NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
1334-4-42

Première édition
First edition
1996-09

Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs –

Partie 4:
Protocoles de communication de données –
Section 42: Protocoles d'application –
Couche application

Distribution automation using distribution line carrier systems –

Part 4:
Data communication protocols –
Section 42: Application protocols –
Application layer

Numéro de référence
Reference number
CEI/IEC 1334-4-42: 1996

Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles auprès du Bureau Central de la CEI.

Les renseignements relatifs à ces révisions, à l'établissement des éditions révisées et aux amendements peuvent être obtenus auprès des Comités nationaux de la CEI et dans les documents ci-dessous:

- **Bulletin de la CEI**
- **Annuaire de la CEI**
  Publié annuellement
- **Catalogue des publications de la CEI**
  Publié annuellement et mis à jour régulièrement

Terminologie

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 50: *Vocabulaire Electrotechnique International* (VEI), qui se présente sous forme de chapitres séparés traitant chacun d'un sujet défini. Des détails complets sur le VEI peuvent être obtenus sur demande. Voir également le dictionnaire multilingue de la CEI.

Les termes et définitions figurant dans la présente publication ont été soit tirés du VEI, soit spécifiquement approuvés aux fins de cette publication.

Symboles graphiques et littéraux

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera:

– la CEI 27: *Symboles littéraux à utiliser en électro-technique;*

– la CEI 417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles;*

– la CEI 617: *Symboles graphiques pour schémas;*

et pour les appareils électromédicaux,

– la CEI 878: *Symboles graphiques pour équipements électriques en pratique médicale.*

Les symboles et signes contenus dans la présente publication ont été soit tirés de la CEI 27, de la CEI 417, de la CEI 617 et/ou de la CEI 878, soit spécifiquement approuvés aux fins de cette publication.

Publications de la CEI établies par le même comité d'études

L'attention du lecteur est attirée sur les listes figurant à la fin de cette publication, qui énumèrent les publications de la CEI préparées par le comité d'études qui a établi la présente publication.

Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available from the IEC Central Office.

Information on the revision work, the issue of revised editions and amendments may be obtained from IEC National Committees and from the following IEC sources:

- **IEC Bulletin**
- **IEC Yearbook**
  Published yearly
- **Catalogue of IEC publications**
  Published yearly with regular updates

Terminology

For general terminology, readers are referred to IEC 50: *International Electrotechnical Vocabulary* (IEV), which is issued in the form of separate chapters each dealing with a specific field. Full details of the IEV will be supplied on request. See also the IEC Multilingual Dictionary.

The terms and definitions contained in the present publication have either been taken from the IEV or have been specifically approved for the purpose of this publication.

Graphical and letter symbols

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications:

– IEC 27: *Letter symbols to be used in electrical technology;*

– IEC 417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets;*

– IEC 617: *Graphical symbols for diagrams;*

and for medical electrical equipment,

– IEC 878: *Graphical symbols for electromedical equipment in medical practice.*

The symbols and signs contained in the present publication have either been taken from IEC 27, IEC 417, IEC 617 and/or IEC 878, or have been specifically approved for the purpose of this publication.

IEC publications prepared by the same technical committee

The attention of readers is drawn to the end pages of this publication which list the IEC publications issued by the technical committee which has prepared the present publication.

NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
1334-4-42

Première édition
First edition
1996-09

Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs –

Partie 4: Protocoles de communication de données – Section 42: Protocoles d'application – Couche application

Distribution automation using distribution line carrier systems –

Part 4: Data communication protocols – Section 42: Application protocols – Application layer

Bureau central de la Commission Electrotechnique Internationale 3, rue de Varembé Genève, Suisse

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE X

Pour prix, voir catalogue en vigueur
For price, see current catalogue

SOMMAIRE

CONTENTS

Articles Pages

Clause Page

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

AUTOMATISATION DE LA DISTRIBUTION À L'AIDE DE SYSTÈMES DE COMMUNICATION À COURANTS PORTEURS –

Partie 4: Protocoles de communication de données – Section 42: Protocoles d'application – Couche application

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 concernant les questions techniques, représentent, dans la mesure du possible un accord international sur les sujets étudiés, étant donné que les Comités nationaux intéressés sont représentés dans chaque comité d'études.

3) Les documents produits se présentent sous la forme de recommandations internationales. Ils sont publiés comme normes, rapports techniques ou guides et agréés comme tels 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.

5) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand un matériel est déclaré conforme à l'une de ses normes.

6) L'attention est attirée sur le fait que certains des éléments de la présente Norme internationale peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

La Norme internationale CEI 1334-4-42 a été établie par le comité d'études 57 de la CEI: Conduite des systèmes de puissance et communications associées.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
57/265/FDIS	57/293/RVD

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'annexe A fait partie intégrante de cette norme.

L'annexe B est donnée uniquement à titre d'information.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

DISTRIBUTION AUTOMATION USING DISTRIBUTION LINE CARRIER SYSTEMS –

Part 4: Data communication protocols – Section 42: Application protocols – Application layer

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 co-operation 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 express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.

3) The documents produced have the form of recommendations for international use and are 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.

5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.

6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 1334-4-42 has been prepared by IEC technical committee 57: Power system control and associated communications.

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

FDIS	Report on voting
57/265/FDIS	57/293/RVD

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

Annex A forms an integral part of this standard.

Annex B is for information only.

INTRODUCTION

La couche application supporte la spécification d'applications DLMS ainsi que la gestion des applications de chaque station. DCP (protocole de communication de données) utilise les normes ISO ACSE (éléments de service de contrôle d'association) en mode sans connexion, ainsi que les éléments de services d'applications de DLMS et la gestion.

INTRODUCTION

The application layer supports the DLMS application specification and the management application of each station. DCP makes use of the connectionless ACSE ISO standards and of the application service elements DLMS and management.

AUTOMATISATION DE LA DISTRIBUTION À L'AIDE DE SYSTÈMES DE COMMUNICATION À COURANTS PORTEURS –

Partie 4: Protocoles de communication de données – Section 42: Protocoles d'application – Couche application

1 Domaine d'application

Les spécifications fournies dans les sections de la CEI 1334-4 s'appliquent à la communication par la technique dite des courants porteurs (DLC), à la fois sur les réseaux basse et moyenne tension. La palette d'applications servies par ce procédé de communication est étendue et ne saurait être décrite de façon exhaustive dans la présente section; on peut citer à titre d'exemple le contrôle et la surveillance de réseaux de distribution, la diffusion de commandes, le contrôle des interfaces clientèles, de l'éclairage public, la supervision des feux de signalisation routière, le relevé automatique de compteurs, etc.

La présente section de la CEI 1334-4 décrit les règles utilisées pour définir les éléments de service d'une application, les ACSE de l'ISO en mode sans connection, et "DLMS Application Service Element" (éléments de service d'application de DLMS).

Des extensions à d'autres supports de communication sont également autorisés.

L'interfaçage des services DLMS est basé sur une pile de protocoles à trois couches. Des extensions futures à des piles de protocoles de plus de trois couches sont possibles et peuvent utiliser les règles décrites dans la présente section. Dans un tel cas, l'interface DLMS avec les couches inférieures devrait être modifiée: la spécification des critères d'interfaçage sera traitée dans des annexes correspondantes, toujours en respectant les règles définies dans la présente section.

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 section de la CEI 1334-4. 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 section de la CEI 1334-4 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.

CEI 1334-4-1: 1996, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4: Protocoles de communication de données – Section 1: Modèle de référence du système de communication*

CEI 1334-4-32: 1996, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4: Protocoles de communication de données – Section 32: Couche liaison de données – Contrôle de liaison logique (LLC)*

CEI 1334-4-41: 1996, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4: Protocoles de communication de données – Section 41: Protocoles d'application – Spécification des messages de ligne de distribution*

DISTRIBUTION AUTOMATION USING DISTRIBUTION LINE CARRIER SYSTEMS –

Part 4: Data communication protocols – Section 42: Application protocols – Application layer

1 Scope

The specifications of the sections of IEC 1334-4 apply to the communication through the so-called distribution line carrier technology (DLC) on both low and medium voltage distribution networks. The application range based on telecommunication processes is wide and cannot be described exhaustively in this section; application examples are: control and monitoring of the distribution network, order broadcast, control of user interfaces, public lighting, traffic lights supervision, automatic meter reading, etc.

This section of IEC 1334-4 describes the rules used to design an application service element, the ISO connectionless ACSE, and the DLMS Application Service Element.

Extensions to other communication media are also allowed.

The mapping of the DLMS services is based on the three-layer stack. Future extensions to stacks of more than three layers are possible and may use the rules described in this section. In such a case, the DLMS interface with the underlying layers should be remapped to the new one: the specification of the mapping criteria will be dealt with in corresponding appendices, in respect of the rules defined in this section.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this section of IEC 1334-4. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this section of IEC 1334-4 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 1334-4-1:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 1: Reference model of the communication system*

IEC 1334-4-32: 1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 32: Data link layer – Logical link control (LLC)*

IEC 1334-4-41: 1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 41: Application protocols – Distribution line message specification*

ISO 7498: 1984, *Systèmes de traitement de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base*

ISO 7498-3: 1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 3: Dénomination et adressage*

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 2 (1991): Service ACSE en mode sans connexion

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 8825: 1990, *Technologies de l'information – Interconnexion de systèmes ouverts – Spécification de règles de base pour coder la notation de syntaxe abstraite numéro une (ASN.1)*
(Publiée actuellement en anglais seulement)

ISO/IEC 9545: 1994, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche application*
(Publiée actuellement en anglais seulement)

3 Définitions

Pour les besoins de la présente Norme internationale, les définitions suivantes s'appliquent:

3.1 *Définitions du modèle de référence*

La présente section est basée sur des concepts développés dans l'ISO 7498 et utilise les termes suivants qui y sont définis:

a) couche application;
b) processus d'application;
c) entité d'application;
d) (N)-function (fonction (N));
e) (N)-layer (couche (N));
f) (N)-protocol (protocole (N));
g) (N)-protocol-control-information (information de contrôle du protocole (N));
h) (N)-protocol-data-unit (unité de données de protocole (N));
i) système ouvert réel.

ISO 7498: 1984, *Information processing systems – Open Systems Interconnection – Basic Reference Model*

ISO 7498-3: 1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 3: Naming and Addressing*

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

ISO 8649: 1988, *Information processing systems – Open Systems Interconnection – Service definition for the Association Control Service Element*
Amendment 2 (1991): Connectionless-mode ACSE service

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

ISO/IEC 8825: 1990, *Information technology – Open Systems Interconnection – Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*

ISO/IEC 9545: 1994, *Information technology – Open Systems Interconnection – Application layer structure*

3 Definitions

For the purposes of this International Standard, the following definitions apply:

3.1 *Reference model definitions*

This section is based on the concepts developed in ISO 7498 and makes use of the following terms defined in it:

a) application layer;
b) application process;
c) application entity;
d) (N) – function;
e) (N) – layer;
f) (N) – protocol;
g) (N) – protocol-control-information;
h) (N) – protocol-data-unit;
i) real open system.

3.2 *Définitions de nomage et d'adressage*

La présente section utilise les termes suivants:

a) titre d'un processus d'application;

b) qualificateur d'entité d'application;

c) titre d'une entité application;

d) application-process invocation-identifier (identificateur d'invocation de processus d'application);

e) application-entity invocation-identifier (identificateur d'invocation d'entité application).

3.3 *Définitions des conventions de service*

La présente section utilise les termes suivants définis dans l'ISO/TR 8509:

a) fournisseur de service;

b) utilisateur d'un service;

c) service confirmé;

d) service non confirmé;

e) service initialisé par le fournisseur;

f) primitive;

g) primitive de demande;

h) primitive d'indication;

i) primitive de réponse;

j) primitive de confirmation.

3.4 *Définitions spécifiques de la couche d'application*

3.4.1 *L'association application-association*

Relation coopérative entre deux entités application, constituée par l'échange des application-protocol-control-information (information de contrôle de protocole d'application) à l'aide des services data link (liaison de données). Elle rend possible la communication d'information ainsi que la coordination de leur exploitation conjointe pour un cas de communication.

3.4.2 *Contexte d'application*

Ensemble identifié explicitement d'application-service-éléments (éléments de service d'application) relatifs aux choix et règles partagés par deux invocations d'entités application, nécessaires à l'interfonctionnement des entités application d'une association d'application.

3.4.3 *Application-directory-function (fonction de répertoire d'application)*

Fonction d'application qui traite des adresses de liaison de données, des titres d'entité d'application, de l'information d'adressage du protocole d'application afin d'établir une correspondance entre ces catégories d'informations.

3.4.4 *Application-entity-invocation (invocation d'entité application)*

Utilisation spécifique de tout ou partie des capacités d'une entité application en support des exigences de communication d'une invocation de processus d'application.

3.2 *Naming and addressing definitions*

This section makes use of the following terms:

a) application-process title;
b) application-entity qualifier;
c) application-entity title;
d) application-process invocation-identifier;
e) application-entity invocation-identifier.

3.3 *Service conventions definitions*

This section makes use of the following terms defined in ISO/TR 8509:

a) service-provider;
b) service-user;
c) confirmed service;
d) non-confirmed service;
e) provider-initiated service;
f) primitive;
g) request (primitive);
h) indication (primitive);
i) response (primitive);
j) confirm (primitive).

3.4 *Application layer specific definitions*

3.4.1 *Application-association, association*

A co-operative relationship between two applications-entities, formed by their exchange of application-protocol-control-information through their use of data link services. The relationship enables the communication of information and the co-ordination of their joint operation for an instance of communication.

3.4.2 *Application context*

An explicitly identified set of application-service-elements, related options and rules shared in common by two application entity-invocations that are necessary for the interworking of application-entities in an application-association.

3.4.3 *Application-directory-function*

An application-function that processes data link addresses, application-entity-titles, and application-protocol-addressing-information to provide mapping among these categories of information.

3.4.4 *Application-entity-invocation*

A specific use of part or all of the capabilities of a given application-entity in support of the communications requirements of an application-process-invocation.

3.4.5 *Application-entity-type (type d'entité application)*

Ensemble spécifique (dit type) de fonctions d'entité application qui partagent des caractéristiques communes. Un tel type peut être référencé par un nom.

3.4.6 *Application-function (fonction d'application)*

Partie autonome identifiable de l'activité d'une entité application.

3.4.7 *Application-process-invocation (invocation d'un processus d'application)*

Utilisation spécifique de tout ou partie des capacités d'un processus d'application à l'occasion d'une utilisation spécifique de traitement de l'information.

3.4.8 *Application-process-type (type de processus d'application)*

Ensemble spécifique (dit type) de fonctions de processus d'application qui partagent des caractéristiques communes et qui peuvent être référencées par un nom.

3.4.9 *Application-protocol (protocole d'application*)

Ensemble de règles et de formats (sémantiques et syntaxiques) qui détermine le comportement en communication d'une entité application dans l'accomplissement d'une fonction application.

3.4.10 *Application-protocol-addressing-information (information d'adressage de protocole d'application)*

Les éléments des protocoles d'application qui contiennent des informations d'adressage.

3.4.11 *Application-protocol-control-information (information de contrôle de protocole application)*

L'information échangée entre les entités application utilisant le service data link (liaison de données) pour coordonner leur exploitation conjointe. Ces informations sont contenues dans les unités de protocole d'application (application PDU)

3.4.12 *Application-protocol-data-unit (unité de données du protocole application)*

Unité d'information, spécifiée dans un protocole application, qui se compose d'informations de contrôle de protocole application et éventuellement d'informations des utilisateurs.

3.4.13 *Application-service-element (ASE) (élément de service d'application)*

Ensemble de fonctions qui fournissent à des invocations d'entités application qui interfonctionnent, des moyens satisfaisant des besoins spécifiques pour une seule association d'applications.

NOTE –Cette définition telle qu'elle est spécifiée dans l'ISO/IEC 9545 affine la définition originale d'élément de service d'application dans l'ISO 7498.

3.4.14 *Association-control-service-element (élément de service de contrôle d'association)*

Elément de service d'application qui fournit le seul moyen consistant pour établir et terminer toutes les associations d'applications.

3.4.5 *Application-entity-type*

A specific set (called type) of application-entity functions that share common characteristics. Such a type may be referenced with a name.

3.4.6 *Application-function*

An identifiable stand alone part of the activity of application-entities.

3.4.7 *Application-process-invocation*

A specific utilization of part or all of the capabilities of a given application-process in support of a specific occasion of information processing.

3.4.8 *Application-process-type*

A specific set (called type) of application-process functions that share common characteristics. Such a type may be referenced with a name.

3.4.9 *Application-protocol*

A set of rules and formats (semantic and syntactic) which determines the communication behaviour of application-entities in the performance of application-functions.

3.4.10 *Application-protocol-addressing-information*

Those elements of the application-protocol-control-information which contain addressing information.

3.4.11 *Application-protocol-control-information*

The information exchanged between application-entities using data link services, to co-ordinate their joint operation. This information is contained in application-protocol-data-unit.

3.4.12 *Application-protocol-data-unit*

A unit of information, specified in an application-protocol, that consists of application-protocol-control-information and, possibly, user information.

3.4.13 *Application-service-element*

A set of functions that provide a capability for the interworking of application-entity-invocations for a specific purpose on a single application-association.

NOTE – This definition refines the original definition of application-service-elements in ISO 7498, but is as specified in ISO/IEC 9545.

