

# INTERNATIONAL STANDARD

# NORME INTERNATIONALE

---

**Industrial communication networks – Fieldbus specifications –  
Part 5-13: Application layer service definition – Type 13 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –  
Partie 5-13: Définition des services de la couche application – Éléments  
de type 13**



**THIS PUBLICATION IS COPYRIGHT PROTECTED**  
**Copyright © 2014 IEC, Geneva, Switzerland**

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office  
3, rue de Varembe  
CH-1211 Geneva 20  
Switzerland

Tel.: +41 22 919 02 11  
Fax: +41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)

#### **About the IEC**

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

#### **About IEC publications**

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

#### **IEC Catalogue - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)**

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

#### **IEC publications search - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)**

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

#### **IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)**

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

#### **Electropedia - [www.electropedia.org](http://www.electropedia.org)**

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

#### **IEC Glossary - [std.iec.ch/glossary](http://std.iec.ch/glossary)**

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

#### **IEC Customer Service Centre - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)**

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: [csc@iec.ch](mailto:csc@iec.ch).

---

#### **A propos de l'IEC**

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

#### **A propos des publications IEC**

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

#### **Catalogue IEC - [webstore.iec.ch/catalogue](http://webstore.iec.ch/catalogue)**

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

#### **Recherche de publications IEC - [www.iec.ch/searchpub](http://www.iec.ch/searchpub)**

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

#### **IEC Just Published - [webstore.iec.ch/justpublished](http://webstore.iec.ch/justpublished)**

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

#### **Electropedia - [www.electropedia.org](http://www.electropedia.org)**

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

#### **Glossaire IEC - [std.iec.ch/glossary](http://std.iec.ch/glossary)**

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

#### **Service Clients - [webstore.iec.ch/csc](http://webstore.iec.ch/csc)**

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: [csc@iec.ch](mailto:csc@iec.ch).



IEC 61158-5-13

Edition 2.0 2014-08

# INTERNATIONAL STANDARD

# NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –  
Part 5-13: Application layer service definition – Type 13 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –  
Partie 5-13: Définition des services de la couche application – Éléments  
de type 13**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

PRICE CODE **XA**  
CODE PRIX

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-1738-2

**Warning! Make sure that you obtained this publication from an authorized distributor.  
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

## CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
1.1 General.....	7
1.2 Specifications.....	8
1.3 Conformance.....	8
2 Normative references .....	8
3 Terms, definitions, symbols, abbreviations and conventions .....	9
3.1 ISO/IEC 7498-1 terms .....	10
3.2 ISO/IEC 8822 terms .....	10
3.3 ISO/IEC 9545 terms .....	10
3.4 Fieldbus application-layer specific definitions .....	10
3.5 Abbreviations and symbols.....	17
3.6 Conventions .....	18
4 Concepts.....	20
4.1 Common concepts.....	20
4.2 Type specific concepts .....	21
5 Data type ASE.....	22
5.1 General.....	22
5.2 Formal definition of data type objects .....	22
5.3 FAL defined data types.....	22
5.4 Data type ASE service specification .....	28
6 Type 13 communication model specification .....	28
6.1 ASEs.....	28
6.2 ARs.....	50
6.3 Summary of FAL classes.....	57
6.4 Permitted FAL services by AREP role.....	58
Bibliography.....	59
Figure 1 – The ASEs of Type 13 .....	22
Figure 2 – The AR ASE conveys APDUs between APs .....	46
Table 1 – Overall structure of the OD.....	21
Table 2 – Ident service .....	31
Table 3 – Status service .....	32
Table 4 – Sync service.....	33
Table 5 – NMT-req-invite service .....	34
Table 6 – NMT-state-command service .....	35
Table 7 – NMT-info service .....	36
Table 8 – SDO-write .....	38
Table 9 – SDO-write-mult.....	39
Table 10 – SDO-read .....	41
Table 11 – SDO-read-mult .....	42
Table 12 – SDO-abort.....	43

Table 13 – PDO-transfer ..... 45

Table 14 – Conveyance of service primitives by AREP role..... 47

Table 15 – Valid combinations of AREP roles involved in an AR ..... 47

Table 16 – FAL class summary ..... 57

Table 17 – Services by AREP role ..... 58

# INTERNATIONAL ELECTROTECHNICAL COMMISSION

---

## INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

### Part 5-13: Application layer service definition – Type 13 elements

#### FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of 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, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). 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. 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 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 IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

NOTE Combinations of protocol types are specified in IEC 61784-1 and IEC 61784-2.

International Standard IEC 61158-5-13 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision. The main changes with respect to the previous edition are listed below:

- addition of synchronization feature,
- corrections, and
- editorial improvements.

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

FDIS	Report on voting
65C/763/FDIS	65C/773/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.

This publication has been drafted in accordance with ISO/IEC Directives, Part 2.

A list of all the parts of the IEC 61158 series, under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under <http://webstore.iec.ch> in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

## INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC 61158-1.

The application service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This standard defines the application service characteristics that fieldbus applications and/or system management may exploit.

Throughout the set of fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the application layer service defined in this standard is a conceptual architectural service, independent of administrative and implementation divisions.



## **INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –**

### **Part 5-13: Application layer service definition – Type 13 elements**

## **1 Scope**

### **1.1 General**

The fieldbus Application Layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs.”

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 13 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the different Types of the fieldbus Application Layer in terms of

- a) an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,
- b) the primitive actions and events of the service;
- c) the parameters associated with each primitive action and event, and the form which they take; and
- d) the interrelationship between these actions and events, and their valid sequences.

The purpose of this standard is to define the services provided to

- 1) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and
- 2) Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

This standard specifies the structure and services of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented Application Service Elements (ASEs) and a Layer Management Entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can

send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

## 1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various Types of IEC 61158, and the corresponding protocols standardized in subparts of IEC 61158-6.

This specification may be used as the basis for formal Application Programming-Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this specification, including

- a) the sizes and octet ordering of various multi-octet service parameters, and
- b) the correlation of paired request and confirm, or indication and response, primitives.

## 1.3 Conformance

This standard does not specify individual implementations or products, nor does it constrain the implementations of application layer entities within industrial automation systems.

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfill any given Type of application layer services as defined in this standard.

## 2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61158-1:2014, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-13, *Industrial communication networks – Fieldbus specifications – Part 3-13: Data-link layer service definition – Type 13 elements*

IEC 61158-4-13, *Industrial communication networks – Fieldbus specifications – Part 4-13: Data-link layer protocol specification – Type 13 elements*

IEC 61158-6 (all parts), *Industrial communication networks – Fieldbus specifications – Part 6: Application layer protocol specification*

IEC 61158-6-13, *Industrial communication networks – Fieldbus specifications – Part 6-13: Application layer protocol specification – Type 13 elements*

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

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

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

IEEE 754, *IEEE Standard for Floating-Point Arithmetic*

### **3 Terms, definitions, symbols, abbreviations and conventions**

For the purposes of this document, the following terms, definitions, symbols, abbreviations and conventions apply

**3.1 ISO/IEC 7498-1 terms**

- 3.1.1 application entity**
- 3.1.2 application process**
- 3.1.3 application protocol data unit**
- 3.1.4 application service element**
- 3.1.5 application entity invocation**
- 3.1.6 application process invocation**
- 3.1.7 application transaction**
- 3.1.8 real open system**
- 3.1.9 transfer syntax**

**3.2 ISO/IEC 8822 terms**

- 3.2.1 abstract syntax**
- 3.2.2 presentation context**

**3.3 ISO/IEC 9545 terms**

- 3.3.1 application-association**
- 3.3.2 application-context**
- 3.3.3 application context name**
- 3.3.4 application-entity-invocation**
- 3.3.5 application-entity-type**
- 3.3.6 application-process-invocation**
- 3.3.7 application-process-type**
- 3.3.8 application-service-element**
- 3.3.9 application control service element**

**3.4 Fieldbus application-layer specific definitions**

For the purposes of this part of IEC 61158, the following terms and definitions apply.

