation shall complete table 35 to indicate support for additional parameter CBBs.

Tableau 35 – CBB de paramètres supplémentaires

Bloc de Conformité de Paramètres Supplémentaires	Pris en Charge
DES	
DEI	
RECL	

9.5 Partie Quatre de la PICS: Valeurs de mise en œuvre locales

Les dispositions de 18.5 de l'ISO/IEC 9506-2 doivent s'appliquer sans modification. Une mise en œuvre conforme doit compléter la partie Quatre de la PICS de l'ISO/IEC 9506-2.

Table 35 – Additional parameter CBBs

Additional Parameter Conformance Building Blocks	Supported
DES	
DEI	
RECL	

9.5 PICS Part Four: Local implementation values

The provisions of 18.5 of ISO/IEC 9506-2 shall apply without alteration. A conforming implementation shall complete the PICS part Four of ISO/IEC 9506-2.

Annexe A (normative)

Modèle d'Association d'Application

A.1 Généralités

La présente annexe prescrit la structure de l'objet Association d'Application disponible dans le MMS. Elle est incluse en tant qu'annexe jusqu'à ce que l'ISO/IEC 9506-1 soit amendée pour incorporer ce modèle.

Le modèle objet du VMD dans l'ISO/IEC 9506-1 est étendu pour inclure une Liste de références d'Association d'Application.

A.2 Association d'Application

L'Association d'Application identifie une instance spécifique de communication du VMD avec un client MMS.

Object: Application Association

Key Attribute: Application Association Identifier

Attribute: AP title of MMS Client

Attribute: Authentication Unit employed (TRUE, FALSE)

Constraint: Authentication Unit employed = TRUE

Attribute: Authentication Value

Attribute: Other ACSE parameters

Attribute: List of AA-Specific named objects

Attribute: List of Transaction Objects

A.2.1 Identificateur d'Association d'Application

Cet attribut identifie l'Association d'Application. Cet attribut n'étant jamais communiqué, sa forme est décidée localement.

A.2.2 Titre AP de Client MMS

Cet attribut, qui résulte des paramètres du service A-ASSOCIATE, identifie le client MMS présent sur cette association.

A.2.3 Unité d'Authentification employée

Cet attribut indique si l'unité d'authentification de la ACSE a été (VRAI) ou non (FAUX) utilisée pour l'établissement de cette association. Si cet attribut est VRAI, l'attribut suivant apparaît également.