3.4.14 *Association-control-service-element*

An application-service-element that provides a single consistent means for establishing and terminating all application-associations.

4 Abréviations

Pour les besoins de la présente Norme internationale, les abréviations suivantes s'appliquent:

AA : association d'applications;
ACSE : élément de service de contrôle d'association;
AE : entité application;
AEI : invocation d'entité application;
AP : processus d'application;
APAI : information d'adressage de protocole d'application;
APCI : information de contrôle de protocole d'application;
APDU : unité de données de protocole d'application;
ASE : élément de service d'application;
DASE : élément de service d'application DLMS;
DLMS : spécification de message de ligne de distribution;
OSI : interconnexion des systèmes ouverts.

5 Concepts de la couche application

Le présent article décrit les concepts de la couche application tels qu'ils sont décrits dans l'ISO/IEC 9545.

Les concepts développés dans la présente norme ont été étudiés pour le modèle OSI à sept couches dans le mode avec connexion. DCP utilise un modèle à trois couches et le mode sans connexion, mais ces concepts s'appliquent toujours.

Cet article réutilise donc la plupart des concepts de l'ISO/IEC 9545 ainsi que la structure avec les considérations suivantes:

– comme il n'y a pas de couche présentation, toutes les références à cette couche sont supprimées et les fonctionnalités nécessaires sont fournies directement par la couche application;

– la couche application utilise directement les services data link (liaison de données);

– l'ACSE référencé par la spécification DCP n'utilise que des services non confirmés tels qu'ils sont définis dans l'amendement 2 de l'ISO 8649.

5.1 *Introduction*

Les normes OSI sont prévues pour satisfaire les besoins de communication des applications (par exemple les tâches de traitement de l'information) qui demandent une coordination d'activités de traitement dans deux ou plusieurs systèmes réels ouverts. En particulier, les normes de la couche application de l'OSI définissent des procédures pour le support du traitement distribué de l'information.

4 Abbreviations

For the purposes of this International Standard, the following abbreviations apply:

AA : application-association;
ACSE : association control service element;
AE : application entity;
AEI : application entity invocation;
AP : application process;
APAI : application protocol addressing information;
APCI : application protocol control information;
APDU : application protocol data unit;
ASE : application service element;
DASE : DLMS application service element;
DLMS : distribution line message specification;
OSI : open systems interconnection.

5 Application layer concepts

This clause describes the application layer concepts as they are defined in ISO/IEC 9545.

The concepts developed in this standard were designed for the seven layers connection oriented OSI model. DCP makes use of a three layers connectionless model, but the concepts still apply.

This specification, then, re-uses most of the ISO/IEC 9545 concepts and structure with the following special considerations:

– because no presentation layer is available, all references to it are removed and its necessary functionalities are performed directly by the application layer;

– the application layer makes a direct use of the data link services;

– the ACSE referenced by the DCP specification uses only unconfirmed services as defined in the ISO 8649 amendment 2.

5.1 *Introduction*

Standards for OSI are intended to support the communication requirements of applications (that is information processing tasks) requiring co-ordinated processing activities in two or more real open systems. In particular, standards for the OSI application layer define procedures for the support of distributed information processing.

La couche application est supportée par les couches de niveau inférieur dans l'OSI. Dans DCP, cela implique en particulier que la couche application utilise les services data link (liaison de données) spécifiés dans la future CEI 1334-4-31 [1]* et la CEI 1334-4-32.

La couche application diffère des autres couches de l'OSI par plusieurs points importants:

– les entités de la couche application sont constituées de collections d'ASE (éléments de service d'application), chacun d'eux étant défini par un ensemble de services et de protocoles normalisés. Ces ASE sont combinés de différentes manières pour former les différents types d'AE (entité application);

– la couche application en tant que couche de niveau le plus élevé de l'OSI, ne fournit aucun service à un niveau supérieur;

– les associations formées par la communication de protocol-control-information (information de contrôle de protocole) entre des entités de la couche application ont une signification particulière.

5.2 *Processus d'application*

Un AP (processus d'application) représente un ensemble de ressources y compris les ressources de traitement, dans un système réel ouvert qui pourrait être utilisé pour réaliser des activités de traitement de l'information (le concept de l'AP est défini dans l'ISO 7498). Un AP peut nécessiter que ses interactions avec d'autres AP soient organisées en parties séparées que l'on pourrait invoquer soit simultanément, soit en série, soit individuellement.

5.3 *Entités application*

Les aspects d'un AP qui demandent à être considérés pour les besoins de l'OSI sont représentés par une ou plusieurs AE (entité application). Une AE représente un ensemble de capacités de communication OSI d'un AP particulier. Un AP sans AE ne peut pas communiquer dans l'environnement OSI, ni donc dans l'environnement DCP.

Une AE représente un et un seul AP dans l'environnement OSI. Des AP différents peuvent être représentés par des AE d'un même type (AE type). Un AP peut contenir un ensemble de AE qui ne sont pas toutes du même type.

Une AE-invocation (invocation de AE) représente une utilisation spécifique des possibilités d'une AE. Elle représente une activité spécifique d'une AP-invocation (invocation de AP). Les aspects d'une AP-invocation qui demandent d'être considérés pour les besoins de l'interconnexion des systèmes ouverts sont représentés par une ou plusieurs AE-invocation.

Il peut y avoir zéro, une ou plusieurs AE-invocation représentant une AP-invocation à un moment donné.

Une AE-invocation modélise les fonctions de communication, avec l'état de l'information associée pour une activité particulière de communication d'une AP-invocation. De telles activités s'accomplissent à l'aide de communications entre des AE-invocation reliées à des associations d'applications.

Une AE-invocation peut être successivement la partenaire de plusieurs associations d'applications. Dans DCP, au plus une association d'application peut être définie dans une AE-invocation, mais il peut y avoir des intervalles de temps où une AE-invocation ne fait pas partie d'une association d'applications. La durée de vie d'une AE-invocation n'est pas déterminée par la durée d'une association d'applications à laquelle elle participe.

---

* Les chiffres entre crochets renvoient à la bibliographie donnée dans l'annexe B.

The application layer is supported by the lower layers in OSI. In DCP, this in particular means the application layer making use of the data link services specified in future IEC 1334-4-31 [1]* and in IEC 1334-4-32.

The application layer differs from the other layers of OSI in several important respects:

– entities in the application layer are made up of a collection of application-service-elements (ASEs), each of which is defined by a set of service and protocol standards. These ASEs are combined in various ways, to form various types of AEs;

– the application layer, as the highest layer of OSI, does not provide a service to a higher layer;

– the associations formed by the communication of protocol-control-information between entities in the application layer have a particular significance.

5.2 *Application processes*

An AP represents a set of resources, including processing resources, within a real open system that may be used to perform a particular information processing activity (the AP concept is defined in ISO 7498). An AP may require that its interactions with other APs be organized into a number of distinct parts, which may be invoked either simultaneously, or serially, or both.

5.3 *Application entities*

The aspects of an AP which need to be taken into account for the purpose of OSI are represented by one or more AEs. An AE represents a set of OSI communication capabilities of a particular AP. An AP without an AE cannot communicate in the OSI environment, nor in the DCP environment.

An AE represents one, and only one, AP in the OSI environment. Different APs may be represented by AEs of the same AE type. An AP may contain a set of AEs that are not all of the same AE type.

An AE-invocation represents a specific use of the possibilities of an AE. It represents specific communication activities of an AP-invocation. The aspects of an AP-invocation that need to be taken into account for the purposes of open systems interconnection are represented by one or more AE-invocations.

There may be zero, one, or more AE-invocations representing an AP-invocation at any particular time.

An AE-invocation models the communication functions, together with the associated state information for particular communication activities of an AP-invocation. Such activities are progressed through communication between AE-invocations related by application-associations.

An AE-invocation might be, consecutively, a partner in a number of application-associations. In DCP, at most one application-association may be defined within an AE-invocation, but there may be periods of time when an AE-invocation is not a party to any application-association. The lifetime of an AE-invocation is not determined by the duration of the application-association in which it is a participant.

* Figures in square brackets refer to the bibliography given in annex B.

5.4 *Elément de service d'application (ASE)*

Un ASE est un ensemble de fonctions qui fournissent les moyens de communication OSI pour l'interfonctionnement des AE-invocation dans un but spécifique.

Les capacités de communication d'un ASE sont définis par la spécification d'un ensemble de services, d'un ensemble de APDU (unités de données de protocole application), et de la procédure régissant leur usage. Cela constitue le protocole application entre deux ASE de même genre.

Une AE peut être composée de un ou plusieurs ASE de genres différents en vue de réaliser des moyens composites spécifiques de communication dans un but particulier.

5.5 *Associations d'applications*

Une association d'applications est une relation coopérative entre deux AE-invocations en vue d'une communication d'informations et d'une coordination d'actions conjointes. Cette relation est composée d'échanges de APCI (information de contrôle de protocole application) utilisant les services data link (liaison de données).

Quand une communication entre deux AE est demandée pour satisfaire les besoins d'une application, une association d'applications est établie entre les AE-invocation des deux AE. Dans DCP, une AE-invocation ne peut supporter qu'une seule association d'applications à la fois avec une autre AE-invocation.

Un identifiant d'association d'applications peut être associé à une association d'applications. Cet identifiant est unique dans le champ de la paire d'AE-invocations associées. Il permet d'identifier l'information de l'état dans chaque AE-invocation.

La dissolution d'une association d'applications résulte des actions des AE-invocation concernées. Les AE-invocations peuvent prendre une telle décision en réponse à une erreur de communication.

L'ACSE (élément de contrôle de service d'application) est un élément de service dans la couche application qui fournit les moyens d'établir, de contrôler et de dissoudre une association d'applications. Cet élément de service est décrit dans la clause ci-dessous.

5.6 *Contexte d'application*

5.6.1 *Définition*

Pour qu'un échange d'information puisse effectivement avoir lieu dans une association d'applications, les AE-invocation homologues doivent en être mutuellement averties et ensuite un ensemble commun de règles doit régir l'échange. Cet ensemble commun de règles est dit «contexte d'application de l'association d'applications». Cet ensemble de règles couvre tout ce qui se rapporte à l'échange d'information nécessaire pour supporter les activités des AE-invocation dans l'association d'applications. Dans DCP, cela inclut tout spécialement toutes les conventions de cryptage et de décryptage utilisées entre les deux entités application.

Une association d'applications n'a qu'un seul contexte d'application. Cependant le jeu de règles qui font le contexte d'application d'une association d'applications peut contenir des règles pour altérer ce jeu de règles pendant la durée de vie de l'association d'applications.

5.4 *Application service elements*

An ASE is a set of functions that provides OSI communication capabilities for the interworking of AE-invocations for a specific purpose.

The communication capabilities of an ASE are defined by the specification of a set of services, a set of application-protocol-data-units (APDU), and the procedure governing their use. This constitutes the application-protocol between two ASEs of the same kind.

An AE may be composed of one or more ASEs of different kinds in order to realize a specific composite communication capability for a particular purpose.

5.5 *Application-associations*

An application-association is a co-operative relationship between two AE-invocations for the purpose of communication of information and co-ordination of their joint operation. This relationship is formed by the exchange of application-protocol-control-information (APCI) using the data link services.

When communication is required between two AEs to meet the needs of an application, an application-association is established between the AE-invocations of the two AEs. In DCP, an AE-invocation may support only one application-association simultaneously with an other AE-invocation.

An application-association-identifier may be associated with an application-association. This application-association-identifier is unique within the scope of the pair of associated AE-invocations. It provides the means to identify the related state information in each AE-invocation.

The termination of an application-association results from the action of the related AE-invocations. The AE-invocations may take such action in response to a failure in communications.

The application control service element (ACSE) is the service element within the application layer that provides the facilities to establish, control and terminate an application-association. This service element is described in a following clause.

5.6 *Application context*

5.6.1 *Definition*

To effectively exchange information on an application-association, the pair of AE-invocations must be mutually aware of, and follow, a common set of rules that governs the exchange. This common set of rules is called the application context to the application-association. This set of rules covers everything relevant to the exchange of information necessary to support the activities of the AE-invocations in the application-association. In DCP, this includes especially all the deciphering and deciphering conventions that are used between two application entities.

An application-association has only one application context. However, the set of rules that make up the application context of an application-association may contain rules for alteration of that set of rules during the lifetime of the application-association.

Un contexte d'application contient des règles pour décrire:

– un jeu de choses qui doivent être connues par les deux AE-invocation;
– les relations entre ces choses;
– les actions qu'on peut leur appliquer;
– les états autorisés les concernant.

Ce jeu de choses qui doivent être connues des deux AE-invocation inclut celles qui peuvent être l'objet de la communication compatible avec l'association d'applications.

NOTE – On pourra trouver des détails sur le contexte d'application dans l'ISO/IEC 9545.

5.6.2 *Procédure*

Le contexte d'application qui s'applique à une association d'applications est déterminé pendant sa création.

Un des moyens suivants peut être utilisé:

a) identifier une définition de contexte d'application préexistante;
b) transférer une description actuelle du contexte d'application.

On peut en particulier utiliser un nom pour identifier une définition de contexte d'application préexistante.

Le comportement en communication d'une AE-invocation dans une association d'applications est obligatoirement compatible avec un comportement générique défini par le contexte d'application en place.

Les ASE et la définition de leur utilisation dans le contexte d'application doivent garantir une utilisation coopérative des services data link (liaison de données).

Dans DCP, le contexte d'application est un des objets gérés (managed objects). Il est donc configuré et initialisé à l'aide des moyens de gestion qui seront décrits dans les futures CEI 1334-4-51 [2] et CEI 1334-4-52 [3]. Il est prévu qu'un contexte d'application préexiste et qu'il soit référencé par son nom pendant la création d'une association d'applications.

5.7 *Objet association unique (SAO)*

Un SAO (objet association unique) est le composant d'une AE-invocation qui modélise les fonctions et les états de l'information relatifs au fonctionnement d'une association d'application particulière. Un SAO contient un ou plusieurs ASE (l'un d'eux est toujours un ACSE) et une fonction de contrôle du SAO.

Le contexte d'application d'une association d'application particulière contient les règles de composition et d'exploitation du SAO supportant cette association d'applications dans l'AE-invocation.

5.8 *Fonction de contrôle d'association unique*

La SACF (fonction de contrôle d'association unique) est le composant du SAO qui modélise la coordination des interactions entre les ASE contenus dans le SAO et qui modélise aussi la coordination de l'usage qu'ils font des services data link (liaison de données). Les règles relatives à ces interactions sont définies dans le contexte d'application de l'association d'application.

An application context includes the rules that describe:

– a set of things that must be known by both AE-invocations;
– the relationships among those things;
– the actions which may be performed on them;
– the permitted states of affairs concerning them.

The set of things which must be known by both AE-invocations includes those which may be the subject of communications with respect to an application-association.

NOTE – Further details on the application context rules may be found in ISO/IEC 9545.

5.6.2 *Procedure*

The application context that applies to an application-association is determined during its establishment.

Either of the following ways might be used:

a) identifying a pre-existing application context definition;
b) transferring an actual description of the application context.

In particular, a name may be used to identify a pre-existing application context definition.

The communications behaviour of an AE-invocation over an application-association is constrained to be compatible with a generic behaviour defined by the application context in use.

The ASEs themselves and the definition of their use in the application context must ensure co-operative use of the data link services.

In DCP, the application context is one of the managed objects. It is then configured and initiated through the use of the management facilities, which will be described in future IEC 1334-4-51 [2] and future IEC 1334-4-52 [3]. It is intended that an application context pre-exists and is referenced by its name during the establishment of an application-association.

5.7 *Single association object*

A single association object (SAO) is the component of an AE-invocation that models the functions and state information related to the operation of an individual application-association. An SAO contains one or more ASEs (one of which is always the association control service element (ACSE)), and a single association control function.

The application context for an individual application-association contains rules for the composition and operation of the SAO supporting that application-association within the AE-invocation.

5.8 *Single association control function*

The single association control function (SACF) is the component of the SAO that models the co-ordination of the interactions among the ASEs contained in the SAO, and also models the co-ordination of their use of the data link services. The rules concerning these interactions are defined by the application context of the application-association. There is no architectural requirement for a separate SACF specification.

5.9 *Noms et fonction de répertoire*

Comme il est spécifié dans l'ISO 7498-3, les application-directory-functions (fonctions de répertoire d'applications) traitent les adresses data link (liaison de données), les AE-title ainsi que les application-control-addressing-information (information de contrôle d'adressage d'application) pour fournir une correspondance entre ces catégories d'informations. Conceptuellement, ces fonctions sont accomplies par des AE afin de déduire les informations d'adressage nécessaires.

L'information sur ces correspondances peut être conservée, soit localement et rendue disponible lors des accès par des application-directory-function (fonction de répertoire d'applications), soit à distance. C'est une responsabilité locale de retrouver l'information et de la rendre disponible à une fonction de répertoire d'applications. Si cette information est conservée à distance, on utilise les protocoles OSI pour y accéder.

NOTE 1 – Il n'est pas exigé que chaque AE contienne un ASE qui fournit le service qui retrouve les informations stockées à distance; un sytème de gestion local peut obtenir ce service d'une autre AE, voire même d'une autre AE dans un autre AP.