**3.4.1**

**application**

function or data structure for which data is consumed or produced

**3.4.2**

**application objects**

multiple object classes that manage and provide a run time exchange of messages across the network and within the network device

**3.4.3**

**application process**

part of a distributed application on a network, which is located on one device and unambiguously addressed

**3.4.4**

**application process identifier**

distinguishes multiple application processes used in a device

### 3.4.5

#### **application process object**

component of an application process that is identifiable and accessible through an FAL application relationship

Note 1 to entry: Application process object definitions are composed of a set of values for the attributes of their class (see the definition for Application Process Object Class Definition). Application process object definitions may be accessed remotely using the services of the FAL Object Management ASE. FAL Object Management services can be used to load or update object definitions, to read object definitions, and to dynamically create and delete application objects and their corresponding definitions.

### 3.4.6

#### **application process object class**

class of application process objects defined in terms of the set of their network-accessible attributes and services

### 3.4.7

#### **application relationship**

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation

Note 1 to entry: This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities.

### 3.4.8

#### **application relationship application service element**

application-service-element that provides the exclusive means for establishing and terminating all application relationships

### 3.4.9

#### **application relationship endpoint**

context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

Note 1 to entry: Each application process involved in the application relationship maintains its own application relationship endpoint.

### 3.4.10

#### **attribute**

description of an externally visible characteristic or feature of an object

Note 1 to entry: The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behavior of an object. Attributes are divided into class attributes and instance attributes.

### 3.4.11

#### **behavior**

indication of how an object responds to particular events

### 3.4.12

#### **channel**

single physical or logical link of an input or output application object of a server to the process

### 3.4.13

#### **class**

set of objects, all of which represent the same kind of system component

Note 1 to entry: A class is a generalisation of an object; a template for defining variables and methods. All objects in a class are identical in form and behavior, but usually contain different data in their attributes.

**3.4.14**

**class attributes**

attribute that is shared by all objects within the same class

**3.4.15**

**class code**

unique identifier assigned to each object class

**3.4.16**

**class specific service**

service defined by a particular object class to perform a required function which is not performed by a common service

Note 1 to entry: A class specific object is unique to the object class which defines it.

**3.4.17**

**client**

- a) object which uses the services of another (server) object to perform a task
- b) initiator of a message to which a server reacts

**3.4.18**

**controlled node**

All nodes except one (this is the Managing Node) in a Type 13 network are of this type

**3.4.19**

**connection**

logical binding between application objects that may be within the same or different devices

Note 1 to entry: Connections may be either point-to-point or multipoint.

**3.4.20**

**connection channel**

description of a connection between a sink and a source of data items

**3.4.21**

**connection path**

an octet stream that defines the application object to which a connection instance applies

**3.4.22**

**connection point**

buffer which is represented as a subinstance of an Assembly object

**3.4.23**

**consume**

act of receiving data from a producer

**3.4.24**

**consumer**

node or sink that is receiving data from a producer

**3.4.25**

**consuming application**

application that consumes data

**3.4.26**

**control commands**

action invocations transferred from client to server to clear outputs, freeze inputs and/or synchronise outputs

**3.4.27****conveyance path**

unidirectional flow of APDUs across an application relationship

**3.4.28****cyclic**

repetitive in a regular manner

**3.4.29****data consistency**

means for coherent transmission and access of the input- or output-data object between and within client and server

**3.4.30****device**

physical hardware connected to the link

Note 1 to entry: A device may contain more than one node.

**3.4.31****device profile**

a collection of device dependent information and functionality providing consistency between similar devices of the same device type

**3.4.32****diagnosis information**

all data available at the server for maintenance purposes

**3.4.33****end node**

producing or consuming node

**3.4.34****endpoint**

one of the communicating entities involved in a connection

**3.4.35****error**

discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

**3.4.36****error class**

general grouping for related error definitions and corresponding error codes

**3.4.37****error code**

identification of a specific type of error within an error class

**3.4.38****event**

an instance of a change of conditions

**3.4.39****FIFO variable**

a Variable Object class, composed of a set of homogeneously typed elements, where the first written element is the first element that can be read

Note 1 to entry: On the fieldbus only one, complete element can be transferred as a result of one service invocation.

#### **3.4.40**

##### **frame**

denigrated synonym for DLPDU

#### **3.4.41**

##### **group**

a) <general> general term for a collection of objects

Specific uses:

b) <addressing> when describing an address, an address that identifies more than one entity

#### **3.4.42**

##### **invocation**

act of using a service or other resource of an application process

Note 1 to entry: Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation. Also for service invocations, an Invoke ID may be used to unambiguously identify the service invocation and differentiate it from other outstanding service invocations.

#### **3.4.43**

##### **index**

address of an object within an application process

#### **3.4.44**

##### **instance**

the actual physical occurrence of an object within a class that identifies one of many objects within the same object class

EXAMPLE California is an instance of the object class US-state.

Note 1 to entry: The terms object, instance, and object instance are used to refer to a specific instance.

#### **3.4.45**

##### **instance attributes**

attribute that is unique to an object instance and not shared by the object class

#### **3.4.46**

##### **instantiated**

object that has been created in a device

#### **3.4.47**

##### **logical device**

a certain FAL class that abstracts a software component or a firmware component as an autonomous self-contained facility of an automation device

#### **3.4.48**

##### **manufacturer ID**

identification of each product manufacturer by a unique number

#### **3.4.49**

##### **management information**

network-accessible information that supports managing the operation of the fieldbus system, including the application layer

Note 1 to entry: Managing includes functions such as controlling, monitoring, and diagnosing.



**3.4.50****message router**

object within a node that distributes messaging requests to appropriate application objects

**3.4.51****managing node**

One node of a type 13 network must act as a managing node.

**3.4.52****multi-point connection**

connection from one node to many

Note 1 to entry: Multipoint connections allow messages from a single producer to be received by many consumer nodes.

**3.4.53****network**

a set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

**3.4.54****node**

host of one AP, identified by a unique DLCEP-address

**3.4.55****object**

abstract representation of a particular component within a device, usually a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behavior

**3.4.56****object dictionary**

Collection of definitions. communication specific attributes and parameters, and application dependent data

**3.4.57****object specific service**

service unique to the object class which defines it

**3.4.58****originator**

client responsible for establishing a connection path to the target

**3.4.59****physical device**

automation or other network device

**3.4.60****point-to-point connection**

connection that exists between exactly two application objects

**3.4.61****pre-established AR endpoint**

AR endpoint that is placed in an established state during configuration of the AEs that control its endpoints

**3.4.62**

**process data**

object(s) which are already pre-processed and transferred cyclically for the purpose of information or further processing

**3.4.63**

**produce**

act of sending data to be received by a consumer

**3.4.64**

**producer**

node that is responsible for sending data

**3.4.65**

**property**

general term for descriptive information about an object

**3.4.66**

**provider**

source of a data connection

**3.4.67**

**publisher**

role of an AR endpoint that transmits APDUs onto the fieldbus for consumption by one or more subscribers

Note 1 to entry: A publisher may not be aware of the identity or the number of subscribers.

**3.4.68**

**publishing manager**

role of an AR endpoint in which it issues one or more confirmed service request APDUs to a publisher to request the publisher to publish a specified object

Note 1 to entry: Two types of publishing managers are defined by this standard, pull publishing managers and push publishing managers, each of which is defined separately.