Les procédures qui constituent les application-directory-functions (fonctions de répertoire d'applications) sont modélisées dans une AE indépendante des ASE.

NOTE 2 – Les fonctions de répertoire d'application sont un exemple des catégories de fonctions d'application qui sont modélisées dans une AE indépendante des ASE. D'autres fonctions d'application de cette catégorie peuvent supporter des aspects de sécurité, de gestion, etc.

Dans l'ISO 7498-3, différents genres de noms sont définis en vue de rendre identifiables certains objets de la couche application.

Ces genres de noms sont les suivants:

1) application-process-title (titre de processus d'application);
2) application-entity-title (titre d'entité application);
3) application-entity-invocation-identifier (identificateur d'invocation d'entité application);
4) application-process-type-title (titre du type du processus d'application);
5) application-entity-type-title (titre du type de l'entité application);
6) system-title (titre du système).

Les façons dont ils peuvent être utilisés lors de l'exploitation des fonctions de répertoire d'applications et pour l'identification des objets spécifiques de la couche application sont décrites dans l'ISO 7498-3.

5.10 *Utilisation de l'association d'applications*

Les moyens d'établir et de dissoudre une association d'applications sont contenus dans un ASE spécifique: l'ACSE. Cet ASE est une partie indispensable de l'AE.

En établissant une association d'applications, une AE-invocation spécifie aux services data link (liaison de données) la localisation des AE homologues par leurs adresses data link. Elle utilise en plus l'identificateur d'AE-invocation pour sélectionner les AE-invocation homologues.

NOTE – Une AE-invocation peut obtenir l'adresse data link d'une AE homologue en utilisant les fonctions de répertoire d'applications pour fournir la correspondance entre le titre de l'AE et l'adresse data link, comme il est décrit dans l'ISO 7498-3.

Une association d'applications est dissoute par l'AE-invocation concernée.

5.9 *Names and directory functions*

As specified in ISO 7498-3, application-directory-functions process data link-addresses, AE-titles, and application-protocol-addressing-information to provide mapping among these categories of information. Conceptually, these functions are performed by the AE to derive the addressing information required.

Information on this mapping may be held locally and made available for access by application-directory-functions, or it may be held remotely. It is a local responsibility to retrieve the information and make it available to an application-directory-function. If this information is stored remotely, OSI protocols are used to access that information.

NOTE 1 – It is not required that every AE contain an ASE that provides the service to retrieve this remote information; local system management may obtain this service from another AE, even another AE in another AP.

The procedures which constitute the application-directory-functions are modelled within an AE independent of any particular ASE.

NOTE 2 – Application-directory-functions are an example of a category of application-functions that are modelled within the AE independent of a particular ASE. Other application-functions in this category may support aspects of security activities, management activities, etc.

In ISO 7498-3, several kinds of names are defined in order to enable the identification of certain objects in the application layer.

These kinds of names are:

1) application-process-title;
2) application-entity-title;
3) application-entity-invocation-identifier;
4) application-process-type-title;
5) application-entity-type-title;
6) system-title.

The ways in which they may be used in the operation of application-directory-functions and the identification of specific application layer objects are described in ISO 7498-3.

5.10 *Use of application-association*

Capabilities for the establishment and termination of application-associations are contained in a specific ASE: the ACSE. This ASE is a necessary part of AEs.

In establishing an application-association, an AE-invocation specifies to the data link services the location of a peer AE by its data link address. Additionally, it uses the AE-invocation identifier to the selection of a peer AE-invocation.

NOTE – An AE-invocation may obtain the data link address of a peer AE by use of application-directory-functions to provide a mapping between AE title and data link address as described in ISO 7498-3.

An application-association is terminated by the related AE-invocations.

5.11 *Utilisation des services data link (liaison de données)*

Une AE est attachée à un DL-service-access-point (point d'accès aux services liaisons de données) afin de la rendre adressable dans l'environnement OSI.

Des AE-invocations qui communiquent utilisent les services data link (liaison de données) pour transférer des APDU entre elles. La méthode d'utilisation des services data link est prescrite par les règles du contexte d'application de l'association d'applications.

5.12 *Normes ASE (élément de service application)*

Chaque ASE est défini en terme de services et de protocole.

Pour certains ASE, il est souhaitable d'établir un modèle expliquant les exigences des utilisateurs du service. Un tel modèle peut inclure des références à des modèles plus généraux. Leurs descriptions doivent rester conceptuelles et incorporer dans l'OSI l'image de leurs opérations réelles. Il n'est pas possible de dériver des exigences de conformité de tels modèles.

Une définition de service permet de comprendre les fonctions composant l'ASE. C'est le premier pas qui conduit à la spécification du protocole correspondant. Par analogie avec les définitions de service aux limites des couches de l'OSI, les définitions de service pour les ASE sont conceptuelles et n'impliquent pas de conformité.

Une spécification de protocole définit les structures de l'échange d'information entre des ASE homologues. Ce faisant, la spécification de protocole peut utiliser les services fournis par n'importe lequel des autres ASE des services data link.

5.13 *Définition de la syntaxe abstraite*

Une syntaxe abstraite est faite des aspects des règles utilisées dans la spécification formelle des données qui sont indépendantes des techniques de codage de la représentation des données.

Pour un ASE donné, la structure des APDU est spécifiée par un ensemble de une ou plusieurs syntaxes abstraites. La structure de toute information utilisateur véhiculée dans ces ASE d'une association est spécifiée par un autre ensemble de une ou plusieurs syntaxes abstraites.

Un nom de syntaxe abstraite est associé à la définition du jeu de APDU ou à la définition du jeu d'information utilisateur.

Toutes les sections à venir de la norme DCP font usage de l'ISO/IEC 8824 (ASN.1). Toutes les normes futures qui feront usage de la spécification DCP doivent aussi se référer à ASN.1.

5.14 *Règles de codage*

Parce qu'il n'y a pas de couche présentation dans DCP, le codage des éléments d'information doit être accompli par le processus d'application. Afin de garantir la compatibilité complète avec ASN.1, d'augmenter l'interopérabilité et de permettre d'avoir des données transmises relativement compactes, les règles BER (Basic Encoding Rules – règles de codage de base) peuvent être utilisées, mais d'autres règles d'encodage peuvent être définies dans des normes d'accompagnement. BER est une norme internationale enregistrée en tant que ISO/IEC 8825.

5.11 *Use of the data link services*

An AE is attached to one DL-service-access-point in order to make it addressable in the OSI environment.

The communicating AE-invocations use the data link services to transfer APDUs between each other. The method of use of the data link services is prescribed by the rules of the application context of an application-association.

5.12 *Application service element standards*

Each ASE is defined in terms of service and protocol.

For some ASEs, it is desirable to establish a model explaining the requirements of the service user. Such a model may include reference to more general models. Their descriptions must remain conceptual, carrying the appearance within OSI of their real operation. No implementation conformance requirements can be derived from such models.

A service definition conveys the understanding of the function carried out by the ASE. It is the first step that leads to the specification of the corresponding protocol. By analogy with the service definitions at the OSI layer boundaries, the service definition for ASEs are conceptual and do not imply conformance.

A protocol specification defines the structure for the exchange of information between peer ASEs. In so doing, the protocol specification may make use of the services provided by any other ASE or of the data link services.

5.13 *Abstract syntax definition*

An abstract syntax is made up of those aspects of the rules used in the formal specification of data which are independent of the encoding techniques to represent the data.

For a given ASE, the structure of the APDUs is specified by a set of one or more abstract syntaxes. The structure of any user information conveyed within these ASEs on an association is specified by another set of one or more abstract syntaxes.

An abstract syntax name is associated with the definition of a set of APDUs or with the definition of a set of user information.

All the following sections of the DCP standard make use of ISO/IEC 8824 (ASN.1). All further standards that make use of the DCP specification shall also refer to ASN.1.

5.14 *Encoding rules*

Because no presentation layer is available in DCP, the encoding of the information elements must be achieved by the application process. In order to ensure full compatibility with ASN.1, to increase interoperability, and to allow a relative compactness of the transmitted data, the basic encoding rules (BER) could be used, but other encoding rules could be defined in a companion standard. BER is an international standard registered ISO/IEC 8825.

5.15 *Restrictions apportées par DCP*

Afin de simplifier dans une large mesure la spécification de la couche application dans DCP, l'utilisation de l'association d'applications entre deux AE invocation a fait l'objet d'une restriction. Dans DCP, une AE-invocation peut contenir au plus une association d'applications en même temps, alors que plusieurs AE-invocation peuvent être actuellement définies pour la même entité application.

Quand une association d'applications est définie entre deux AE-invocation, ces AE-invocation (ainsi que l'association d'applications) sont identifiées sans ambiguïtés par l'ensemble composé de leurs adresses data link (L_SAP et Physical_Attachment). Dans DCP, il n'est donc nul besoin de fournir un identificateur spécial pour spécifier l'accès à une AE-invocation particulière. Faire référence à l'ensemble des quatre éléments (SAP destination, SAP origine, adresse destination, adresse origine) fournis par la couche liaison de données est préférable.

A la réception d'une indication DL_Data, la couche application

– calcule le Calling Title (titre appelant) à l'aide de la fonction A_Directory à partir de la Source_SAP (SAP origine) et de la Source_address (adresse origine);

– calcule le Called Title (titre appelé) à l'aide de la fonction A_Directory à partir de la Destination_SAP (SAP destination) et de la Destination_address (adresse destination);

– identifie l'AE-invocation destinataire à l'aide du couple (Calling Title, Called Title).

5.16 *Besoins d'enregistrement*

L'utilisation des normes de la couche application exige la mise en place de procédures d'enregistrement pour l'assignation des noms (qui sont sans ambiguïtés dans l'environnement OSI) pour les objets suivants:

a) application context (contexte d'application), identifié par un seul nom de contexte d'application;

b) AE (entité d'application), identifié dans DCP par un type d'entité d'application unique et par le nom d'un processus d'application;

c) processus d'application, identifié par un nom unique de processus d'application.

Dans DCP, un seul type d'entité application est prédéfini, associé à la DLMS application entity (entité application DLMS).

Dans DCP, une structure des noms des processus d'application sera proposée dans la future CEI 1334-4-52 [3].

Une définition d'un contexte d'application, ou un type d'entité application, ou un nom de processus d'application peuvent être, lorsqu'ils sont enregistrés, soit des normes nationales ou internationales, soit des définitions développées et publiées par une communauté d'intérêts, soit des accords privés.

Les procédures d'enregistrement utilisées dans chacune de ces situations doivent être conformes à un cadre reconnu au niveau international pour les procédures d'enregistrement.

6 Elément de service de contrôle d'association (ACSE)

Afin de supporter une association d'applications entre deux entités application, la spécification DCP utilise l'ACSE décrit dans l'amendement 2 de l'ISO 8649. Cette norme internationale décrit les définitions de services pour ACSE en mode sans connexion. Il s'applique à la spécification DCP bien que il n'y ait pas de service présentation disponible. Le présent article décrit les services ACSE tels qu'ils sont demandés dans DCP.

5.15 *DCP restrictions*

In order to greatly simplify the specification of the DCP application layer, a restriction on the use of the application-association between two AE-invocations is made. In DCP, an AE-invocation may contain at most one application-association at the same time, although many AE-invocations may be defined concurrently for an application-entity.

When an application-association is defined between two AE-invocations, these AE-invocations (and the application-association) are unambiguously identified by the set formed with both their data link addresses (L_SAP and Physical_Attachment). Within DCP, there is then no need to provide a special identifier to specify access to a particular AE-invocation. Reference to the four-elements set (Destination_SAP, Source_SAP, Destination_address, Source_address) as provided by the data link layer is more convenient.

On receipt of a DL_Data indication, the application layer:

– computes the Calling Title with the A_Directory function from the Source_SAP and the Source_address;

– computes the Called Title with the A_Directory function from the Destination_SAP and the Destination_address;

– identifies the destination AE-invocation by the couple (Calling Title, Called Title).

5.16 *Registration requirements*

The use of application layer standards requires the establishment of registration procedures for the assignment of names (which are unambiguous throughout the OSI environment) for the following objects:

a) application context, identified by an unique application context name;

b) application-entities, identified in DCP by an unique application entity type and the application process name;

c) application-processes identified by an unique application process name.

In DCP, only one application entity type is predefined associated to the DLMS application entity.

In DCP, an application process name structure will be proposed in future IEC 1334-4-52 [3].

An application context definition, an application entity type or an application process name that is registered may be an international or national standard, a published definition developed by a community of interest, or a private agreement.

The registration procedures used in each of these situations shall be compatible with an internationally recognized framework for registration procedures.

6 Association control service element (ACSE)

In order to support the application-association between two application entities, the DCP specification makes use of the ACSE described in amendment 2 of ISO 8649. That international standard describes the service definition for ACSE in connectionless mode. It applies to the DCP specification, although no presentation service is available. This clause describes the ACSE service as it is required in the DCP specification.

6.1 *Définitions spécifiques à ACSE*

6.1.1 *Association control service element (ACSE)*

ACSE (élément de service de contrôle d'association) est un élément de service d'application (ASE) particulier décrit dans cet article. Il fournit dans la couche applications tous les moyens d'établir, de contrôler et de dissoudre une association d'application.

6.1.2 *ACSE service-user (utilisateur des services de ACSE)*

L'ACSE service-user (utilisateur des services de ACSE) est la partie de l'entité application qui utilise les services de ACSE.

6.1.3 *ACSE service-provider (prestataire de service ACSE)*

L'ACSE service-provider (prestataire de service ACSE) est une abstraction de la totalité de ces entités qui fournissent les services ACSE aux utilisateurs homologues des services ACSE.

6.1.4 *Requestor (demandeur)*

Le requestor est l'utilisateur de service qui émet la primitive request (demande) pour un service ACSE particulier.

6.1.5 *Acceptor (receveur)*

L'acceptor est l'utilisateur de service qui reçoit la primitive indication pour un service ACSE particulier.

6.1.6 *Abréviations*

APDU: application protocol data unit (unité de données de protocole);

ACPM: application control protocol machine (machine de protocole de contrôle d'application);

ACSE: association control service élément (élément de service de contrôle d'association).

6.2 *Concepts fondamentaux*

Le modèle de référence (ISO 7498) représente la communication entre une paire de processus d'application (AP) en termes de communication entre leurs entités application (AE) utilisant les services de la couche soujacente. Une fonctionnalité d'une AE est divisée en plusieurs ASE (éléments de service d'application). L'interaction entre les AE est décrite en termes d'utilisation de services de leurs ASE.

La présente section de la CEI 1334-4 introduit des concepts supplémentaires de modélisation de l'association d'applications et du contexte d'application.

Une association d'applications est une relation coopérative entre deux AE. Elle fournit le cadre de référence nécessaire entre les deux AE afin qu'il puisse y avoir effectivement interopérabilité. Cette relation est constituée par la communication d'information de contrôle de protocole application (A_PCI) entre les entités application à l'aide de leurs services data link (liaison de données).

Un contexte d'application est un ensemble explicitement identifié de ASE (éléments de service d'application), d'options possibles, de règles et de toutes autres informations nécessaires à l'interopérabilité des entités application d'une association d'applications.

6.1 *ACSE specific definitions*

6.1.1 *Association control service element (ACSE)*

The association control service element is the particular application-service-element (ASE) described in this clause. It provides all the facilities within an application layer to establish, control and terminate an application-association.

6.1.2 *ACSE service-user*

The ACSE service-user is the part of the application-entity that makes use of ACSE services.

6.1.3 *ACSE service-provider*

The ACSE service-provider is an abstraction of the totality of those entities which provide ACSE services to peer ACSE service-users.

6.1.4 *Requestor*

The requestor is the service-user that issues the request primitive for a particular ACSE service.

6.1.5 *Acceptor*

The acceptor is the ACSE service-user that receives the indication primitive for a particular ACSE service.

6.1.6 *Abbreviations*

APDU : application protocol data unit;

ACPM : application control protocol machine;

ACSE : association control service element.

6.2 *Basic concepts*

The reference model (ISO 7498) represents communication between a pair of application-processes (APs) in terms of communication between their application-entities (AEs) using the services of the underlying layer. The functionality of an AE is factored into a number of application-service-elements (ASEs). The interaction between AEs is described in terms of the use of their ASEs services.

This section of IEC 1334-4 introduces the additional modelling concepts of application-association and application context.

An application-association is a co-operative relationship between two AEs. It provides the necessary frame of reference between the AEs in order that they may interwork effectively. This relationship is formed by the communication of application-protocol-control-information (A_PCI) between the application-entities through their use of data link services.

An application context is an explicitly identified set of application-service-elements, related options, rules and any other necessary information for the interworking of application-entities on an application-association.

Le service A_Unit_Data fournit un service pour le transfert d'information entre AEI utilisant les services data link (liaison de données) en mode sans connection.

6.3 *Relations avec les autres ASE*

ACSE est censé être utilisé avec d'autres ASE afin de supporter une tâche spécifique de traitement de l'information. On s'attend donc que ACSE soit inclus dans la spécification de tous les contextes d'application.

Les assemblages des ACSE et d'autres ASE dans un contexte d'application sont indispensables pour utiliser les facilités des services data link de manière coordonnée.

6.4 *Services ACSE en mode sans connexion*

6.4.1 *Objet*

La norme ACSE fournit un service unique pour le mode sans connexion: le service A_Unit_Data. Le service A_Unit_Data sert à transférer l'information entre des AEI utilisant les services data-link en mode sans connection. C'est un service sans confirmation.

6.4.2 *Le service A_Unit_Data (accepte une unité de données)*

Le tableau 1 donne la liste des paramètres du service A_Unit_Data. Sont définis en plus des groupes de paramètres pouvant être référencés par d'autres ASE.

Tableau 1 – Le service A_Unit _Data

A_Unit_Data Service	Req	Ind
Application Context Name	M	M(=)
User Information	M	M(=)
Calling AP Title	U	C(=)
Calling AE Qualifier	U	C(=)
Calling AP Invocation Identifier	U	C(=)
Calling AE Invocation Identifier	U	C(=)
Called AP Title	U	C(=)
Called AE Qualifier	U	C(=)
Called AP Invocation Identifier	U	C(=)
Called AE Invocation Identifier	U	C(=)
Quality of Service	U	C(=)
NOTES 1 Le Calling AE Title (titre de l'AE appelant) est un combiné des paramètres Calling AP Title (titre de l'AP appelant) et Calling AE Qualifier (modificateur de l'AE appelant). 2 Le Called AE Title (titre de l'AE appelée) est un combiné des paramètres Called AP Title (titre de l'AP appelé) et Called AE Qualifier (modificateur de l'AE appelée). 3 Les deux composants de l'AE-Title (titre de l'AE): AP Title (titre de l'AP) et AE Qualifier (modificateur de l'AE) sont définis dans l'ISO 7498-3.		

The A_Unit_Data service provides a service for information transfer between AEIs using the connectionless mode data link services.

6.3 *Relationship with other ASEs*

The ACSE is intended to be used with other ASEs in order to support a specific information processing task. Therefore, it is expected that an ACSE will be included in all application context specifications.

The collection of the ACSE and other ASE(s) included in an application context are required in order to use the facilities of the data link services in a co-ordinated manner.

6.4 *Connectionless-mode ACSE service*

6.4.1 *Purpose*

The ACSE standard provides a unique service for the connectionless mode: the A_Unit_Data service. The A_Unit_Data service is used to transfer information between AEIs using connectionless mode data link services. It is a non-confirmed service.

6.4.2 *A_Unit_Data service*

Table 1 lists the A_Unit_Data service parameters. In addition, groups of parameters are defined to be referenced by other ASEs.

Table 1 – The A_Unit_Data service

A_Unit_Data Service	Req	Ind
Application Context Name	M	M(=)
User Information	M	M(=)
Calling AP Title	U	C(=)
Calling AE Qualifier	U	C(=)
Calling AP Invocation Identifier	U	C(=)
Calling AE Invocation Identifier	U	C(=)
Called AP Title	U	C(=)
Called AE Qualifier	U	C(=)
Called AP Invocation Identifier	U	C(=)
Called AE Invocation Identifier	U	C(=)
Quality of Service	U	C(=)

NOTES

1 The Calling AE Title is the composite of the Calling AP Title and the Calling AE Qualifier parameters.

2 The Called AE Title is the composite of the Called AP Title and the Called AE Qualifier parameters.

3 The two components of the AE Title (AP Title and AE Qualifier) are defined in ISO 7498-3.

6.4.3 *Paramètres*

Le paramètre Application Context Name (nom du contexte d'application) identifie le contexte d'application nommé qui sera utilisé sur proposition du requestor.

Le paramètre User Information (information utilisateur) est obligatoire. La signification de ce paramètre dépend du contexte d'application qui accompagne la primitive. Ce paramètre sert aux ASE utilisateurs à construire leurs configurations au sein de l'association d'applications.

Le paramètre Calling AP Title (titre de l'AP appelant) identifie l'AP particulier qui contient le requestor du service A_Unit_Data.

Le paramètre Calling AE Qualifier (modificateur de l'AE appelant) identifie l'AE particulière de l'AP qui contient le requestor du service A_Unit_Data.

Le paramètre Calling AP Invocation-Identifier (identificateur d'invocation de l'AP appelant) identifie l'invocation d'AP qui contient le requestor du service A_Unit_Data.

Le paramètre Calling AE Invocation-Identifier (identificateur d'invocation d'AE) identifie l'invocation d'AE qui contient le requestor du service A_Unit_Data.

Le paramètre Called AP Title (titre de l'AP appelé) identifie l'AP qui contient l'acceptor envisagé du service A_Unit_Data.

Le paramètre Called AE Qualifier (modificateur de l'AE appelée) identifie l'AE particulière de l'AP qui contient l'acceptor envisagé du service A_Unit_Data.

Le paramètre Called AE Invocation-Identifier (identificateur d'invocation de l'AE appelée) identifie l'invocation d'AE qui contient l'acceptor envisagé du service A_Unit_Data.

Le paramètre Called AP Invocation-Identifier (identificateur d'invocation d'AP) identifie l'invocation d'AP qui contient l'acceptor envisagé du service A_Unit_Data.

Le paramètre Quality of Service (qualité du service) spécifie la qualité du service que les AE-invocation doivent fournir au service. Ce paramètre contient la durée maximale exprimée en 1/10 s nécessaire pour accomplir le service soumis.

6.4.4 *Procédure du service A-Unit-Data*

La procédure A_Unit_Data est directement apparentée à celle qui est définie pour le service DL_Data.

Le requestor émet une primitive request (demande) de A_Unit_Data. L'AEI appelé est identifié par les paramètres de la primitive request. L'usage du service A_Unit_Data est restreint à l'exploitation en mode sans connexion.

Le prestataire de service ACSE émet une primitive indication A_Unit_Data vers l'acceptor.

L'acceptor accepte l'indication de A_Unit_Data. Il ne renvoie pas de primitive réponse.

6.4.5 *A_Unit_Data: caratéristiques*

Le service A_Unit_Data est un service non confirmé. Le service A_Unit_Data n'interrompt aucun autre service. Le service A_Unit_Data n'est pas interrompu par un autre service.

6.4.3 *Parameters*

The Application Context Name parameter identifies the application context to be used, nominated by the requestor.

The User Information parameter is mandatory. The meaning of this parameter depends on the application context that accompanies the primitive. This parameter is used by the user ASEs for achieving their configuration within the application-association.

The Calling AP Title parameter identifies the AP that contains the requestor of the A_Unit_Data service.

The Calling AE Qualifier parameter identifies the particular AE of the AP that contains the requestor of the A_Unit_Data service.

The Calling AP Invocation-Identifier parameter identifies the AP invocation that contains the requestor of the A_Unit_Data service.

The Calling AE Invocation-Identifier parameter identifies the AE invocation that contains the requestor of the A_Unit_Data service.

The Called AP Title parameter identifies the AP that contains the intended acceptor of the A_Unit_data service.

The Called AE Qualifier parameter identifies the particular AE of the AP that contains the intended acceptor of the A_Unit_Data service.

The Called AE Invocation-Identifier parameter identifies the AE invocation that contains the intended acceptor of the A_Unit_Data service.

The Called AP Invocation-Identifier parameter identifies the AP invocation that contains the intended acceptor of the A_Unit_Data service.

The Quality of Service parameter specifies the quality of service that the AE-invocation must provide for its service. This parameter contains the maximum amount of time, expressed in 1/10 s, required to process a submitted service.

6.4.4 *A_Unit_Data procedure*

The A_Unit_Data procedure is directly related to that defined for DL_Data service.

The requestor issues an A_Unit_Data request primitive. The called AEI is identified by parameters of the request primitive. Use of the A_Unit_Data service is restricted to connectionless operation.

The ACSE service-provider issues an A_Unit_Data indication primitive to the acceptor.

The acceptor accepts the A_Unit_Data indication. No response primitive is returned.

6.4.5 *A_Unit_Data: characteristics*

The A_Unit_Data service is a non-confirmed service. The A_Unit_Data service does not disrupt any services. The A_Unit_Data service is not disrupted by any services.

L'émission simultanée de l'une vers l'autre de demandes de service A_Unit_Data par deux AEI aboutit à l'acceptation des deux primitives indication de A_Unit_Data. Il n'en résulte pas une situation de collision.

6.5 *Protocole ACSE en mode sans connexion*

6.5.1 *Généralités sur le protocole*

Cette spécification de ACSE utilise le service DL_Data en mode sans connexion pour passer l'information sous la forme d'un AUDT APDU, entre des entités application homologues. Le protocole machine A_Unit_Data (c'est aussi un ACPM) communique avec son utilisateur de service au moyen de primitives A_Unit_Data définies dans le paragraphe ci-dessus. L'ACPM est conduit par la réception de request (demande) de A_Unit_Data et d'indication de DL_Data.

Durant une occurence de communication, on s'attend à la fois à l'existence des AE-invocation émettrice et réceptrice. Comment ces AEI sont créées n'est pas du ressort de ce paragraphe.

NOTE – Une solution possible pour la création d'AE-invocation pour une entité application définie est de considérer qu'on a toujours un numéro d'AE-invocation créé mais pas encore utilisé. Il est alors possible de se référer à une AE-invocation alors qu'elle n'est pas encore configurée.

6.5.2 *Eléments de procédure*

Le protocole A_Unit_Data est constitué de la procédure de transfert A_Unit_Data. La procédure de transfert A_Unit_Data est utilisée pour transmettre une unité d'information d'une AEI à une autre AEI. Elle supporte le service A_Unit_Data et utilise l'APDU A_Unit_Data (AUDT). L'ACPM émetteur compose un AUDT APDU en utilisant les valeurs des paramètres de la primitive request (demande) de A_Unit_Data. Elle émet ensuite une primitive request (demande) DL_Data.

Les paramètres de la demande de DL_Data sont

– la Destination_LSAP (LSAP destinataire), le Source_LSAP (LSAP origine), le Destination_address (adresse destinataire) traités par les fonctions de répertoire d'applications en utilisant les paramètres associés à l'AE-invocation locale;

– le L_SDU directement appliqué sur le AUDT APDU.

L'ACPM récepteur reçoit l'AUDT APDU comme des données utilisateur d'une primitive indication de DL_Data. Si des paramètres de la primitive indication de DL_Data ou les champs de AUDT APDU sont inacceptables par cet ACPM, il écarte cet AUDT APDU.

Les paramètres de la primitive indication de A_Unit_Data sont traités comme suit:

– le Calling AP Title (titre de l'AP appelant), le Qualifier (modificateur) et le Invocation identifier (identificateur d'invocation) de l'AE sont extraits, s'il est présent, de l'AUDT APDU. Autrement, ces paramètres sont traités par les fonctions de répertoire d'application à partir des paramètres de la primitive indication de DL_Data;

– le paramètre User Information (information utilisateur) est extrait des AUDT APDU et transmis à l'ASE approprié comme il est spécifié dans le contexte d'application.

The simultaneous issuing of A_Unit_Data request from two AEIs to each other results in the acceptance of both A_Unit_Data indication primitives. No collision situation results.

6.5 *Connectionless-mode ACSE protocol*

6.5.1 *Overview of protocol*

This specification of the ACSE uses the DL_Data connectionless service to pass information in the form of an AUDT APDU, between peer application entities. The A_Unit_Data protocol machine (this is the ACPM too) communicates with its service user by means of A_Unit_Data primitives defined in the subclause above. The ACPM is driven by the reception of A_Unit_Data request and DL_Data indication.

During an instance of communication, the existence of both the sending and the receiving AE-invocations is presumed. How these AEIs are created is outside the scope of this subclause.

NOTE – A possible solution of the creation of AE-invocations for a defined application entity is to consider that a number of AE-invocations are always created but not yet used. It is then possible to refer to an AE-invocation although it is not yet configured.

6.5.2 *Elements of procedure*

The A_Unit_Data protocol consists of the A_Unit_Data transfer procedure. The A_Unit_Data transfer procedure is used to transmit a unit of information from one AEI to another AEI. It supports the A_Unit_Data service and uses the A_Unit_Data (AUDT) APDU. The sending ACPM forms an AUDT APDU using parameters values from the A_Unit_Data request primitive. It then issues a DL_Data request primitive.

The parameters of the DL_Data request are:

– the Destination_LSAP, the Source_LSAP and the Destination_address computed with the application directory function using the parameters associated to the local AE-invocation;

– the L_SDU directly mapped on the AUDT APDU.

The receiving ACPM receives the AUDT APDU as user data on a DL_Data indication primitive. If any of the parameters of the DL_Data indication primitive or the fields of the AUDT APDU are unacceptable to this ACPM, it discards the AUDT APDU.

The parameters of the A_Unit_Data indication primitive are computed as follows:

– the Calling AP Title, the AE Qualifier and the AE Invocation Identifier are extracted, if present, from the AUDT APDU. Otherwise, these parameters are computed with the application directory function from the DL_Data indication parameters;

– the User Information parameter is extracted from the AUDT APDU and passed to the appropriate ASE as specified in the application context.

6.5.3 *A_Data_Unit APDU*

Le présent paragraphe décrit le AUDT APDU à l'aide de ASN.1.

```
DCP-ACSE DEFINITIONS IMPLICIT TAGS::= BEGIN

AUDT-APDU::=    [15]    SEQUENCE {
        application-context-name            OBJECT IDENTIFIER
        user-information                    ANY,
        quality-of-service                  [1] Integer             OPTIONAL,
        calling-AP-title                    [2] VisibleString       OPTIONAL,
        calling-AE-qualifier                [3] VisibleString       OPTIONAL,
        calling-AP-invocation-identifier    [4] Integer             OPTIONAL,
        calling-AE-invocation-identifier    [5] Integer             OPTIONAL,
        called-AP-title                     [6] VisibleString       OPTIONAL,
        called-AE-qualifier                 [7] VisibleString       OPTIONAL,
        called-AP-invocation-identifier     [8] Integer             OPTIONAL,
        called-AE-invocation-identifier     [9] Integer             OPTIONAL
        }
Integer::= INTEGER (0..127)
END.
```

6.5.4 *Restrictions dans DCP*

6.5.4.1 *Utilisation de ACSE par DCP*

DCP a défini quelques restrictions à l'usage de ACSE avec les ASE de DLMS. La procédure de service de DLMS définit pour ses propres besoins un DLMS-context (contexte DLMS). Plus spécialement, ce contexte contient le contexte d'application négocié par la procédure de service ACSE.

L'usage de ce contexte d'application sans aucune modification n'exige pas l'usage du service ACSE pour toutes les transmissions. Le service A_Unit_Data est alors résrvé à la notification des modifications du contexte ou à sa mise en place.

6.5.4.2 *Paramètres soumis par AUD*

Selon les remarques de 5.15, les paramètres titres du service A_Unit_Data ne sont pas soumis avec la demande. En particulier, le DLMS-ASE n'utilise que les paramètres obligatoires et le paramètre quality of service (qualité du service).

6.5.4.3 *Nom du contexte par défaut dans DCP*

DCP définit son nom de contexte par défaut par l'identificateur d'objet suivant:

default_DCP_Application_Context_Name::= { 3 2 18 19 67 }

Le contexte d'application par défaut de DCP contient la méthode de cryptage (algorithme et règles internes d'acceptation) ainsi que la clé globale de cryptage (définie dans l'article 7 suivant). Le contexte d'application par défaut de DCP est décrit dans l'annexe A.

6.5.3 *A_Data_Unit APDU*

This subclause describes the AUDT APDU using ASN.1.

```
DCP-ACSE DEFINITIONS IMPLICIT TAGS::= BEGIN

AUDT-APDU::=    [15]    SEQUENCE {
    application-context-name            OBJECT IDENTIFIER
    user-information                    ANY,
    quality-of-service                  [1] Integer             OPTIONAL,
    calling-AP-title                    [2] VisibleString       OPTIONAL,
    calling-AE-qualifier                [3] VisibleString       OPTIONAL,
    calling-AP-invocation-identifier    [4] Integer             OPTIONAL,
    calling-AE-invocation-identifier    [5] Integer             OPTIONAL,
    called-AP-title                     [6] VisibleString       OPTIONAL,
    called-AE-qualifier                 [7] VisibleString       OPTIONAL,
    called-AP-invocation-identifier     [8] Integer             OPTIONAL,
    called-AE-invocation-identifier     [9] Integer             OPTIONAL
    }

Integer::= INTEGER (0..127)

END.
```

6.5.4 *DCP restrictions*

6.5.4.1 *DCP use of the ACSE*

DCP defines some restrictions on the use of the ACSE with the DLMS ASEs. The DLMS service procedure defines for its own purpose a DLMS-context. Especially, this context contains the application context negotiated by the ACSE service procedure.

The use of the application context without any modification does not require the use of the ACSE service for all transmissions. The A_Unit_Data service is then reserved for the notification of a change in the application context, or for its establishment.

6.5.4.2 *AUD submitted parameters*

Considering the remarks in 5.15, the titles parameters of the A_Unit_Data service are not submitted with the request. In particular, the DLMS-ASE make only use of the mandatory parameters and of the quality of service parameter.

6.5.4.3 *DCP default context name*

DCP defines its default application context name by the following object identifier:

default_DCP_Application_Context_Name::= { 3 2 18 19 67 }

The default DCP application context contains the ciphering method (algorithm and internal acceptance rules) and the global ciphering key (defined in clause 7). The default DCP application context is described in annex A.

7 Entité application de DLMS

7.1 *Généralités*

DLMS (Spécification de Message de Ligne de Distribution) est une spécification de la couche application étudiée pour supporter une communication de messages de et vers un équipement de distribution.