**3.4.69**

**push publisher**

type of publisher that publishes an object in an unconfirmed service request APDU

**3.4.70**

**push publishing manager**

type of publishing manager that requests that a specified object be published using an unconfirmed service

**3.4.71**

**push subscriber**

type of subscriber that recognizes received unconfirmed service request APDUs as published object data

**3.4.72**

**server**

a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

b) object which provides services to another (client) object

**3.4.73**

**service**

operation or function than an object and/or object class performs upon request from another object and/or object class

**3.4.74****subscriber**

role of an AREP in which it receives APDUs produced by a publisher

**3.4.75****target**

end-node to which a connection is established

**3.5 Abbreviations and symbols**

AE	Application entity
AL	Application layer
ALME	Application layer management entity
ALP	Application layer protocol
APO	Application object
AP	Application process
APDU	Application protocol data unit
AR	Application relationship
AREP	Application relationship end point
ASCII	American Standard Code for Information Interchange
ASE	Application service element
BNB-PEC	Buffered network-scheduled bi-directional pre-established connection
BNU-PEC	Buffered network-scheduled uni-directional pre-established connection
CID	Connection ID
CN	Controlled node
Cnf	Confirmation
DL-	(as a prefix) data-link-
DLCEP	Data-link connection end point
DLL	Data-link layer
DLM	Data-link-management
DLSAP	Data-link service access point
DLSDU	DL-service-data-unit
DNS	Domain name service
FAL	Fieldbus application layer
Ind	Indication
IP	Internet protocol (see RFC 791)
MN	Managing node
NMT	Network management
OD	Object dictionary
PDO	Process data object
PReq	PollRequest (Type 13 frame type)
Pres	PollRespon (Type 13 frame type)
QUB-CL	Queued user-triggered bi-directional connectionless
QUB-COS	Queued user-triggered bi-directional connection-oriented with segmentation
QUU	Queued user-triggered uni-directional

Req	Request
RPDO	Receive process data object
Rsp	Response
SDO	Service data object
TPDO	Transmit process data object

### 3.6 Conventions

#### 3.6.1 Overview

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of two parts, its class specification, and its service specification.

The class specification defines the attributes of the class. The access to these attributes is beyond the scope of this document except where specified. The service specification defines the services that are provided by the ASE.

#### 3.6.2 General conventions

This standard uses the descriptive conventions given in ISO/IEC 10731.

#### 3.6.3 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is shown below:

<b>FAL ASE:</b>		<b>ASE name</b>
<b>CLASS:</b>		<b>Class name</b>
<b>CLASS ID:</b>		<b>#</b>
<b>PARENT CLASS:</b>		<b>Parent class name</b>
<b>ATTRIBUTES:</b>		
1	(o)	Key Attribute: numeric identifier
2	(o)	Key Attribute: name
3	(m)	Attribute: attribute name(values)
4	(m)	Attribute: attribute name(values)
4.1	(s)	Attribute: attribute name(values)
4.2	(s)	Attribute: attribute name(values)
4.3	(s)	Attribute: attribute name(values)
5.	(c)	Constraint: constraint expression
5.1	(m)	Attribute: attribute name(values)
5.2	(o)	Attribute: attribute name(values)
6	(m)	Attribute: attribute name(values)
6.1	(s)	Attribute: attribute name(values)
6.2	(s)	Attribute: attribute name(values)
<b>SERVICES:</b>		
1	(o)	OpsService: service name
2	(c)	Constraint: constraint expression
2.1	(o)	OpsService: service name
3	(m)	MgtService: service name

(1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.

- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is not used for Type 13 elements.
- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

NOTE The parent-class "TOP" indicates that the class being defined is an initial class definition. The parent class TOP is used as a starting point from which all other classes are defined. The use of TOP is reserved for classes defined by this standard.

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
  - a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
  - b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
  - c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify
    - i) fields of a structured attribute (4.1, 4.2, 4.3),
    - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).
    - iii) the selection fields of a choice type attribute (6.1 and 6.2).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
  - a) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.
  - b) The label "OpsService" designates an operational service (1).
  - c) The label "MgtService" designates a management service (2).
  - d) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

### 3.6.4 Conventions for service definitions

#### 3.6.4.1 General

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

#### 3.6.4.2 Service parameters

Service primitives are used to represent service user/service provider interactions (ISO/IEC 10731). They convey parameters which indicate information available in the user/provider interaction. In any particular interface, not all parameters need be explicitly stated.

The service specifications of this standard use a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

- 1) parameter name,
- 2) request primitive,
- 3) indication primitive,
- 4) response primitive, and
- 5) confirm primitive.

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column:

- M parameter is mandatory for the primitive
- U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.
- C parameter is conditional upon other parameters or upon the environment of the service user.
- (blank) parameter is never present.
- S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

- a) a parameter-specific constraint:
  - “(=)” indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.
- b) an indication that some note applies to the entry:
  - “(n)” indicates that the following note "n" contains additional information pertaining to the parameter and its use.

## Service procedures

The procedures are defined in terms of

- the interactions between application entities through the exchange of fieldbus Application Protocol Data Units, and
- the interactions between an application layer service provider and an application layer service user in the same system through the invocation of application layer service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the fieldbus application layer.

## 4 Concepts

### 4.1 Common concepts

IEC 61158-1, Clause 9 is incorporated by reference, except as specifically overridden in 4.2.

## 4.2 Type specific concepts

### 4.2.1 Managing Node and Controlled Nodes

The node managing the permission to send messages to the Ethernet is called the Managing Node (MN). Only the MN may send messages independently – i.e. not as a response to a received message. All available nodes in the network are configured on the MN. Only one active MN is permitted in the network.

All other nodes transmit only within communication slots assigned by the MN. They are thus called Controlled Nodes (CN). CNs are passive bus nodes which only send when requested by the MN. There are two types of CNs:

- Isochronous CNs are able to communicate both via network-scheduled and user-triggered ARs.
- Async-only CNs communicate only via user-triggered ARs.

The sequence and the assignment of time slots for network-scheduled and user-triggered interactions is specified in IEC 61158-3-13 and IEC 61158-4-13.

### 4.2.2 Node, AP, and Object dictionary

Each node hosts exactly one AP. All APOs for this AP are collected in the so called Object dictionary (OD). The overall structure of the OD is as described in Table 1.

**Table 1 – Overall structure of the OD**

Area	Contents
Data type area	Definition of the data types
Communication profile area	Contains the communication specific parameters for the Type 13 network. These entries are common to all devices
Manufacturer specific area	Definition of manufacturer specific variables
Device profile area	Definition of the variables defined in a device profile (not in the scope of this standard)
Reserved area	Reserved for future use

### 4.2.3 Object management ASE

There is no object management ASE defined for Type 13.

### 4.2.4 Timeliness

The "timeliness" feature is not implemented in the Type 13 FAL. The monitoring and control of the cycle timing is managed by the DLL, see IEC 61158-3-13 and IEC 61158-4-13.

### 4.2.5 APO ASEs

The FAL ASEs of Type 13 and their interrelationships are described in Figure 1.

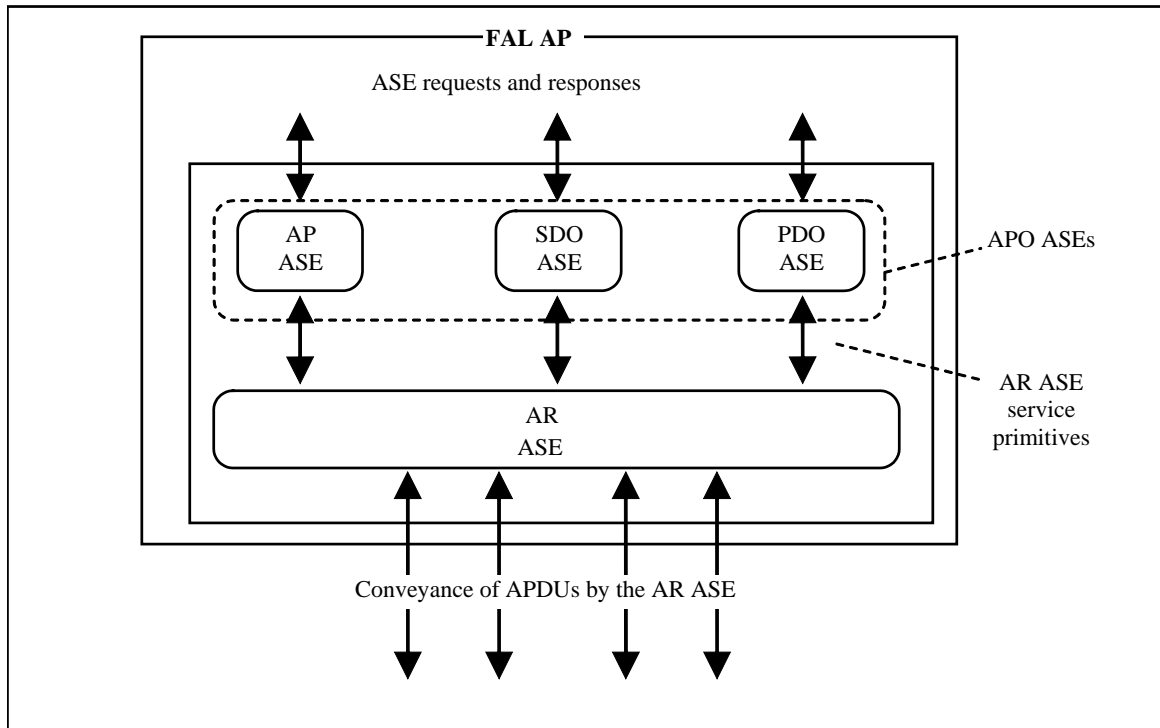


Figure 1 – The ASEs of Type 13

## 5 Data type ASE

### 5.1 General

IEC 61158-1, 10.1 is incorporated by reference with one restriction: Only nesting level of 1 (see IEC 61158-1, 10.1.4.4) is supported.