Le présent article décrit la structure générale d'une entité application de DLMS. Cette description utilise les concepts introduits dans les articles.

Pour les termes ci-dessous, se référer à l'article 5 :

– application association (association d'applications);
– application context (contexte d'application);
– association control service element (élément de service de contrôle d'association (ACSE));
– application entity (entité application);
– application entity invocation (invocation d'entité application);
– application process (processus d'application);
– application protocol control information (information de contrôle de protocole application);
– application protocol data unit (unité de données de protocole application);
– application service element (élément de service d'application).

L'entité application de DLMS est une entité application dédiée au support des services de DLMS. Le présent article décrit le contenu de l'entité application en termes de ASE (éléments de services d'application), d'accès aux fonctions d'application, et de définition du contenu des contextes.

7.2 *Structure générale*

7.2.1 *Contenu de l'entité application de DLMS*

L'entité application de DLMS est définie de façon à être la seule entité application supportant les processus d'application DLMS dans DCP. Il doit y avoir une AE-invocation (invocation d'entité application) pour chaque association d'applications définie dans le processus d'application. Ces AE-invocation sont des *instanciations* des AE de DLMS.

Une DLMS application entity invocation (invocation d'entité application de DLMS) contient:

– l'ACSE;
– l'ASE du client DLMS (C-DASE), ou l'ASE du serveur DLMS (S-DASE), ou les deux;
– associé à l'ACSE, un contexte d'application;
– un DLMS-Context (contexte DLMS);
– diverses fonctions d'application.

Si l'utilisateur DLMS agit en tant que serveur DLMS, son AE-invocation (invocation d'entité application) doit contenir un DASE serveur. Si l'utilisateur DLMS agit en tant que client, son AE-invocation doit contenir un DASE client.

NOTE – Le contenu de l'AE de DLMS n'est pas restreint à la liste proposée ci-dessus, mais les autres objets contenus dans l'AE de DLMS ne sont pas spécifiés.

7 DLMS application entity

7.1 *Overview*

The distribution line message specification (DLMS) is an application layer specification designed to support messaging communications to and from distribution devices.

This clause describes the general structure of a DLMS application entity. This description makes use of the concepts introduced in the previous clauses.

Refer especially to clause 5 for the following terms:

- application-association;
- application context;
- association control service element;
- application entity;
- application entity invocation;
- application process;
- application protocol control information;
- application protocol data unit;
- application service element.

The DLMS application entity is an application entity dedicated to support the distribution line message specification services. This clause describes the contents of the application entity in terms of application service elements (ASE), access to application functions, and definition of the context contents.

7.2 *General structure*

7.2.1 *DLMS application entity contents*

The DLMS application entity is defined as the only application entity supporting the DLMS application process within DCP. There must be one AE-invocation for each application-association defined within the application process. These AE-invocations are instances of the DLMS AE.

A DLMS application entity invocation contains:

- the association control service element;
- the client DLMS ASE (C-DASE), or the server DLMS ASE (S-DASE), or both;
- associated to the ACSE, an application context;
- a DLMS-Context;
- various application functions.

If the DLMS user acts as a DLMS server, its application entity invocation must contain a server DASE. If the DLMS user acts as a DLMS client, its application entity invocation must contain a client DASE.

NOTE – The contents of the DLMS AE are not restricted to the list proposed above, but the other objects contained within the DLMS AE are not specified.

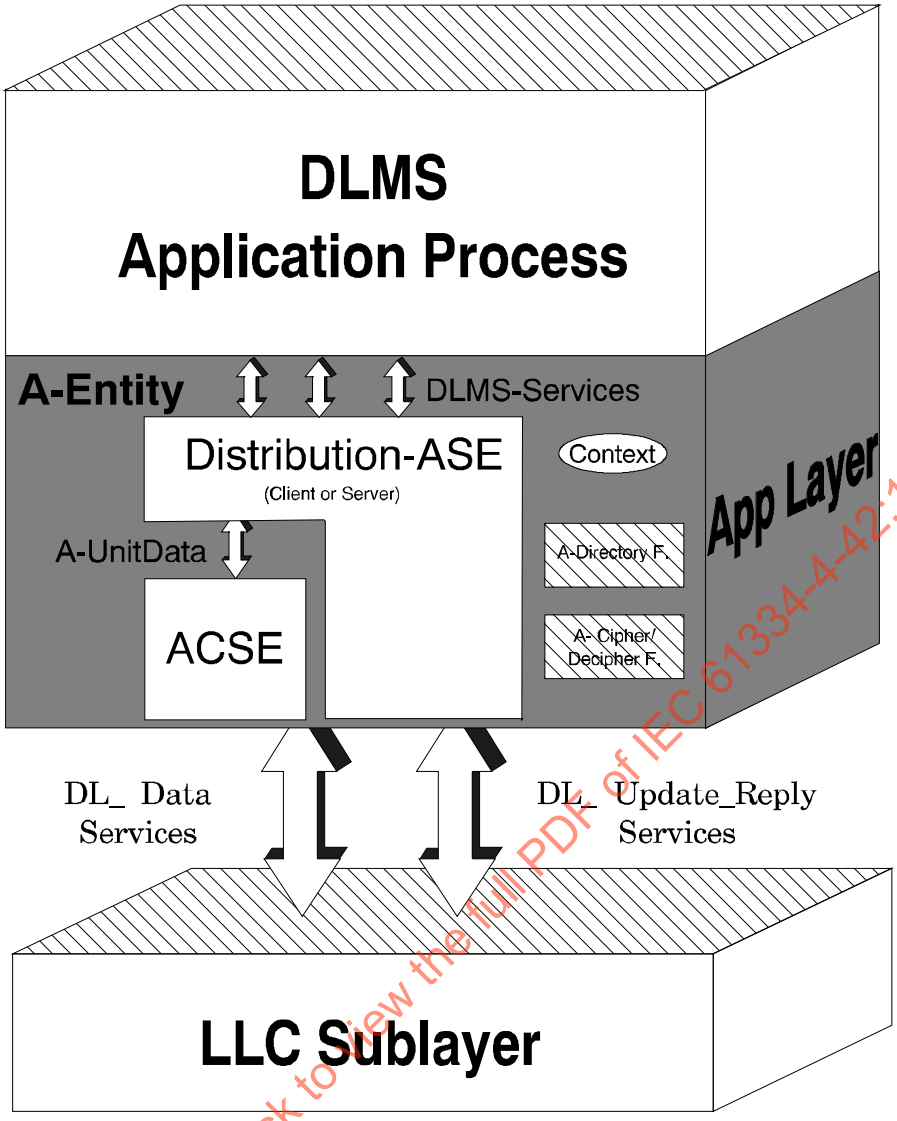

Figure 1 – Structure d'un processus d'application

7.2.2 *ACSE*

L'AE de DLMS utilise l'ACSE en mode sans connexion. Les services de ACSE sont décrits plus hauts. Le service A_Unit_Data est utilisé pour établir une association d'applications entre deux invocations d'AE de DLMS. L'application d'applications est supporté par un partage d'informations de communication communes. Ces informations sont contenues dans le contexte d'application.

7.2.3 *Les ASE de DLMS*

En vue de simplifier la description de l'entité application de DLMS, l'ASE (élément de service d'application), l'ASE support principal de l'entité application est divisé en deux parties, l'une spécifiant le comportement du client DLMS et l'autre celui du serveur DLMS. Une AE de DLMS peut contenir un seul ou les deux ASE, cela dépend du rôle joué, client ou serveur, ou les deux. Pour la clarté de cette spécification, on va maintenant supposer que, lors de la communication entre deux AE-invocation, l'une agit en tant que client et l'autre en tant que serveur. La première contient alors un DASE client et la seconde un DASE serveur.

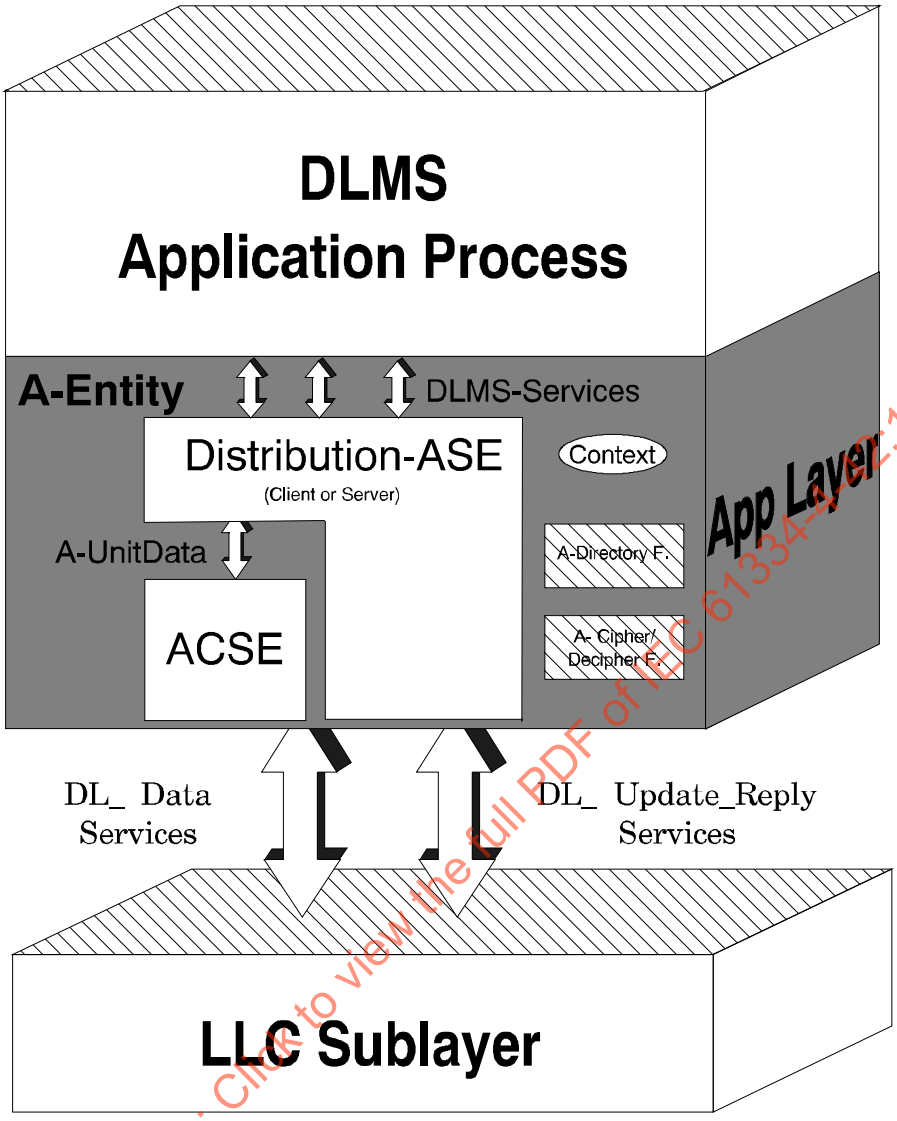

Figure 1 – Application process structure

7.2.2 *ACSE*

The DLMS AE makes use of the ACSE in connectionless mode. The ACSE services are described above. The A_Unit_Data service is used to establish an application-association between two DLMS AE invocations. The application-association is supported by sharing common communication information. This information is contained in the application context.

7.2.3 *DLMS ASEs*

In order to simplify the description of the DLMS application entity, the application service element that mainly supports the application entity is divided in two parts, one for specifying the client DLMS behaviour, and one for specifying the server DLMS behaviour. A DLMS AE may contain one or both of these DLMS ASEs, depending on whether it is acting as a client, as a server or as both a client and a server. For the clarity of this standard, it is now assumed that communication between two DLMS AE-invocations involves an AE-invocation acting as a client and the other acting as server. The first one then contains a client DASE and the second a server DASE.

Les deux ASE de DLMS traitent les primitives des services venant des utilisateurs, de l'ACSE et de la couche liaison de données. Ils composent les A_PDU, traitent les paramètres des primitives des services et émettent des primitives de service vers l'utilisateur DLMS ou la couche liaison de données.

Parce que le DASE client et le DASE serveur sont une part très importante de la couche application dans la spécification DCP, elles sont décrites avec plus de détails dans les paragraphes suivants.

7.2.4 *Le contexte d'application*

Le contexte d'application contient toutes les informations nécessaires pour réaliser une relation coopérative entre deux AEI.

Cette information contient:

– une spécification de la syntaxe abstraite utilisée (définie pour être ASN.1);

– une spécification des règles de codage;

– la liste des ASE impliqués dans l'association d'applications;

– les règles relatives à la correspondance entre les APCI des ASE et, soit les services de la couche liaison de données, soit d'autres ASE;

– la désignation des fonctions d'application et des règles gouvernant leur usage;

– la méthode de cryptage et les règles gouvernant son usage;

– certaines valeurs par défaut, y compris une clé globale de cryptage commune à toutes les AE-invocation qui utilisent ce contexte d'application.

Le contexte d'application est prévu pour être référencé par son nom fourni par le service A_Unit_ Data de ACSE.

Le nom de contexte d'application qui réfère le contexte d'application décrit dans ce paragraphe est

default_DCP_Application_Context_Name::= { 3 2 18 19 67 }

On trouvera plus de détails sur le contexte d'application dans l'annexe A.

7.2.5 *Le contexte DLMS*

7.2.5.1 *Définition*

Le contexte DLMS est un objet spécifique dédié au maintien des caractéristiques de la communication dans un environnement DLMS. Ce contexte DLMS est spécifique aux AE-invocation. Il est vide avant que le premier DLMS Initiate (initialisation DLMS) du service n'ait été accompli avec succès ou après un service Abort (abandon).

Dans DCP, le contexte DLMS est lié directement à l'existence de l'environnement DLMS, et par là, à l'existence d'une association d'applications entre l'AE-invocation locale et une AE-invocation homologue, l'une agissant en tant que client DLMS et l'autre en tant que serveur DLMS. La définition du contexte DLMS dans une AE-invocation correspond exactement à la définition de l'objet VAA dans VDE.

La VAA représente alors dans le VDE l'existence

– d'un contexte DLMS;

– d'une association d'applications;

– de l'autre AE-invocation homologue.

Both DLMS application service elements process the service primitives coming from the DLMS user, from the ACSE, and from the data link layer. They form the A_PDUs, compute the service primitive parameters, and issue the service primitives to the DLMS user, or to the data link layer.

Because the client DASE and the server DASE are a very important part of the application layer of the DCP specification, they are described in more detail in the following subclauses.

7.2.4 *Application context*

The application context contains all the necessary information to make a co-operative relationship between two AEIs.

This information includes:

- specification of the abstract syntax used (defined to be ASN.1);
- specification of the encoding rules;
- list of the ASEs involved in the application-association;
- rules concerning the mapping of the APCI from ASEs onto the services of either the data link layer or of other ASEs;
- designation of application functions and rules governing their use;
- the ciphering method and the rules governing its use;
- some default values including a global ciphering key common to all the AE-invocations that use this application context.

The application context is intended to be referenced by its name, provided by the A_Unit_Data service of the ACSE.

The application context name that refers the application context described in this subclause is:

default_DCP_Application_Context_Name::= { 3 2 18 19 67 }

More details about the default DCP application context may be found in annex A.

7.2.5 *DLMS context*

7.2.5.1 *Definition*

The DLMS context is a specific object dedicated to the maintenance of the communication characteristics of the DLMS environment. This DLMS context is specific to the AE-invocation. It is empty before the first DLMS Initiate service is successfully performed or after an Abort service.

In DCP, the DLMS context is directly bound to the existence of a DLMS environment, and then to the existence of an application-association between the local AE-invocation and a peer AE-invocation, one acting as a DLMS client and the other as a DLMS server. The definition of a DLMS context within an AE-invocation corresponds exactly to the definition of a VAA object within the VDE.

The VAA then represents within the VDE the existence of:

- a DLMS context;
- an application-association;
- the other peer AE-invocation.

7.2.5.2 *Contenu*

Une fois qu'il est dans l'environnement DLMS, le contexte DLMS contient:

– toutes les caractéristiques de localisation de l'AE-invocation y compris les adresses et les titres du processus d'application et de l'entité application;

– toutes les caractéristiques de localisation de l'AE-invocation homologue y compris les adresses et les titres de l'AP et de l'AE;

– la qualité du service demandée par l'utilisateur DLMS (elle exprime le temps maximal accordé pour réaliser le service);

– le nom du contexte d'application qui supporte la communication;

– la clé secrète utilisée par les processus de cryptage et décryptage associés à la fonction A_Ciphering/Deciphering;

– le drapeau réponse autorisée associé à cette occurence de l'AE-invocation. S'il prend la valeur "faux", un service confirmé ne pourra pas être initialisé. Le A_PDU correspondant sera écarté.

La plupart des attributs du contexte DLMS sont remplis par les paramètres de la primitive invocation du service A_Unit_Data. Le reste des paramètres, y compris la clé de cryptage dédiée, ainsi que le drapeau réponse autorisée, sont remplis avec l'information contenue dans la demande d'initialisation du A_PDU.

Un contexte DLMS défini peut être modifié en utilisant à nouveau le service Initiate (initialisation). Ces modifications sont autorisées dans la spécification DCP à tout instant de la vie d'une association d'application particulière.

7.2.6 *Fonctions application*

Deux fonctions applications sont en général utilisées par l'ASE de DLMS: la fonction de répertoire d'applications et la fonction de cryptage et décryptage de l'application.

La fonction application directory (répertoire d'applications) est définie dans l'ISO 7498-3. Elle traite les titres des processus d'application, les modificateurs des entités application et les identificateurs des AE-invocation pour renvoyer les adresses correspondantes en termes de physical attachment (rattachement physique) et sélecteur link service access point (point d'accès au service de liaison). Elle réalise aussi l'inverse. Toutes les tables de dénomination sont des managed objects (objets gérés) et sont donc contenus dans le MIB.

La fonction application cryptage/décryptage est fournie pour assurer la sécurité et la confidentialité des données transmises. L'algorithme dépend de l'application et il convient qu'il soit spécifié dans une norme d'accompagnement. Le seul paramètre est alors celui de la clé utilisée. Deux clés sont définies: la clé globale: Global Ciphering Key et la clé dédiée: Dedicated Ciphering Key.

La clé globale de cryptage peut être dans le contexte d'application; elle est alors commune à toutes les AE-invocation qui utilisent ce contexte d'application. L'objectif d'une clé globale est de permettre une diffusion cryptée. La clé dédiée est contenue dans le contexte DLMS et elle est spécifique à cette occurence de l'association d'applications. Elle fournit des moyens pour l'authentification et la confidentialité pendant un échange d'informations intelligibles. La clé dédiée est transmise dans le champ A_PCI de la demande d'initialisation de A_PDU.

Un moyen supplémentaire d'augmenter la protection contre les rejeux non autorisés de messages préalablement émis réside dans l'inclusion de l'algorithme de cryptage "champ de vérification à copie unique" (one time copy check field) qui identifie de façon unique le message chiffré.

7.2.5.2 *Contents*

Once in the DLMS environment, the DLMS context contains:

– all the location characteristics of the AE-invocation including the addresses and titles of the application process and of the application entity;

– all the location characteristics of the peer AE-invocation, including the addresses and titles of the corresponding AP and AE;

– the quality of service that is requested from the DLMS user (it expresses the maximum delay allowed to achieve the service process);

– the application context name that supports the communication;

– the secret keys used to perform the ciphering and the deciphering processes associated with the A_Ciphering/Deciphering function;

– the response allowed flag associated with the instance of AE-invocation. If set to false, a confirmed service cannot be initiated. The corresponding A_PDU will be discarded.

Most of the DLMS context attributes are filled with the parameters of the A_Unit_Data service invocation primitive. The rest of the parameters, including the dedicated ciphering key, and the response allowed flag, are filled with the information contained in the Initiate request A_PDU.

A defined DLMS context may be modified by using the Initiate service again. These modifications are authorized in the DCP specification at any time during the life of a particular application-association.

7.2.6 *Application functions*

Two application functions are used mainly by the DLMS application service element, the application directory function and the application ciphering/deciphering function.

The application directory function is defined in ISO 7498-3. It processes application process titles, application entity qualifier and AE-invocation identifier to return the corresponding addresses in terms of physical attachment and link service access point selector. It also performs the opposite. All the naming tables are managed objects and contained then within the MIB.

The application ciphering/deciphering function is provided to ensure security and confidentiality of the transmitted data. The algorithm is application dependant and should be specified in a companion standard. The only parameter then is the keys to be used. Two keys are defined: the global ciphering key and the dedicated ciphering key.

The global ciphering key may be found in the application context, and is then common to all the AE-invocations that make use of this application context. The intent of the global ciphering key is to allow ciphered broadcasting. The dedicated ciphering key is contained in the DLMS context and is specific to that instance of application-association. It provides a means for authentication and confidentiality during the exchange of meaningful information. The dedicated ciphering key is transmitted in the A_PCI field of the Initiate request A_PDU.

An additional means for increasing the protection against unauthorized replay of previously sent messages resides in the inclusion, by the encryption algorithm, of a one-time copy-check field that will identify uniquely the instance of the message being encrypted.

7.3 *L'ASE client-DLMS*

Les primitives de services utilisées par le DASE du client sont spécifiées ci-dessous. Les primitives des services DLMS ainsi que l'usage qu'en fait ACSE ne sont pas symétriques entre le DASE du client DLMS et le DASE du serveur DLMS.

Des couches inférieures, le DASE client reçoit les primitives:

- indication du service DL_Data;
- confirm (confirmation) du service DL_Data.

Le DASE client envoie aux couches inférieures les primitives:

- request (demande) du service DL_Data;
- request (demande) du service A_Unit_Data.

De l'utilisateur de l'application, le DASE client reçoit les primitives:

- request (demande) pour les services DLMS Initiate (initialisation) et Abort (abandon);
- request (demande) pour les services confirmés de DLMS;
- request (demande) pour le service UnconfirmedWrite (écriture sans confirmation).

Le DASE client envoie à l'utilisateur de l'application les primitives:

- confirm (confirmation) du service DLMS Initiate (initialisation);
- confirm (confirmation) des services confirmés de DLMS;
- indication du service InformationReport de DLMS.

7.4 *L'ASE serveur-DLMS*

Des sous-couches, le DASE serveur reçoit les primitives:

- indication du service DL_Data;
- confirm (confirmation) du service DL_Data;
- confirm (confirmation) du service DL_Update_Reply;
- indication du service A_Unit_Data.

Le DASE serveur envoie aux sous-couches les primitives:

- request (demande) du service DL_Data;
- request (demande) du service DL_Update_Reply.

De l'utilisateur DLMS le DASE serveur reçoit les primitives:

- response (réponse) des services confirmés de DLMS;
- response (réponse) du service DLMS Initiate (initialisation);
- request (demande) du service InformationReport.

Le DASE serveur envoie à l'utilisateur DLMS les primitives:

- indication des services confirmés de DLMS;
- indication des services DLMS Initiate (initialisation) et Abort (abandon);
- indication du service UnconfirmedWrite.

7.3 *Client-DLMS ASE*

The service primitives used by the client DASE are specified below. The DLMS service primitives and the use made of the ACSE is unbalanced between the client DLMS ASE and the server DLMS ASE.

From the lower layers, the client DASE receives:

- the DL_Data service indication;
- the DL_Data service confirm.

To the lower layers, the client DASE sends:

- the DL_Data service request;
- the A_Unit_Data service request.

From the application user, the client DASE receives:

- the DLMS Initiate and Abort service request;
- the DLMS confirmed services request;
- the DLMS UnconfirmedWrite service request.

To the application user, the client DASE sends:

- the DLMS Initiate service confirm;
- the DLMS confirmed services confirm;
- the DLMS InformationReport service indication.

7.4 *Server-DLMS ASE*

From the lower layers, the server DASE receives:

- the DL_Data service indication;
- the DL_Data service confirm;
- the DL_Update_Reply service confirm;
- the A_Unit_Data service indication.

To the lower layers, the server DASE sends:

- the DL_Data service request;
- the DL_Update_Reply service request.

From the DLMS user, the server DASE receives:

- the DLMS confirmed services response;
- the DLMS Initiate service response;
- the DLMS InformationReport service request.

To the DLMS user, the server DASE sends:

- the DLMS confirmed services indication;
- the DLMS Initiate and Abort service indication;
- the DLMS UnconfirmedWrite service indication.

8 Eléments de procédure du protocole

Le présent article décrit les éléments de procédure du protocole relatif à l'émission et à la réception de PDU de DLMS ainsi que leurs relations avec les événements des primitives de services à la frontière entre l'utilisateur DLMS et le serveur DLMS.

8.1 *Conventions de description*

Dans le présent article, on utilise un mécanisme descriptif de diagrammes d'états. Le texte ci-dessous résume ce mécanisme. Il faut remarquer que tous les diagrammes d'états sont exposés selon le point de vue du prestataire de service de DLMS.

Chaque état est représenté par un cercle. Le nom de l'état est indiqué dans ce cercle. Chaque flèche représente une transition hors d'un état. La tête de la flèche pointe sur l'état objectif, qui est l'état admis comme résultat de la transition. L'événement qui a causé la transition du DLPM d'un état à l'autre est indiqué dans un cadre. L'action associée à la transition est décrite à la pointe de la flèche de transition.

Les primitives de service auxquelles on a attaché un "+" contiennent un paramètre Result(+). Les primitives de service auxquelles on a attaché un "–" contiennent un paramètre Result(–).

8.2 *Correspondance avec ACSE et les services data link (liaison de données)*

Le présent paragraphe définit la manière dont l'ACSE et les services de la couche liaison de données sont utilisés par DLPM (machine de protocole de ligne de distribution). Tout usage de ACSE ou du service liaison de données autre que celui qui est décrit ici est une erreur de protocole.

Le protocole DLMS est placé dans la couche application de l'environnement OSI. En tant que ASE (élément de service d'application), DLMS ASE utilise les services et les primitives de service de ACSE et de la couche liaison de données. L'utilisateur DLMS peut être fait d'éléments d'un processus d'application, ou d'autres ASE.

8.2.1 *Correspondance des PDU*

Un PDU de DLMS doit être traité comme des données utilisateur par ACSE et les primitives des services de data link (liaison de données). La correspondance des PDU et des services doit s'établir comme suit (tous les PDU sont envoyés dans des primitives request (demande) ou response (réponse), et sont reçus dans des primitives indication ou confirm (confirmation)):

DLMS PDU	*ACSE or Data Link Service Primitive*
ConfirmedServiceRequest	DL_Data request, indication
ConfirmedServiceResponse	DL_Data request, indication
ConfirmedServiceError	DL_Data request, indication
UnconfirmedServiceRequest	DL_Data request, indication
UnsolicitedServiceRequest	DL_UpdateReply request, DL_Reply confirm
InitiateRequest	A_UnitData request, indication
InitiateResponse	DL_Data request, indication
InitiateError	DL_Data request, indication
AbortRequest	DL_Data request, indication

8 Elements of protocol procedure

This clause describes the elements of protocol procedure related to the sending and receiving of DLMS PDUs and their relation to service primitive events at the DLMS user and DLMS provider boundary.

8.1 *Descriptive convention*

In this clause, a state diagram descriptive mechanism is used. The following text summarizes this mechanism. Note that all state diagrams are shown from the viewpoint of the DLMS provider.

Each state is represented by a circle. The name of the state is shown inside the circle. Each arrow represents a transition in or out of a state. The head of the arrow points to the output state, which is the state entered as a result of the transition. The event that generated the transition of the DLPM from one state to the other is shown in a box. The action associated with the transition is described at the head of the crossing arrow.

The service primitives to which a "+" is appended indicate a service primitive containing a Result(+) parameter. Service primitives to which a "–" is appended indicate a service primitive containing a Result(–) parameter.

8.2 *Mapping to ACSE and data link services*

This subclause defines the way in which the association control service element (ACSE) and the data link layer services are used by the distribution line protocol machine (DLPM). Any use of the ACSE service or data link service other than as described in this subclause is a protocol error.

The DLMS protocol is positioned within the Open Systems Interconnection environment at the application layer. As an application service element (ASE), DLMS ASE makes use of, and maps onto, the services and service primitives of ACSE and the data link layer. The DLMS user may be elements within the application process or another ASE.

8.2.1 *Mapping of PDUs*

All DLMS PDUs shall be carried as user data on an ACSE or data link service primitive. The mapping of PDUs to services shall be as follows (all PDUs are sent on a request or response service primitive, and are received on an indication or confirm service primitive):

DLMS PDU	*ACSE or Data Link Service Primitive*
ConfirmedServiceRequest	DL_Data request, indication
ConfirmedServiceResponse	DL_Data request, indication
ConfirmedServiceError	DL_Data request, indication
UnconfirmedServiceRequest	DL_Data request, indication
UnsolicitedServiceRequest	DL_UpdateReply request, DL_Reply confirm
InitiateRequest	A_UnitData request, indication
InitiateResponse	DL_Data request, indication
InitiateError	DL_Data request, indication
AbortRequest	DL_Data request, indication

8.2.2 *Construction des PDU de DLMS*

Lors de la réception d'une primitive request ou response d'un des services de DLMS, DLPM doit

1) construire le PDU demandé par le DLMS ASE protocol specification (spécification de protocole DLMS ASE) pour le service spécifié dans la primitive conformément aux exigences du protocole pour ce service;

2) envoyer le PDU construit en tant que données utilisateur à l'ACSE ou aux primitives du service data link spécifié ci-dessus.

8.2.3 *Construction des PDU de l'application*

Les 18 PDU utilisés pour exploiter le protocole DASE sont décrits, ainsi que les neuf PDU du pur protocole DLMS, dans l'annexe A de la CEI 1334-4-41.

8.2.3.1 *Sémantique de la partie cryptée*

Chaque PDU de DASE peut être crypté. En plus du mode en clair, utilisé pour les protocoles dits "pure DLMS", il est possible de crypter chaque PDU à l'aide, soit d'une clé de cryptage globale, soit d'une clé de cryptage dédiée. Pour ces deux derniers modes, la valeur des PDU cryptés doit être codée à l'aide du type OCTET STRING (chaîne d'octets) de ASN.1. Pour le cryptage global et le cryptage dédié, le PDU crypté doit être codé conformément aux spécifications de DLMS (CEI 1334-4-41).

8.2.3.2 *Syntaxe de la partie cryptée*

Afin de contrôler le processus de décryptage, un octet spécial est incorporé dans la première position de la chaîne cryptée. Avant le cryptage et donc après le décryptage, cette position doit être égale à zéro (00 Hex). Une valeur différente sera interprétée comme une erreur de cryptage/décryptage; l'APDU sera rejeté. Cet octet est ajouté pendant le cryptage et supprimé pendant le décryptage une fois qu'il aura été correctement décodé. Il n'y a pas de valeur sémantique attribuée à ce premier octet dont la valeur est zéro.

8.2.4 *Livraison de primitives de service à un utilisateur DLMS*

Après réception d'une primitive de service indication ou confirmation de ACSE ou de data link, DLPM doit déterminer si la primitive de service reçue contient en tant que données utilisateur un DASE PDU valable. Un DASE PDU valable est celui qui, une fois décrypté, satisfait les exigences de la syntaxe abstraite du DASE pour la définition des PDU de DASE, correspond à la bonne primitive de service de ACSE ou de data link (tel que défini ci-dessus), et arrive conformément aux règles de séquencement définies dans la présente norme.

Si la primitive de service contient un DASE PDU valable, DLPM doit émettre la primitive de service indication ou confirm appropriée avec des valeurs de primitives déduites conformément aux exigences spécifiées dans l'annexe A de la CEI 1334-4-41.

Si la primitive de service ne contient pas un DASE PDU valable, DLPM doit l'ignorer.

8.2.5 *Service sous-jacent fiable*

DLMS ne prévoit pas de traiter des messages en désordre, des erreurs de transmissions, des messages perdus ou des messages en double. DLMS suppose qu'il existe un service fiable de communication de données entre deux entités application. La seule protection contre de telles erreurs est le rejet des DASE PDU rencontrés par DLPM.

8.2.2 *Construction of DLMS PDUs*

Upon receipt of a request or response service primitive for any DLMS service, the DLPM must:

1) construct the PDU required by the DLMS ASE protocol specification for the service specified in the primitive, in accordance with the protocol requirements for that service;

2) send the constructed PDU as user data on the ACSE or data link service primitive specified above.

8.2.3 *Construction of application PDUs*

The 18 PDUs used to operate the DASE protocol are described, along with the nine PDUs of the pure DLMS protocol, in annex A of IEC 1334-4-41.

8.2.3.1 *Semantic of the ciphered part*

Each DASE PDU may be ciphered. In addition to the non-ciphering mode, used for the so-called "pure DLMS protocol", it is possible to cipher each PDU using either a global ciphering key or a dedicated ciphering key. For the latter two modes, the value of the ciphered PDU shall be encoded as an ASN.1 OCTET STRING type. For global ciphering and dedicated ciphering, the ciphered PDU shall be encoded in accordance to the DLMS specification (IEC 1334-4-41).

8.2.3.2 *Syntax of the ciphered part*

In order to control the deciphering process, a special byte is inserted at the first position of the ciphered string. Before ciphering and then after deciphering, this byte must be equal to zero (00 Hex). A different value is interpreted as a ciphering/deciphering error; the APDU will be rejected. This byte is added at ciphering time and deleted at deciphering time once properly decoded. No semantic is attached to this first byte with a zero value.

8.2.4 *Delivery of service primitives to an DLMS-user*

Upon receipt of an indication or confirm service primitive from ACSE or data link, the DLPM shall determine if the service primitive received contains as user data a valid DASE PDU. A valid DASE PDU is one that, once deciphered, meets the requirements of the DASE abstract syntax for the definition of DASE PDUs, is mapped to the correct ACSE or data link service primitive (as defined above), and arrives in conformance with all sequencing rules defined in this standard.

If the service primitive received contains a valid DASE PDU, the DLPM shall issue the appropriate indication or confirm service primitive, with values in the primitive derived in accordance with the requirements specified in annex A of IEC 1334-4-41.

If the service primitive received does not contain a valid DASE PDU, the DLPM shall discard it.

8.2.5 *Reliable underlying service*

DLMS makes no provisions for handling of misordered messages, transmission errors, lost messages, or duplicated messages. DLMS assumes that a reliable underlying service for communicating data between two application entities exists. The only protection against such errors is the discard of the encountered DASE PDUs by the DLPM.