### 5.2 Formal definition of data type objects

IEC 61158-1, 10.2 is incorporated by reference.

### 5.3 FAL defined data types

#### 5.3.1 Fixed length types

##### 5.3.1.1 Boolean types

CLASS:		Data type	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	1
2	Data type name	=	Boolean
3	Format	=	FIXED LENGTH
3.1	Octet length	=	1

This data type expresses a Boolean data type with the values TRUE and FALSE.



### 5.3.1.2 Date/Time types

#### 5.3.1.2.1 TimeOfDay

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 12
2	Data type name	= TimeOfDay
3	Format	= FIXED LENGTH
3.1	Octet length	= 6

This data type is composed of two elements of unsigned values and expresses the time of day and the date. The first element is an Unsigned32 data type and gives the time after the midnight in milliseconds. The second element is an Unsigned16 data type and gives the date counting the days from January 1, 1984.

#### 5.3.1.2.2 TimeDifference

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 13
2	Data type name	= TimeDifference
3	Format	= FIXED LENGTH
3.1	Octet length	= 4 or 6

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The optional second element is an Unsigned16 data type that provides the difference in days.

### 5.3.1.3 Numeric types

#### 5.3.1.3.1 Real32

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 8
2	Data type name	= Real32
3	Format	= FIXED LENGTH
3.1	Octet length	= 4

This type has a length of four octets. The format for Real32 is that defined by IEEE 754 as single precision.

#### 5.3.1.3.2 Real64

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 17
2	Data type name	= Real64
3	Format	= FIXED LENGTH
3.1	Octet length	= 8

This type has a length of eight octets. The format for Real64 is that defined by IEEE 754 as double precision.

**5.3.1.3.3 Integer types**

**5.3.1.3.3.1 Integer8**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 2
2	Data type name	= Integer8
3	Format	= FIXED LENGTH
4.1	Octet length	= 1

This integer type is a two's complement binary number with a length of one octet.

**5.3.1.3.3.2 Integer16**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 3
2	Data type name	= Integer16
3	Format	= FIXED LENGTH
3.1	Octet length	= 2

This integer type is a two's complement binary number with a length of two octets.

**5.3.1.3.3.3 Integer24**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 16
2	Data type name	= Integer24
3	Format	= FIXED LENGTH
3.1	Octet length	= 3

This integer type is a two's complement binary number with a length of three octets.

**5.3.1.3.3.4 Integer32**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 4
2	Data type name	= Integer32
3	Format	= FIXED LENGTH
3.1	Octet length	= 4

This integer type is a two's complement binary number with a length of four octets.

**5.3.1.3.3.5 Integer40**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 18
2	Data type name	= Integer40
3	Format	= FIXED LENGTH
3.1	Octet length	= 5

This integer type is a two's complement binary number with a length of five octets.

**5.3.1.3.3.6 Integer48**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 19
2	Data type name	= Integer48
3	Format	= FIXED LENGTH
3.1	Octet length	= 6

This integer type is a two's complement binary number with a length of six octets.

**5.3.1.3.3.7 Integer56**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 20
2	Data type name	= Integer56
3	Format	= FIXED LENGTH
3.1	Octet length	= 7

This integer type is a two's complement binary number with a length of seven octets.

**5.3.1.3.3.8 Integer64**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 21
2	Data type name	= Integer64
3	Format	= FIXED LENGTH
3.1	Octet length	= 8

This integer type is a two's complement binary number with a length of eight octets.

**5.3.1.3.4 Unsigned types****5.3.1.3.4.1 Unsigned8**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 5
2	Data type name	= Unsigned8
3	Format	= FIXED LENGTH
3.1	Octet length	= 1

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of one octet.

**5.3.1.3.4.2 Unsigned16**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 6
2	Data type name	= Unsigned16
3	Format	= FIXED LENGTH
3.1	Octet length	= 2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets.

**5.3.1.3.4.3 Unsigned24**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	22
2	Data type name	=	Unsigned24
3	Format	=	FIXED LENGTH
3.1	Octet length	=	3

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of three octets.

**5.3.1.3.4.4 Unsigned32**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	7
2	Data type name	=	Unsigned32
3	Format	=	FIXED LENGTH
3.1	Octet length	=	4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of four octets.

**5.3.1.3.4.5 Unsigned40**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	24
2	Data type name	=	Unsigned40
3	Format	=	FIXED LENGTH
3.1	Octet length	=	5

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of five octets.

**5.3.1.3.4.6 Unsigned48**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	25
2	Data type name	=	Unsigned48
3	Format	=	FIXED LENGTH
3.1	Octet length	=	6

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of six octets.

**5.3.1.3.4.7 Unsigned56**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	26
2	Data type name	=	Unsigned56
3	Format	=	FIXED LENGTH
3.1	Octet length	=	7

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of seven octets.

**5.3.1.3.4.8 Unsigned64**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	27
2	Data type name	=	Unsigned64
3	Format	=	FIXED LENGTH
3.1	Octet length	=	8

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of eight octets.

**5.3.2 String types****5.3.2.1 OctetString**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	10
2	Data type name	=	OctetString
3	Format	=	STRING
3.1	Octet length	=	1 to n

An OctetString is an ordered sequence of octets, numbered from 1 to n. For the purposes of discussion, octet 1 of the sequence is referred to as the first octet.

NOTE IEC 61158-6-13 defines the order of transmission.

**5.3.2.2 VisibleString**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	9
2	Data type name	=	VisibleString
3	Format	=	STRING
3.1	Octet length	=	1 to n

This type is defined as the ISO/IEC 646 string type.

### 5.3.2.3 UnicodeString

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	11
2	Data type name	=	UnicodeString
3	Format	=	STRING
3.1	Octet length	=	1 to n

This type is defined as the UNICODE string type.

### 5.4 Data type ASE service specification

There are no operational services defined for the type object.

## 6 Type 13 communication model specification

### 6.1 ASEs

#### 6.1.1 Application process ASE

##### 6.1.1.1 Overview

This standard models a fieldbus Application Process (AP). Fieldbus application processes represent the information and processing resources of a system that can be accessed through FAL services.

The Application Service Element in the FAL that provides these services is called an Application Process ASE. In the AP ASE, the AP is modeled and accessed as an APO with a standardized and predefined identifier.

##### 6.1.1.2 AP class specification

###### 6.1.1.2.1 Formal model

The AP class specifies the attributes and services defined for application processes. Its parent class "top" indicates the top of the FAL class tree.

<b>ASE:</b>		<b>AP ASE</b>
<b>CLASS:</b>		<b>AP</b>
<b>CLASS ID:</b>		<b>not used</b>
<b>PARENT CLASS:</b>		<b>TOP</b>
<b>ATTRIBUTES:</b>		
1	(m) Attribute:	NMT-status
2	(m) Attribute:	Type 13 fieldbus-version
3	(m) Attribute:	feature-flags
4	(m) Attribute:	MTU
5	(m) Attribute:	cycle-timing
6	(m) Attribute:	identity
7	(m) Attribute:	verify-configuration
8	(o) Attribute:	application-software-version
9	(c) Attribute:	IP-address
10	(c) Attribute:	host-name
11	(o) Attribute:	vendor-specific-extension
12	(m) Attribute:	static-error
13	(o) Attribute:	error-history
14	(m) Attribute:	sync-control
15	(m) Attribute:	sync-status
<b>SERVICES:</b>		
1	(m) OpsService:	Ident
2	(m) OpsService:	Status
3	(m) OpsService:	Sync
4	(m) OpsService:	NMT-req-invite
5	(m) OpsService:	NMT-state-command
6	(m) OpsService:	NMT-info

#### 6.1.1.2.2 Attributes

##### **NMT-status**

This attribute reports the current status of the CN's NMT state machine.

##### **Type 13 fieldbus-version**

This attribute indicates the Type 13 fieldbus-version to which the CN conforms.

##### **feature-flags**

This attribute holds information about the device's behavior in the Type 13 fieldbus cycle, the implementation of NMT features and its capability of acting as a router. The detailed specification of this attribute is included in IEC 61158-6-13.

##### **MTU**

This attribute reports the size of the largest IP frame that can be transmitted over the network, including the size of the transport header.

##### **cycle-timing**

This attribute specifies the timing performance of the CN during the isochronous phase of the Type 13 fieldbus cycle.

##### **identity**

This attribute specifies the CN's type and vendor ID. As an optional addition this attribute may also contain the product code, revision number, and serial number of the CN.

##### **verify-configuration**

This attribute holds the configuration date and time of the node.

##### **application-software-version**

This optional attribute holds date and time of the last update for the CN's application software.

**IP-address**

This attribute provides IP-address informations of the CN. It is only used if IP is supported by the device.

**host-name**

This attribute reports the current DNS host name of the CN. It is only used if IP is supported by the device.

**vendor-specific-extension**

This optional attribute is used for vendor specific purpose.

**static-error**

This attribute indicates a static error on the CN and its type.

**error-history**

This attribute documents past error events. It holds the error type, its occurrence time, and additional information.

**sync-control**

This attribute holds data necessary for the configuration/synchronization of a CN with a time-triggered PRes.

**sync-status**

This attribute holds data necessary for the synchronization of a CN with a time-triggered PRes.

**6.1.1.2.3 Services****Ident**

This service is used to retrieve relevant characteristics of the CNs.

**Status**

This service is used to query the current status and error data from CNs.

**Sync**

This service is used to configure a CN with a time-triggered PRes.

**NMT-req-invite**

With this service the MN invites a CN to send a request for a NMT command.

**NMT-state-command**

This service is used to control the operational status of the CNs.

**NMT-info**

This service is used to publish system relevant information to the CNs in the network.

**6.1.1.3 AP ASE service specification****6.1.1.3.1 Supported services**

Subclause 6.1.1.3 contains the definition of services that are unique to this ASE. The services defined for this ASE are

- Ident
- Status
- Sync
- NMT-req-invite
- NMT-state-command
- NMT-info services



### 6.1.1.3.2 Ident service

#### 6.1.1.3.2.1 Service overview

The Ident service is used by the MN to identify configured but unrecognized CNs at system start-up or after loss of communication. The service may be used after start-up to query a CN's setup information.

#### 6.1.1.3.2.2 Service primitives

The service parameters for this service are shown in Table 2.

**Table 2 – Ident service**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
Result(+)			S	S (=)
NMT-status			M	M (=)
Type 13 fieldbus-version			M	M (=)
feature-flags			M	M (=)
cycle-timing			M	M (=)
identity			M	M (=)
verify-configuration			M	M (=)
application-software-version			M	M (=)
IP-address			M	M (=)
host-name			M	M (=)
vendor-specific-extensions			M	M (=)
Result(-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

#### **Argument**

The argument conveys the service specific parameters of the service request.

#### **Result(+)**

This selection type parameter indicates that the service request succeeded.

#### **NMT-status**

This parameter specifies the value of the NMT-status attribute of the AP.

#### **Type 13 fieldbus-version**

This parameter specifies the value of the Type 13 fieldbus-version attribute of the AP.

#### **feature-flags**

This parameter specifies the value of the feature-flags attribute of the AP.

#### **cycle-timing**

This parameter specifies the value of the cycle-timing attribute of the AP.

#### **identity**

This parameter specifies the value of the identity attribute of the AP.

**verify-configuration**

This parameter specifies the value of the verify-configuration attribute of the AP.

**application-software-version**

This parameter specifies the value of the application-software-version attribute of the AP. The parameter is to be filled with zeros if the attribute is not in use.

**IP-address**

This parameter specifies the value of the IP-address attribute of the AP. The parameter is to be filled with zeros if the attribute is not in use.

**host-name**

This parameter specifies the value of the host-name attribute of the AP. The parameter is to be filled with zeros if the attribute is not in use.

**vendor-specific-extensions**

This parameter specifies the value of the vendor-specific-extensions attribute of the AP. The parameter is to be filled with zeros if the attribute is not in use.

**Result(-)**

This selection type parameter indicates that the service request failed.

**6.1.1.3.2.3 Service procedure**

This service procedure is a sequence of two successive unconfirmed services (as specified in IEC 61158-1, 9.6.2) in opposite directions.

APDUs containing the Ident response (i.e. the Result(+)) are sent via multicast ARs. The MN receives Ident responses; CNs may receive Ident responses if configured to do so.

The MN checks the incoming Ident response against its own informations about the requested node. In case of differences it tries to reconfigure the node with its data or, if not possible, it sends an error signal to its user.

**6.1.1.3.3 Status service**

**6.1.1.3.3.1 Service overview**

The Status service is used by the MN to query status information from the CN about current status and error data.

**6.1.1.3.3.2 Service primitives**

The service parameters for this service are shown in Table 3.

**Table 3 – Status service**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
Result(+)			S	S (=)
NMT-status			M	M (=)
static-error			M	M (=)
error-history			U	U (=)
Result(-)			S	S (=)

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

**Argument**

The argument conveys the service specific parameters of the service request.

**Result(+)**

This selection type parameter indicates that the service request succeeded.

**NMT-status**

This parameter specifies the value of the NMT-status attribute of the AP.

**static-error**

This parameter specifies the value of the static-error attribute of the AP.

**error-history**

This parameter specifies the value of the error-history attribute of the AP.

**Result(-)**

This selection type parameter indicates that the service request failed.

**6.1.1.3.3 Service procedure**

This service procedure is a sequence of two successive unconfirmed services (as specified in IEC 61158-1, 9.6.2) in opposite directions.

APDUs containing the Status response (i.e. the Result(+)) are sent via multicast ARs. The MN receives Status responses; CNs may receive Status responses if configured to do so.

**6.1.1.3.4 Sync service****6.1.1.3.4.1 Service overview**

The Sync service is used by the MN to configure/synchronize a CN with a time-triggered PRes.

**6.1.1.3.4.2 Service primitives**

The service parameters for this service are shown in Table 4.

**Table 4 – Sync service**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
sync-control	M	M (=)		
Result				
sync-status			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

**Argument**

The argument conveys the service specific parameters of the service request.

**sync-control**

This parameter specifies the value of the sync-control attribute of the AP.

**Result**

This selection type parameter indicates the result of the service request.

**Sync-status**

This parameter specifies the value of the sync-status attribute of the AP.

**6.1.1.3.4.3 Service procedure**

This service procedure is a sequence of two successive unconfirmed services (as specified in IEC 61158-1, 9.6.2) in opposite directions.

APDUs containing the Sync response (i.e. the Result) are sent via multicast ARs. The MN receives Sync responses; CNs may receive Sync responses if they support the time-triggered sending of PRes.

**6.1.1.3.5 NMT-req-invite service**

**6.1.1.3.5.1 Service overview**

The MN uses this service to find out what service the CN wants to be performed next by the MN.

**6.1.1.3.5.2 Service primitives**

The service parameters for this service are shown in Table 5.

**Table 5 – NMT-req-invite service**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
Result(+)			S	S (=)
command-ID			M	M (=)
target-node			M	M (=)
data			M	M (=)
Result(-)			S	S (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

**Argument**

The argument conveys the service specific parameters of the service request.