8.2.6 *Contrôle de flux*

Il n'existe pas de contrôles de flux homologues dans DLMS. Le DLPM destinataire peut exercer une pression en retour à travers l'association d'applications, afin de limiter la capacité de son homologue à émettre des données. La décision de quand et comment cette pression s'exerce est un problème local.

8.3 *Entrée et sortie de l'environnement DLMS*

8.3.1 *Généralités sur les procédures*

Les services Initiate (initialisation) et Abort (abandon) fournissent les mécanismes pour entrer et sortir de l'environnement DLMS. Le modèle pour ces deux services est décrit dans l'article 5 de la CEI 1334-4-41. Deux AE-invocation entrent dans l'environnement DLMS quand elles ont chacune défini le même contexte DLMS et quand ceci est connu de l'autre AE-invocation.

Du côté client, tout commence par la réception d'une primitive request du service Initiate. Le DASE client constitue le A_PDU de la primitive request (demande) de Initiate, en attachant les paramètres soumis avec la demande de service, la clé de cryptage dédiée et l'indicateur response allowed (réponse autorisée). Il crypte alors le A_PDU de la demande d'Initiate à l'aide de la clé globale qui se trouve dans le contexte d'application. Un A_Unit_Data est émis avec les paramètres appropriés: nom du contexte d'application, A_PDU de la demande d'Initiate et qualité de service.

Le client DASE attend la confirmation locale. Si elle est négative, un Initiate confirm négatif est envoyé à l'utilisateur DLMS, précisant l'origine de l'erreur. Si elle est positive, une temporisation est lancée pour récupérer les trames perdues.

Sur réception d'A_Unit_Data indication, le serveur DASE vérifie les paramètres et déchiffre l'Initiate request A_PDU lui arrivant, avec la clé globale de cryptage contenue dans le contexte application. S'il rencontre une erreur, l'A_PDU est écartée. Sinon, le serveur DASE crée (ou modifie) le contexte DLMS avec les valeurs extraites des paramètres ou issues de l'A_PDU. Une Initiate indication est alors émise vers l'utilisateur DLMS.

S'il reçoit une primitive Initiate response positive, le serveur DASE constitue un A_PDU pour l'Initiate response avec les paramètres soumis et la qualité de service. Le A_PDU est alors crypté à l'aide de la clé de cryptage dédiée et un DL_Data request est émis vers la couche liaison de données. Si une confirmation locale négative est reçue en retour, une indication Abort (abandon) spécifiant un prestataire source initialisé vers l'utilisateur DLMS et le contexte DLMS est annulé.

Si une Initiate response négative est reçue, le serveur DASE annule le contexte DLMS après avoir émis une primitive DL_Data request contenant un A_PDU Initiate Error (erreur d'initialisation).

Le client DASE décrypte les A_PDU des réponses Initiate qui arrivent et, quand elles sont correctement constituées, émet une confirmation Initiate positive. Si la temporisation expire avant de recevoir une A_PDU de réponse Initiate, une confirmation Initiate négative est émise.

8.3.2 *Gestion des tables de changements d'état de contexte*

Le présent paragraphe spécifie les séquences d'événements admissibles.

8.2.6 *Flow control*

There is no peer flow control in DLMS. The receiving DLPM may apply back pressure across the application-association to limit the ability of the peer to send data. The decision on when or how back pressure is applied is a local matter.

8.3 *Entering and leaving the DLMS environment*

8.3.1 *Overview of the procedures*

The Initiate and the Abort services provide the mechanisms for entering and leaving the DLMS environment. The model for these services is described in clause 5 of IEC 1334-4-41. Two AE-invocations enter in the DLMS environment when they have both defined the same DLMS context, and when this is known from the other AE-invocation.

At the client end, all begins with the reception of an Initiate request primitive. The client DASE forms the Initiate request A_PDU by appending the parameters submitted with the service request, the dedicated ciphering key, and the indicator response allowed. It then ciphers the Initiate request A_PDU with the global ciphering key found in the application context. An A_Unit_Data is issued with the proper parameters: application context name, Initiate request A_PDU, and quality of service.

The client DASE waits on the local confirm. If it is negative, a negative Initiate confirm is issued to the DLMS user, specifying the error source. If it is positive, a timer is set to recover from lost frames.

On reception of an A_Unit_Data indication, the server DASE checks the submitted parameters and deciphers the incoming Initiate request A_PDU with the global ciphering key contained in the application context. If an error is encountered, the A_PDU is discarded. Otherwise, the server DASE creates (or modifies) the DLMS context with the proper values extracted from the parameters, or from the A_PDU. An Initiate indication is then issued to the DLMS user.

If a positive Initiate response primitive is received, the server DASE forms an Initiate response A_PDU with the submitted parameters and the quality of service. The A_PDU is then ciphered with the dedicated ciphering key and a DL_Data request is issued to the data link layer. If a negative local confirm is received in return, an Abort indication specifying a provider initiated source is issued to the DLMS user and the DLMS context is deleted.

If a negative Initiate response is received, the server DASE deletes the DLMS context after it has issued a DL_Data request containing an Initiate Error A_PDU.

The client DASE deciphers an incoming Initiate response A_PDU and, when properly formed, issues a positive Initiate confirm. If the timer expires before receiving any Initiate response A_PDU, a negative Initiate confirm is issued.

8.3.2 *Context management state tables*

This subclause specifies the allowable sequences of events.

Tableau 2 – Gestion de la table de changements d'état du contexte serveur

Initial state	Events	Actions	Final state
NoCon	A_Unit_Data.ind and Check_Parameter () = Ok and Deciphering () = Ok	Extract_Field (UserInformation) Set_Dlms_Context (>A_DirectoryF () !Calling Title, !Called Title, A_Context_Name, Response-Allowed, Dedicated_Key) Initiate.ind	Con.E
NoCon	A_Unit_Data.ind and (Check_Parameter () <> Ok or Deciphering () <> Ok)	None	NoCon
Con.E	Initiate.rsp (+)	Build_APDU (Type = InitiateRsp_APDU, Cipher (Dedicated_Key, A_SDU, Qos)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, InitiateRsp_APDU)	Con.W
Con.E	Initiate.rsp (–)	Build_APDU (Type = InitiateErr_APDU, Cipher (Global_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, InitiateErr_APDU)	Con.R
Con.W	DL_Data.cnf (+)	None	IDLE
Con.W	DL_Data.cnf (–)	Abort.ind (Service_Provider) Delete_Dlms_Context ()	NoCon
Con.R	DL_Data.cnf	Delete_Dlms_Context ()	NoCon
IDLE	DL_Data.ind (AbortReq_PDU) and Deciphering () = Ok	Extract_Field (AbortReq_APDU) Abort.ind Delete_Dlms_Context ()	NoCon
IDLE	DL_Data.ind (AbortReq_PDU) and Deciphering () <> Ok	None	IDLE
IDLE	A_Unit_Data.ind and Check_Parameter () = Ok and Deciphering () = Ok	Extract_Field (UserInformation) Set_Dlms_Context (A_Context_Name, Response-Allowed, Dedicated_Key) Initiate.ind	ID.E
IDLE	A_Unit_Data.ind and (Check_Parameter () <> Ok or Deciphering () <> Ok)	None	IDLE

Table 2 – Server context management state table

Initial state	Events	Actions	Final state
NoCon	A_Unit_Data.ind and Check_Parameter () = Ok and Deciphering () = Ok	Extract_Field (UserInformation) Set_Dlms_Context (>A_DirectoryF () !Calling Title, !Called Title, A_Context_Name, Response-Allowed, Dedicated_Key) Initiate.ind	Con.E
NoCon	A_Unit_Data.ind and (Check_Parameter () <> Ok or Deciphering () <> Ok)	None	NoCon
Con.E	Initiate.rsp (+)	Build_APDU (Type = InitiateRsp_APDU, Cipher (Dedicated_Key, A_SDU, Qos)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, InitiateRsp_APDU)	Con.W
Con.E	Initiate.rsp (–)	Build_APDU (Type = InitiateErr_APDU, Cipher (Global_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, InitiateErr_APDU)	Con.R
Con.W	DL_Data.cnf (+)	None	IDLE
Con.W	DL_Data.cnf (–)	Abort.ind (Service_Provider) Delete_Dlms_Context ()	NoCon
Con.R	DL_Data.cnf	Delete_Dlms_Context ()	NoCon
IDLE	DL_Data.ind (AbortReq_PDU) and Deciphering () = Ok	Extract_Field (AbortReq_APDU) Abort.ind Delete_Dlms_Context ()	NoCon
IDLE	DL_Data.ind (AbortReq_PDU) and Deciphering () <> Ok	None	IDLE
IDLE	A_Unit_Data.ind and Check_Parameter () = Ok and Deciphering () = Ok	Extract_Field (UserInformation) Set_Dlms_Context (A_Context_Name, Response-Allowed, Dedicated_Key) Initiate.ind	ID.E
IDLE	A_Unit_Data.ind and (Check_Parameter () <> Ok or Deciphering () <> Ok)	None	IDLE

Tableau 2 *(fin)*

Initial state	Events	Actions	Final state
ID.E	Initiate.rsp (+) (note)	Build_APDU (Type = InitiateRsp_APDU, Cipher (Dedicated_Key, A_SDU, Qos)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, InitiateRsp_APDU)	ID.W
ID.W	DL_Data.cnf	None	IDLE
NOTE – The response must be (+) as specified in DLMS.			

Tableau 3 – Gestion de la table de changements d'état du contexte client

Initial state	Events	Actions	Final state
NoCon	Initiate.req	Build_APDU (Type = Initiate.req, Cipher (Global_Key, A_SDU, >From_Dlms_Context() !Response_Allowed, !Dedicated_Key) A_UnitData.req (A_Context_Name, InitiateReq_APDU)	NC.W
NC.W	DL_Data.cnf (+)	Init_Timer ()	NC.T
NC.W	DL_Data.cnf (–)	Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	DL_Data.ind (InitiateRsp_APDU) and Deciphering () = Ok	Stop_Timer () Extract_Field (InitiateRsp_APDU) A_DirectoryF (!SourceTitle, Quality_of_Service) Initiate.cnf (+)	IDLE
NC.T	DL_Data.ind (InitiateRsp_APDU) and Deciphering () <> Ok	Stop_Timer () Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	DL_Data.ind (InitiateErr_APDU) and Deciphering () = Ok	Stop_Timer () Extract_Field (InitiateErr_APDU) Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	DL_Data.ind (InitiateErr_APDU) and Deciphering () <> Ok	Stop_Timer () Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	Time_Out ()	Initiate.cnf (–) Delete_Dlms_Context ()	NoCon

Table 2 *(concluded)*

Initial state	Events	Actions	Final state
ID.E	Initiate.rsp (+) (note)	Build_APDU (Type = InitiateRsp_APDU, Cipher (Dedicated_Key, A_SDU, Qos)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, InitiateRsp_APDU)	ID.W
ID.W	DL_Data.cnf	None	IDLE
NOTE – The response must be (+) as specified in DLMS.			

Table 3 – Client context management state table

Initial state	Events	Actions	Final state
NoCon	Initiate.req	Build_APDU (Type = Initiate.req, Cipher (Global_Key, ASDU, >From_Dlms_Context() !Response_Allowed, !Dedicated_Key) A_UnitData.req (A_Context_Name, InitiateReq_APDU)	NC.W
NC.W	DL_Data.cnf (+)	Init_Timer ()	NC.T
NC.W	DL_Data.cnf (–)	Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	DL_Data.ind (InitiateRsp_APDU) and Deciphering () = Ok	Stop_Timer () Extract_Field (InitiateRsp_APDU) A_DirectoryF (!SourceTitle, Quality_of_Service) Initiate.cnf (+)	IDLE
NC.T	DL_Data.ind (InitiateRsp_APDU) and Deciphering () <> Ok	Stop_Timer () Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	DL_Data.ind (InitiateErr_APDU) and Deciphering () = Ok	Stop_Timer () Extract_Field (InitiateErr_APDU) Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	DL_Data.ind (InitiateErr_APDU) and Deciphering () <> Ok	Stop_Timer () Initiate.cnf (–) Delete_Dlms_Context ()	NoCon
NC.T	Time_Out ()	Initiate.cnf (–) Delete_Dlms_Context ()	NoCon

Tableau 3 *(fin)*

Initial state	Events	Actions	Final state
IDLE	Abort.req	Build_APDU (Type = Abort.req, Cipher (Dedicated_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, AbortReq_APDU) Delete_Dlms_Context ()	NoCon
IDLE	Initiate.req	Build_APDU (Type = Initiate.req, Cipher (Global_Key, A_SDU, >From_Dlms_Context() !Response_Allowed, !Dedicated_Key)) A_UnitData (A_Context_Name, InitiateReq_APDU)	NC.W

Les contextes DLMS doivent être définis avant l'émission d'une primitive request d'Initiate.

8.3.3 *Notations utilisées dans les tables d'états*

8.3.3.1 *Notations pour les services DLMS*

Les services DLMS utilisés ici sont les services Initiate (initialisation) et Abort (abandon). Ils sont décrits dans la CEI 1334-4-41. Les paramètres soumis à un service DLMS ne sont pas explicites mais doivent correspondre à une syntaxe abstraite du service spécifié dans l'annexe A de la CEI 1334-4-41.

8.3.3.2 *Notations pour les services ACSE*

Le service A_Unit_Data est décrit à l'article 6. Les paramètres soumis avec la primitive request sont: nom du contexte d'application, données utilisateur et qualité de service. Ces paramètres sont extraits par l'ACSE destinataire et soumis avec la primitive indication, bien que ce ne soit pas explicite dans la table d'états.

8.3.3.3 *Notations pour le service data link (liaison de données)*

Le service DL_Data est le seul service data link (liaison de données) utilisé par les AE de DLMS pour entrer ou sortir de l'environnement DLMS. Les paramètres soumis avec les primitives de service DL_Data sont: Destination_SAP et Destination_address (compactés en Called Titles (titre appelé)), Source_SAP et Source_address (compactés en Calling Titles (titre appelant)) et le L_SDU.

8.3.4 *Description des états*

Pour le serveur, les états suivants sont définis dans la table d'états:

– NoCon (no context, pas de contexte). Il n'y a pas de contexte DLMS défini dans l'AE-invocation;

Table 3 *(concluded)*

Initial state	Events	Actions	Final state
IDLE	Abort.req	Build_APDU (Type = Abort.req, Cipher (Dedicated_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, AbortReq_APDU) Delete_Dlms_Context ()	NoCon
IDLE	Initiate.req	Build_APDU (Type = Initiate.req, Cipher (Global_Key, A_SDU, >From_Dlms_Context() !Response_Allowed, !Dedicated_Key)) A_UnitData (A_Context_Name, InitiateReq_APDU)	NC.W

The DLMS context have to be defined before issuing an Initiate request primitive.

8.3.3 *Notation used in state tables*

8.3.3.1 *Notation for DLMS services*

The DLMS services used here are the Initiate and the Abort services. They are as described in IEC 1334-4-41. The parameters submitted with a DLMS service are not explicit, but must correspond to the abstract syntax of the service specified in annex A of IEC 1334-4-41.

8.3.3.2 *Notation for the ACSE service*

The A_Unit_Data service is described in clause 6. The parameters submitted with the request primitive are: application context name, user data and quality of service. These parameters are extracted by the receiver ACSE and submitted with the service indication primitive, although this is not explicit in the state table.

8.3.3.3 *Notation for data link services*

The DL_Data service is the only data link service used by the DLMS AE for entering or leaving the DLMS environment. The parameters submitted with a DL_Data service primitive are: Destination_SAP and Destination_address (compacted in Called Titles), Source_SAP and Source_address (compacted in Calling Titles) and the L_SDU.

8.3.4 *States description*

For the server, the following states are defined in the state table:

– NoCon (No Context). No DLMS context is defined within the AE-invocation;

– Con.E (expect context, attente de contexte). Un A_PDU de demande d'initiate a été reçu pendant l'état NoCon et l'AE-invocation attend une réponse de l'utilisateur DLMS;

– Con.W (waiting for confirm, attente de confirmation). Une réponse positive a été envoyée et l'AE-invocation attend la confirmation locale;

– Con.R (waiting for confirm and realease, attente de confirmation et de libération). La réponse négative a été envoyée et l'AE-invocation attend la confirmation locale;

– ID.E (Idle expected, Inoccupé attendu). Un A_PDU d'une demande Initiate a été reçu dans l'état IDLE (inoccupé) et l'AE-invocation attend une réponse de l'utilisateur DLMS;

– ID.W (waiting for idle, attente du passage en inoccupé). La réponse a été envoyée et l'AE-invocation attend une confirmation locale;

– IDLE (inoccupé). L'AE-invocation est prête à fonctionner et la configuration du contexte DLMS a été réalisée avec succès.

Pour le client, les états suivants sont définis dans la table d'états:

– NoCon (no context, pas de contexte). Il n'y a pas de contexte DLMS défini dans l'AE-invocation;

– NC.W (no context and waiting for confirm, pas de contexte et attente de confirmation). La demande a été envoyée et l'AE-invocation attend une confirmation locale;

– NC.T (no context and waiting under timer, pas de contexte et attente avec minuterie). L'AE-invocation attend une réponse de l'utilisateur homologue de DLMS ou la fin du délai;

– IDLE (inoccupé). L'AE-invocation est prête à fonctionner et la configuration du contexte DLMS a été réalisée avec succès.