**Result(+)**

This selection type parameter indicates that the service request succeeded.

**command-ID**

This parameter specifies the ID of the requested command.

**target-node**

This parameter indicates the target node of the requested NMT command.

**data**

This parameter specifies command-specific data to be issued by the MN.

**Result(-)**

This selection type parameter indicates that the service request failed.

**6.1.1.3.5.3 Service procedure**

This service is triggered by a preceding "Ready-to-send"-notification issued by a CN. After that notification the MN addresses the calling CN at the next opportunity.

This service procedure is a sequence of two successive unconfirmed services (as specified in IEC 61158-1, 9.6.2) in opposite directions.

This service is always unicast.

**6.1.1.3.6 NMT-state-command service****6.1.1.3.6.1 Service overview**

The network management (NMT) is node-oriented and follows a master / slave relationship.

The function of the NMT master is carried out by the MN. CNs are administrated as NMT slaves by the master.

The MN uses NMT-state-command services to control the state machine(s) of the CN(s).

**6.1.1.3.6.2 Service primitives**

The service parameters for this service are shown in Table 6.

**Table 6 – NMT-state-command service**

Parameter name	Req	Ind
Argument		
AREP	M	M (=)
command-ID	M	M (=)
node-list	C	C (=)

**Argument**

The argument conveys the service specific parameters of the service request.

**command-ID**

This parameter identifies the NMT state command to be issued. A list of valid commands is given in IEC 61158-6-13.

**node-list**

This conditional parameter specifies the addresses of the nodes to be affected by the command. It is used only if the address is neither unicast nor broadcast.

**6.1.1.3.6.3 Service procedure**

This service is issued by the MN upon an internal request or upon an external request via the NMT-req-invite service. The addressed CN(s) change their state upon receipt of the state command.

**6.1.1.3.7 NMT-info service**

**6.1.1.3.7.1 Service overview**

This service is used to transmit complex status information in the form of bundles as well as to distribute system-relevant setup information from the MN to the CNs.

**6.1.1.3.7.2 Service primitives**

The service parameters for this service are shown in Table 7.

**Table 7 – NMT-info service**

Parameter name	Req	Ind
Argument		
AREP	M	M (=)
publish-node-list	C	C (=)
publish-time	C	C (=)

**Argument**

The argument conveys the service specific parameters of the service request.

**publish-node-list**

This parameter specifies a list of nodes being in the same state, alternatively it provides a list of all nodes indicating their respective states. Details are specified in IEC 61158-6-13.

**publish-time**

This parameter provides the system time.

**6.1.1.3.7.3 Service procedure**

This service is issued by the MN upon an internal request or upon an external request via NMT-req-invite. The service is either unicast or broadcast.

**6.1.2 Service data object ASE**

**6.1.2.1 Overview**

For all transfer types it is the client that takes the initiative for a transfer. The owner of the accessed object dictionary is the server of the service data object (SDO). Either the client or the server can take the initiative to abort the transfer of a SDO. All commands are confirmed. The remote result parameter indicates the success of the request. In case of a failure, an abort transfer request must be executed.

It is possible that the data has to be segmented because its size is greater than a single transferable element. This situation is handled by the QUB-COS (CmdL) ARPM. Details are given in IEC 61158-6-13.

## 6.1.2.2 SDO class specification

### 6.1.2.2.1 Formal model

The SDO ASE is specified by the following template:

<b>ASE:</b>		<b>SDO ASE</b>
<b>CLASS:</b>		<b>SDO</b>
<b>CLASS ID:</b>		<b>not used</b>
<b>PARENT CLASS:</b>		<b>TOP</b>
<b>ATTRIBUTES:</b>		
1	(m) Attribute:	index
2	(m) Attribute:	sub-index
3	(o) Attribute:	name
4	(m) Attribute:	command-ID
5	(c) Attribute:	segment-size
6	(c) Attribute:	data-size
<b>SERVICES:</b>		
1	(m) OpsService:	SDO-write
2	(o) OpsService:	SDO-write-mult
3	(m) OpsService:	SDO-read
4	(o) OpsService:	SDO-read-mult
5	(m) OpsService:	SDO-abort

#### 6.1.2.2.2 Attributes

##### **index**

This attribute specifies the index of the OD entry to be accessed.

##### **sub-index**

This attribute specifies the sub-index of the OD entry to be accessed.

##### **name**

This optional attribute specifies the name of the OD entry to be accessed.

##### **command-ID**

This attribute specifies the type of read / write operation

##### **segment-size**

This conditional attribute specifies the length of segment data.

##### **data-size**

This conditional attribute specifies the length of the transferred data block.

#### 6.1.2.2.3 Services

##### **SDO-write**

This service is used to write data to one entry of the server's object dictionary.

##### **SDO-write-mult**

This service is used to write data to multiple entries of the server's object dictionary.

##### **SDO-read**

This service is used to read data from one entry of the server's object dictionary.

##### **SDO-read-mult**

This service is used to read data from multiple entries of the server's object dictionary.

##### **SDO-abort**

The service is used to abort a SDO communication.

**6.1.2.3 SDO ASE specification**

**6.1.2.3.1 Supported services**

Subclause 6.1.2.3 contains the definition of services that are unique to this ASE. The services defined for this ASE are

- SDO-write
- SDO-write-mult
- SDO-read
- SDO-read-mult
- SDO-abort

**6.1.2.3.2 SDO-write**

**6.1.2.3.2.1 Service overview**

Using the SDO-write service the client uploads data to the server (owner of the object dictionary), affecting one entry of its object dictionary.

**6.1.2.3.2.2 Service primitives**

The service parameters for this service are shown in Table 8.

**Table 8 – SDO-write**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
segment-size	C	C (=)		
data-size	C	C (=)		
OD-identifier	M	M (=)		
payload-data	M	M (=)		
Result(+)			S	S (=)
Result(-)			S	S (=)
error-info			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

**Argument**

The argument conveys the service specific parameters of the service request.

**command-ID**

This parameter indicates how the affected OD entry is to be identified. The identification is done either by index and subindex or by index alone. In the latter case the service addresses all sub-indices (except sub-index 0) of the indicated index. Details are given in IEC 61158-6-13. The length of the payload data must confirm to the length of data for all sub-indices and all sub-indices must be writable.



**segment-size**

This conditional parameter is used if the data has to be segmented because its size is greater than a single transferable element. The parameter indicates the size of one segment.

**data-size**

This conditional parameter specifies the number of octets of the transferred block in total. It is provided in the first segment of a segmented transfer.

**OD-identifier**

This parameter identifies the entry of the target object dictionary to be read by this service. The used identification type is specified by the "command-ID" parameter.

**payload-data**

This parameter specifies the data to be uploaded.

**Result(+)**

This selection type parameter indicates that the service request succeeded.

**Result(-)**

This selection type parameter indicates that the service request failed.

**6.1.2.3.2.3 Service procedure**

The confirmed service procedure specified in IEC 61158-1, 9.6.1, applies to this service.

**6.1.2.3.3 SDO-write-mult****6.1.2.3.3.1 Service overview**

Using the SDO-write-mult service the client uploads data to the server (owner of the object dictionary), affecting multiple entries of its object dictionary.

**6.1.2.3.3.2 Service primitives**

The service parameters for this service are shown in Table 9.

**Table 9 – SDO-write-mult**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
segment-size	C	C (=)		
data-size	C	C (=)		
OD-identifier (n)	M	M (=)		
payload-data (n)	M	M (=)		
Result(+)			S	S (=)
Result(-)			S	S (=)
error-info (n)			M	M (=)

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.

**Argument**

The argument conveys the service specific parameters of the service request.

**command-ID**

This parameter indicates, how the affected OD entries are to be identified. With the SDO-write-mult service the only allowed type of identification is by index and sub-index. Details are given in IEC 61158-6-13.

**segment-size**

This parameter is used if the data has to be segmented because its size is greater than a single transferable element. The parameter indicates the size of one segment.

**data-size**

This conditional parameter specifies the number of octets of the transferred block in total. It is provided in the first segment of a segmented transfer.

**OD-identifier (n)**

This parameter identifies the n entries of the target object dictionary to be written by this service.

**payload-data (n)**

This parameter specifies the n data sets to be uploaded.

**Result(+)**

This selection type parameter indicates that the service request succeeded.

**Result(-)**

This selection type parameter indicates that the service request for one or more entries failed.

**error-info(n)**

This parameter indicates the entries for which the write failed. The respective error code is given.

**6.1.2.3.3.3 Service procedure**

The confirmed service procedure specified in IEC 61158-1, 9.6.1, applies to this service.

**6.1.2.3.4 SDO-read****6.1.2.3.4.1 Service overview**

Using the SDO-read service the client downloads data from the server (owner of the object dictionary), affecting one entry of its object dictionary.

**6.1.2.3.4.2 Service primitives**

The service parameters for this service are shown in Table 10.

**Table 10 – SDO-read**

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
OD-identifier	M	M (=)		
Result(+)			S	S (=)
segment-size			C	C (=)
data-size			C	C (=)
payload-data			M	M (=)
Result(-)			S	S (=)
error-info			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

**Argument**

The argument conveys the service specific parameters of the service request.

**command-ID**

This parameter indicates how the affected OD entry is to be identified. The identification is done either by index and subindex or by index alone. In the latter case the service addresses all sub-indices (except sub-index 0) of the indicated index. Details are given in IEC 61158-6-13. The length of the payload data must confirm to the length of data for all sub-indices and all sub-indices must be readable.

**OD-identifier**

This parameter identifies the entry of the target object dictionary to be read by this service. The used identification type is specified by the "command-ID" parameter.

**Result(+)**

This selection type parameter indicates that the service request succeeded.

**segment-size**

This parameter is used if the data has to be segmented because its size is greater than a single transferable element. The parameter indicates the size of one segment.

**data-size**

This conditional parameter specifies the number of octets of the transferred block in total. It is provided in the first segment of a segmented transfer.

**payload-data**

This parameter specifies the downloaded data.

**Result(-)**

This selection type parameter indicates that the service request failed.

**6.1.2.3.4.3 Service procedure**

The confirmed service procedure specified in IEC 61158-1, 9.6.1, applies to this service.

**6.1.2.3.5 SDO-read-mult**

**6.1.2.3.5.1 Service overview**

Using the SDO-read-mult service the client downloads data from the server (owner of the object dictionary), affecting multiple entries of its object dictionary.

**6.1.2.3.5.2 Service primitives**

The service parameters for this service are shown in Table 11.

**Table 11 – SDO-read-mult**

Parameter name	Req	Ind	Rsp	Cnf
<b>Argument</b>				
AREP	M	M (=)		
command-ID	M	M (=)		
OD-identifier (n)	M	M (=)		
<b>Result</b>				
segment-size			C	C (=)
data-size			C	C (=)
payload-data / error-info (n)			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.				

**Argument**

The argument conveys the service specific parameters of the service request.

**command-ID**

This parameter indicates how the affected OD entries are to be identified. With the SDO-read-mult service the only allowed type of identification is by index and sub-index. Details are given in IEC 61158-6-13.

**OD entry (n)**

This parameter identifies the n entries of the target object dictionary to be read by this service.

**Result**

This selection type parameter indicates the Result of the service request.

**segment-size**

This parameter is used if the data has to be segmented because its size is greater than a single transferable element. The parameter indicates the size of one segment.

**data-size**

This conditional parameter specifies the number of octets of the transferred block in total. It is provided in the first segment of a segmented transfer.

**payload-data / error-info (n)**

For each of the n object dictionary entries this parameter specifies either the successfully downloaded data or in case of failure the respective error code.

**6.1.2.3.5.3 Service procedure**

The confirmed service procedure specified in IEC 61158-1, 9.6.1, applies to this service.

### 6.1.2.3.6 SDO-abort

#### 6.1.2.3.6.1 Service overview

This service may be executed at any time by either the client or the server of a SDO transfer. If the client of a SDO has a confirmed service outstanding, the indication of the abort is taken to be the confirmation of that service.

#### 6.1.2.3.6.2 Service primitives

The service parameters for this service are shown in Table 12.

**Table 12 – SDO-abort**

Parameter name	Req	Ind
Argument		C
AREP	M	C (=)
error-info	M	C (=)

#### 6.1.2.3.6.3 Service procedure

Although being partially a response termination of a confirmed communication, the service itself is unconfirmed.

### 6.1.3 Process data object ASE

#### 6.1.3.1 Overview

The real-time data transfer is performed by means of Process Data Objects (PDO). PDO communication in a Type 13 network is always performed isochronously via PReq and/or PRes frames (for details see IEC 61158-3-13 and IEC 61158-4-13). The PRes frames are sent as broadcasts or multicasts following the publisher/subscriber scheme. The PReq frames with unicast addresses comply with the master/slave relationship.

The transmission type of PDO is continuous. There is no “on event” or “on change” transmission type provided.

From the device's view, there are two types of PDO usage: data transmission and data reception. Transmit PDOs (TPDOs) and Receive PDOs (RPDOs) are distinguished. Devices supporting TPDOs are PDO publishers and devices which are able to receive PDOs are called PDO subscribers.

The numbers of supported channels is provided by the application. The size of PDO channels is application-specific.

### 6.1.3.2 PDO class specification

#### 6.1.3.2.1 Formal model

<b>ASE:</b>		<b>PDO ASE</b>
<b>CLASS:</b>		<b>PDO</b>
<b>CLASS ID:</b>		<b>not used</b>
<b>PARENT CLASS:</b>		<b>TOP</b>
<b>ATTRIBUTES:</b>		
1	(m)	Attribute: list-of-RPDO-channels
1.1	(m)	Attribute: node-ID
1.2	(m)	Attribute: mapping-version
1.3	(m)	Attribute: list-of-mapped-OD-entries
1.3.1	(m)	Attribute: index
1.3.2	(m)	Attribute: sub-index
1.3.3	(m)	Attribute: offset
1.3.4	(m)	Attribute: length
2	(m)	Attribute: List of TPDO channels
2.1	(m)	Attribute: node-ID
2.2	(m)	Attribute: mapping-version
2.3	(m)	Attribute: list-of-mapped-OD-entries
2.3.1	(m)	Attribute: index
2.3.2	(m)	Attribute: sub-index
2.3.3	(m)	Attribute: offset
2.3.4	(m)	Attribute: length
<b>SERVICES:</b>		
1	(m)	OpsService: PDO-transfer

#### 6.1.3.2.2 Attributes

##### list-of-RPDO-channels

The following attributes specify the logical channels for incoming PDOs. Every type of node can hold up to 254 RPDO channels.

##### node-ID

This attribute specifies the ID of the sending node.

##### mapping-version

This attribute ensures the compatibility of PDO mappings.

##### list-of-mapped-OD-entries

The following attributes specify the OD entries to be mapped to the PDO of the respective channel. A PDO consists of up to 256 mapped OD entries.

##### index

This attribute specifies the index of the OD entry to be mapped.

##### sub-index

This attribute specifies the sub-index of the OD entry to be mapped.

##### offset

This attribute provides the offset related to the start of PDO payload (Bit count).

##### length

This attribute provides the length of the mapped object (Bit count).

### List of TPDO channels

The following attributes specify the logical channels for outgoing PDOs. A MN can hold up to 254 TPDO channels, a CN can have one, an async-only CN has none.

All following attributes correspond to their equivalents under "list-of-RPDO-channels" and are therefore not repeated here.

#### 6.1.3.2.3 Services

##### PDO-transfer

This service is used to transfer process data objects between nodes.

#### 6.1.3.3 PDO ASE service specification

##### 6.1.3.3.1 Supported services

Subclause 6.1.3.3 contains the definition of services that are unique to this ASE. The service defined for this ASE is:

- PDO-transfer

##### 6.1.3.3.2 PDO-transfer

###### 6.1.3.3.2.1 Service overview

This service is used to transfer process data objects between nodes. At the requesting node the outgoing PDO is composed according to the appropriate TPDO attributes; at the receiving node the incoming PDO is handled subject to the corresponding RPDO attributes.

###### 6.1.3.3.2.2 Service primitives

The service parameters for this service are shown in Table 13.