8.3.5 *Description des fonctions événements*

8.3.5.1 *Deciphering() (décryptage)*

Le A_PDU contient un champ qui indique la clé de cryptage utilisée pour le A_PDU. Cette clé peut être la clé globale, la clé dédiée ou pas de clé du tout (pas de cryptage). La fonction de décryptage vérifie que la partie cryptée de l'A_PDU peut être effectivement décryptée avec le clé contenue dans le contexte DLMS. Elle renvoie OK si le résultat du processus de décryptage est conforme avec la syntaxe abstraite spécifiée pour cette partie de l'A_PDU quand elle a été décryptée.

8.3.5.2 *Check-Parameters() (vérification des paramètres)*

La fonction Check_Parameters vérifie les paramètres fournis par la primitive de service indication de A_UnitData. Les paramètres obligatoires doivent être présents et tous les paramètres doivent être conformes à la syntaxe abstraite spécifiée par la primitive indication de A_UnitData.

8.3.5.3 *Time_Out() (hors délais)*

L'événement Time_Out est généré quand le délai généré par la fonction Init_Timer expire. Comment signaler cet événement au DASE client est un problème local.

8.3.6 *Description des actions*

8.3.6.1 *Set_ Dlms_Context() (créer un contexte DLMS)*

La fonction Set_Dlms_Context crée ou modifie le contexte DLMS. Les paramètres de la fonction sont les calling and called titles (titres appelants et appelés), le nom du contexte d'application, la Response_Allowed (réponse autorisée) et la clé de cryptage dédiée.

– Con.E (Expect Context). An Initiate request A_PDU has been received while in the NoCon state and the AE-invocation is waiting for the answer of the DLMS user;

– Con.W (Waiting for Confirm). The positive answer has been sent and the AE-invocation waits for the local confirm;

– Con.R (Waiting for Confirm and Release). The negative answer has been sent and the AE-invocation waits for the local confirm;

– ID.E (Idle Expected). An Initiate request A_PDU has been received while in the IDLE state and the AE-invocation is waiting for the answer of the DLMS user;

– ID.W (Waiting for Idle). The answer has been sent and the AE-invocation is waiting for the local confirm;

– IDLE. The AE-invocation is ready to operate and the configuration of the DLMS context is successful.

For the client, the following states are defined in the state table:

– NoCon (No Context). No DLMS Context is defined within the AE-invocation;

– NC.W (No Context and Waiting for confirm). The request has been sent and the AE-invocation waits for the local confirm;

– NC.T (No Context and waiting under Timer). The AE-invocation is waiting for an answer of the peer DLMS user or for the time out;

– IDLE. The AE-invocation is ready to operate and the configuration of the DLMS context is successful.

8.3.5 *Events functions description*

8.3.5.1 *Deciphering ()*

The A_PDU contains a field that indicates the ciphering key used for the A_PDU. This key may be the global key, the dedicated key or no key at all (no ciphering). The deciphering function verifies that a ciphered A_PDU may be consistently deciphered with the keys contained in the DLMS context. It returns OK if the result of the deciphering process is consistent with the abstract syntax specified for this deciphered part of the A_PDU.

8.3.5.2 *Check_Parameters()*

The Check_Parameters function checks the parameters provided by the A_UnitData indication service primitive. The mandatory parameters must be present and all parameters must be consistent with the abstract syntax specified for the A_UnitData indication.

8.3.5.3 *Time_Out ()*

The Time_Out event is generated when the timer initiated by the Init_Timer function expires. How this event is signalled to the client DASE is a local issue.

8.3.6 *Actions description*

8.3.6.1 *Set_Dlms_Context ()*

The Set_Dlms_Context function creates or modifies the DLMS context. The parameters of the function are the Calling and Called Titles, the application context name, the Response_Allowed and the dedicated ciphering key.

8.3.6.2 *Delete_Dlms_Context() (annuler un contexte DLMS)*

La fonction Delete_Dlms_Context efface toutes les informations contenues dans le contexte DLMS.

8.3.6.3 *Build_APDU (Type, Cipher(A_SDU)) (construire une A_PDU)*

La fonction Build_APDU construit une A_PDU du type spécifié dans le paramètre. Ce type doit être un des huit types définis dans l'annexe A de la CEI 1334-4-41. L'A_PDU est composé avec le résultat du processus de cryptage indiqué par la fonction de cryptage.

Le premier paramètre de la fonction cryptage est la clé de cryptage à utiliser. L'autre paramètre de la fonction de cryptage contient les données à crypter. L'argument A_SDU décrit les paramètres émis par l'utilisateur DLMS avec la primitive de service.

Le résultat de la fonction Build_APDU est un A_PDU du type requis.

8.3.6.4 *Extract_Field() (extracteur de champs)*

La fonction extract_field extrait les champs d'un A_PDU donné conformément à une syntaxe abstraite liée au type de l'A_PDU. Ce processus peut éventuellement comprendre le décryptage d'une partie de l'A_PDU. Les clés de décryptage pourront être trouvées dans le contexte DLMS. Les différents champs extraits sont alors disponibles sous leurs noms usuels pour les besoins de l'AE-invocation.

8.3.6.5 *Init_Timer() et Stop_Timer() (initialiser et arrêter une temporisation)*

Les fonctions Init_Timer et Stop_Timer (initialiser et arrêter une temporisation) démarrent et arrêtent une temporisation. Cette temporisation sert à la détection et à la récupération des trames perdues pendant la procédure de communication. C'est la seule temporisation définie dans la spécification DCP. Le paramètre valeur de la fonction Init_Timer doit spécifier un intervalle de temps supérieur au délai maximum nécessaire à l'exécution réussie d'un service. La valeur peut être exprimée à l'aide de toute unité de temps appropriée.

NOTE – Le paramètre valeur de la fonction Init_Timer peut être traité dans DCP à l'aide de la Management Information Base, en fonction des paramètres disponibles dans les autres couches. Ces paramètres sont: Number of Subframes (nombre de sous-trames), Crédit Initial, diffusion ou non, et la qualité de service. Le nombre de sous-trames n'est pas un objet géré (managed object) mais il peut être calculé à partir de la longueur de l'A_PDU, du L_PCI, du M_PCI et de la longueur de la sous-trame. La valeur finale peut être exprimée en secondes ou en Time_Slot (tops horaires).

8.4 *Exploitation dans l'environnement DLMS*

Une fois que l'on est dans DLMS, et pour une association d'applications données, il ne peut y avoir qu'un seul service actif à un instant donné. Ce service doit être totalement satisfait avant l'exploitation de tout autre service. La présente norme décrit les diagrammes d'état pour chaque cas indépendant de demande de service.

8.4.1 *Services DLMS confirmés*

8.4.1.1 *Domaine d'application*

Le présent paragraphe décrit les changements d'état pour tous les services confirmés qui peuvent être invoqués dans l'environnement DLMS. Cet ensemble de services est constitué de tous les services que l'on peut demander à l'aide de ConfirmedService RequestDASEPDU.

8.3.6.2 *Delete_Dlms_Context ()*

The Delete_Dlms_Context function deletes all information contained in the DLMS context.

8.3.6.3 *Build_APDU (Type, Cipher(A_SDU))*

The Build_APDU function constructs an A_PDU of the type specified in parameter. The type must be one of the eight types defined in IEC 1334-4-41, annex A. The A_PDU is formed with the result of the ciphering process designed by the cipher function.

The first parameter of the cipher function is the ciphering key to be used. The other parameters of the cipher function contain the data to be encrypted. The A_SDU argument describes the parameters issued by the DLMS user with the service primitive.

The result of the Build_APDU function is an A_PDU of the required type.

8.3.6.4 *Extract_Field ()*

The Extract_Field function extracts the various fields of a specified A_PDU according to the abstract syntax attached to the A_PDUs type. This process eventually includes the deciphering of part of the A_PDU. The ciphering keys may be found in the DLMS context. The various extracted fields are then available for the AE-invocation under their usual names.

8.3.6.5 *Init_Timer () and Stop_Timer ()*

The Init_Timer and Stop_Timer functions start and stop a timer. This timer is provided for the detection and recovery of lost frames in the communication procedure. It is the only timer defined in the DCP specification. The parameter value of the Init_Timer function must specify an amount of time greater than the maximum time of a successful service completion. The value may be expressed in any appropriate time unit.

> NOTE – The parameter value of the Init_Timer function may be computed in DCP according to parameters of other layers available through the Management Information Base. The parameters are the Number of Subframes, the Initial Credit, whether it is broadcasting or not, and the quality of service. The Number of Subframes is not a Managed Object, but may be computed from the A_PDU length, the L_PCI length, the M_PCI length and the length of a subframe. The final value may be expressed either in seconds or in Time_Slot.

8.4 *Operating in the DLMS environment*

Once in the DLMS environment and for a particular application-association, there may be only one service outstanding at any instant of time. This service must be fully completed before performing any other service. This standard describes the state diagram for such service request instances independently.

8.4.1 *Confirmed DLMS services*

8.4.1.1 *Scope of application*

This subclause describes the state transitions for all confirmed services that may be invoked within the DLMS environment. This set of services consists of all services that are requested through the use of the ConfirmedServiceRequestDASEPDU.

La liste de ces services est la suivante:

- GetStatus service (obtenir l'état);
- GetNameList service (obtenir la liste de noms);
- InitiateLoad service (initialiser le chargement);
- LoadSegment service (charger un segment);
- TerminateLoad service (terminer un chargement);
- InitiateUpLoad service (initialiser le rapatriement);
- UpLoadSegment service (rapatrier un segment);
- TerminateUpLoad service (terminer un rapatriement);
- GetDataSetAttribute service (obtenir les attributs du jeu de données);
- Start service (démarrer);
- Stop service (arrêter);
- Resume service (reprise);
- MakeUsable service (rendre utilisable);
- GetTIAttribute service (obtenir attributs de l'invocation de tâches (TI));
- ChangeScope service (changer le champ d'accès);
- GetVAA Attribute service (obtenir attributs de Association Virtuelle Applications (VAA));
- Read service (lecture);
- Write service (écriture).

8.4.1.2 *Généralités sur les procédures*

Une fois dans DLMS, les services confirmés fournissent les moyens d'accéder aux facilités de DLMS. Le modèle de ces services est décrit dans la CEI 1334-4-41. Un sevice confirmé est un des services de la liste ci-dessus. Le comportement des DASE client ou serveur sont identiques vis-à-vis de ces services.

Du côté client, tout commence par la réception d'une primitive request (demande) d'un service confirmé. Le DASE client constitue le A_PDU de la primitive request du service confirmé en codant les paramètres soumis avec la primitive request. Il crypte ensuite l'A_PDU de la demande de service confirmé avec la clé de cryptage dédiée qui se trouve dans le contexte DLMS ou avec la clé de cryptage globale qui se trouve dans le contexte d'application. Une demande de DL_Data est émise avec les paramètres appropriés: Source_LSAP, Destination_LSAP, Destination_address et l'A_PDU du service confirmé.

Le DASE client attend la confirmation locale. Si elle est négative, une primitive confirm négative du service confirmé est émise vers l'utilisateur DLMS en spécifiant la raison de la faute. Si elle est positive, une temporisation est lancée pour la récupération des trames perdues.

A la réception d'une indication DL_Data contenant un A_PDU d'une demande de service confirmé, le DASE serveur vérifie les paramètres soumis et décrypte le A_PDU reçu avec la clé de cryptage appropriée. S'il y a une erreur, l'A_PDU est ignoré. Autrement, le DASE serveur émet la primitive indication adéquate vers le client DLMS et attend la réponse.

S'il reçoit une primitive response positive, le DASE serveur constitue un A_PDU de réponse avec les paramètres soumis. L'A_PDU est ensuite crypté avec la clé de cryptage dédiée et une demande de DL_Data est émise vers la couche liaison de données.

These services are listed as follows:

- GetStatus service;
- GetNameList service;
- InitiateLoad service;
- LoadSegment service;
- TerminateLoad service;
- InitiateUpLoad service;
- UpLoadSegment service;
- TerminateUpLoad service;
- GetDataSetAttribute service;
- Start service;
- Stop service;
- Resume service;
- MakeUsable service;
- GetTIAttribute service;
- ChangeScope service;
- GetVAAAttribute service;
- Read service;
- Write service.

8.4.1.2 *Overview of the procedures*

Once in the DLMS environment, the confirmed services provide means to access the DLMS facilities. The model for these services is described in IEC 1334-4-41. A confirmed service is one of the services listed above. The client or server DASE behaviour is the same for all these services.

At the client end, all begins with the reception of a confirmed service request primitive. The client DASE forms the confirmed service request A_PDU by coding the parameters submitted with the service request. It then ciphers the confirmed service request A_PDU with the dedicated ciphering key found in the DLMS context or with the global ciphering key found in the application context. A_DL_Data request is issued with the proper parameters: Source_LSAP, Destination_LSAP, Destination_address and the Confirmed Service request A_PDU.

The client DASE waits for the local confirm. If it is negative, a negative confirmed service confirm is issued to the DLMS user, specifying the error cause. If it is positive, a timer is set to recover from lost frames.

On reception of a DL_Data indication containing a Confirmed Service Request A_PDU, the server DASE checks the submitted parameters and deciphers the incoming A_PDU with the proper ciphering key. If an error is encountered, the A_PDU is discarded. Otherwise, the server DASE issues the proper confirmed service indication to the DLMS user and waits for the response.

If a positive response primitive is received, the server DASE forms a confirmed service response A_PDU with the submitted parameters. The A_PDU is then ciphered with the dedicated ciphering key and a DL_Data request is issued to the data link layer.

S'il reçoit une primitive response négative, le DASE serveur constitue un A_PDU d'erreur de service confirmé avec les paramètres soumis. L'A_PDU est ensuite crypté avec la clé de cryptage dédiée et une demande DL_Data est émise vers la couche liaison de données.

Le DASE client décrypte l'A_PDU de la réponse du service confirmé et, quand elle est correctement constituée, il émet une confirmation positive. Dans l'autre cas, une confirmation négative est envoyée à l'utilisateur.

Le DASE client décrypte l'A_PDU d'erreur du service confirmé et émet une confirmation négative du service confirmé. Si le délai expire avant la réception d'un A_PDU de réponse, une confirmation négative est envoyée à l'utilisateur DLMS.

8.4.1.3 *Tables d'états des services confirmés*

Les tableaux 4 et 5 de changement d'état s'appliquent à chacun des services ci-dessus,et ils sont appliqués séparément dans chaque cas de demande de service. Le cas de demandes de service multiples et convergentes n'est pas pris en compte par la présente norme.

Tableau 4 – Table d'états de serveur de services confirmés

Initial state	Events	Actions	Final state
IDLE	DL_Data.ind (ConfServiceReq_PDU) and Deciphering () = Ok	Extract_Field (ConfServiceReq_APDU) ConfService.ind	CS.E
IDLE	DL_Data.ind (ConfServiceReq_PDU) and Deciphering () <> Ok	None	IDLE
CS.E	ConfService.rsp (+)	Build_APDU (Type = ConfServiceRsp_APDU, Cipher (Dedicated_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, ConfServiceRsp_APDU)	CS.W
CS.E	ConfService.rsp (–)	Build_APDU (Type = ConfServiceErr_APDU, Cipher (Dedicated_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, ConfServiceErr_APDU)	CS.W
CS.W	DL_Data.cnf	None	IDLE
NOTE – ConfService est une abréviation de Confirmed Service (service confirmé) (voir la liste des services en 8.4.1.1).			

If a negative response primitive is received, the server DASE forms a confirmed service error A_PDU with the submitted parameters. The A_PDU is then ciphered with the dedicated ciphering key and a DL_Data request is issued to the data link layer.

The client DASE deciphers the incoming confirmed service response A_PDU and, when it is properly formed, issues a positive confirmed service confirm. Otherwise a negative confirm is issued to the DLMS user.

The client DASE deciphers the incoming confirmed service error A_PDU and issues a negative confirmed service confirm. If the timer expires before receiving any response A_PDU, a negative confirm is issued to the DLMS user.

8.4.1.3 *Confirmed service state tables*

The state transition diagrams in tables 4 and 5 are applicable for each of the above services, and are applied separately to each instance of each service request. Multiple concurrent service request instances that are outstanding at any given instant in time are not considered in this standard.

Table 4 – Server confirmed service state table

Initial state	Events	Actions	Final state
IDLE	DL_Data.ind (ConfServiceReq_PDU) and Deciphering () = Ok	Extract_Field (ConfServiceReq_APDU) ConfService.ind	CS.E
IDLE	DL_Data.ind (ConfServiceReq_PDU) and Deciphering () <> Ok	None	IDLE
CS.E	ConfService.rsp (+)	Build_APDU (Type = ConfServiceRsp_APDU, Cipher (Dedicated_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, ConfServiceRsp_APDU)	CS.W
CS.E	ConfService.rsp (–)	Build_APDU (Type = ConfServiceErr_APDU , Cipher (Dedicated_Key, A_SDU)) DL_Data.req (>A_DirectoryF () !Destination_Title, !Source_Title, ConfServiceErr_APDU)	CS.W
CS.W	DL_Data.cnf	None	IDLE
NOTE – ConfService is an abbreviation for ConfirmedService (c.f. services listed in 8.4.1.1.).			