**Table 13 – PDO-transfer**

Parameter name	Req	Ind
Argument		
AREP	M	M (=)
PDO	M	M (=)
PDO-version	M	M (=)

##### Argument

The argument conveys the service specific parameters of the service request.

##### PDO

This parameter specifies the PDO data.

##### PDO-version

This parameter specifies the version of the used PDO mapping.

###### 6.1.3.3.2.3 Service procedure

This service is either performed as one unconfirmed service (from the MN to CNs) or as a sequence of two successive unconfirmed services in opposite directions.

**6.1.4 Application relationship ASE**

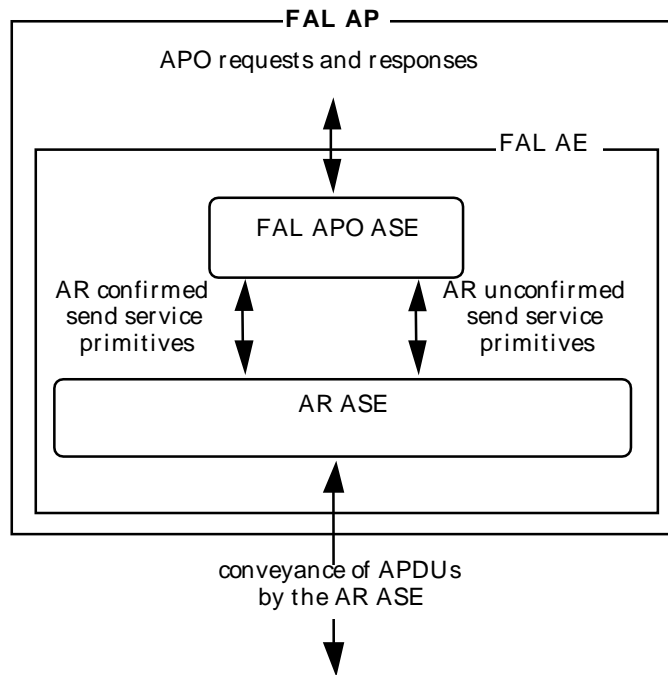
**6.1.4.1 Overview**

**6.1.4.1.1 General**

In a distributed system, application processes communicate with each other by exchanging application layer messages across well-defined application layer communications channels. These communication channels are modeled in the Fieldbus Application Layer (FAL) as application relationships (ARs).

ARs are responsible for conveying messages between applications according to specific communications characteristics required by time-critical systems. Different combinations of these characteristics lead to the definition of different types of ARs. The characteristics of ARs are defined formally as attributes of AR Endpoint classes.

The messages that are conveyed by ARs are FAL service requests and responses. Each is submitted to the AR ASE for transfer by an FAL Application Service Element (ASE) that represents the class of the APO being accessed. Figure 2 illustrates this concept.



**Figure 2 – The AR ASE conveys APDUs between APs**

Depending on the type of AR, APDUs may be sent to one or more destination application processes connected by the AR. Other characteristics of the AR determine how APDUs are to be transferred. These characteristics are described below.

**6.1.4.1.2 Endpoint context**

Each AP involved in an AR contains an endpoint of the AR. Each AR endpoint is defined within the AE of the AP. Its definition, when combined with the definitions of the other endpoints defines an AR. To ensure communications compatibility among or between endpoints, each endpoint definition contains a set of compatibility-related characteristics. These characteristics need to be configured appropriately for each endpoint for the AR to operate properly.



Endpoint definitions also contain a set of characteristics that describe the operation of the AR. These characteristics, when combined with those used to specify compatibility, define the context of the endpoint. The endpoint context is used by the AR ASE to manage the operation of the endpoint and the conveyance of APDUs. The characteristics which comprise the endpoint context are described next.

#### 6.1.4.1.2.1 Endpoint role

The role of an AREP determines the permissible behavior of an AP at the AREP. An AREP role may be that of a client, server, push publisher, push subscriber, master, or slave.

Table 14 and Table 15 summarize the characteristics and combinations of each of the AREP roles.

**Table 14 – Conveyance of service primitives by AREP role**

	Client	Server	Push Publisher	Push Subscriber	Master	Slave
Send Service Req	X		X		X	
Recv Service Req		X		X		X
Send Service Rsp		X				
Recv Service Rsp	X					

**Table 15 – Valid combinations of AREP roles involved in an AR**

AREP Roles by Interaction Model	Client	Server	Push Publisher	Push Subscriber	Master	Slave
Client/server						
Client		Yes				
Server						
Publisher/subscriber						
Push Publisher				Yes		
Push Subscriber						
Master						Yes
Slave						

#### 6.1.4.1.2.2 Cardinality

The cardinality of an AR specifies, from the point of view of a client or publisher endpoint, how many remote application processes are involved in an AR. Cardinality is never expressed from the viewpoint of a server or a subscriber.

When expressed from the viewpoint of a client or peer endpoint, ARs are always 1-to-1. Clients are never capable of issuing a request and waiting for responses from multiple servers.

When expressed from the viewpoint of a publisher endpoint, they indicate that multiple subscribers are supported. These ARs are 1-to-many. ARs that are 1-to-many provide for communications between one application and a group of (one or more) applications. They are often referred to as multicast.

### **6.1.4.1.3 Conveyance model**

#### **6.1.4.1.3.1 General**

The conveyance model defines how APDUs are sent between endpoints of an AR. Three characteristics are used to define these transfers:

- conveyance paths,
- trigger policy, and
- conveyance policy.

#### **6.1.4.1.3.2 Conveyance paths**

The purpose of AR ASEs is to transfer information between AR endpoints. This information transfer occurs over the conveyance paths of an AR. A conveyance path represents a one-way communication path used by an endpoint for input or output.

To support the role of the application process, the endpoint is configured with either one or two conveyance paths. Endpoints that only send, or only receive, are configured with either a send or receive conveyance path, and those that do both are configured with both. ARs with a single conveyance path are called uni-directional, and those with two conveyance paths are called bi-directional.

Uni-directional ARs are capable of conveying service requests only. To convey service responses, a bi-directional AR is necessary. Therefore, uni-directional ARs support the transfer of unconfirmed services in one direction only, while bi-directional ARs support the transfer of unconfirmed and confirmed services initiated by only one endpoint, or by both endpoints.

#### **6.1.4.1.3.3 Trigger policy**

Trigger policy indicates when APDUs are transmitted by the data-link layer over the network.

The first type is referred to as user-triggered. User-triggered AREPs submit FAL APDUs to the data-link layer for transmission at its earliest opportunity.

The second type is referred to as network-scheduled. Network scheduled AREPs submit FAL APDUs to the data-link layer for transmission according to a schedule configured by management.

This network scheduling mechanism is cyclic.

#### **6.1.4.1.3.4 Conveyance policy**

Conveyance policy indicates whether APDUs are transferred according to a buffer model or a queue model. These models describe the method of conveying APDUs from sender to receiver.

Buffered ARs contain conveyance paths that have a single buffer at each endpoint. Updates to the source buffer are conveyed to the destination buffer according to the trigger policy of the AR. Updates to either buffer replace its contents with new data. In buffered ARs, not conveyed or undelivered data that has been replaced is lost. Additionally, data contained in a buffer may be read multiple times without destroying its contents.

Queued ARs contain conveyance paths that are represented as a queue between endpoints. Queued ARs convey data using a FIFO queue. Queued ARs are not overwritten; new entries are queued until they can be conveyed and delivered.

If a queue is full, a new message will not be placed into the queue.

NOTE The AR conveyance services are described abstractly in such a way that they are capable of being implemented to operate using buffers or queues. These services can be implemented in a number of ways. For example, they can be implemented such that the capability is provided to load the buffer/queue, and subsequently post it for transfer by the underlying data-link layer. Or, these services can be implemented such that these capabilities are combined so that the buffer/queue can be loaded and transferred in a single request. On the receiving side, these services may be implemented by delivering the data when it is received, or by indicating its receipt and allowing the user to retrieve it in a separate operation. Another option is to require the user to detect that the buffer or queue has been updated.

#### **6.1.4.1.4 Underlying communications services**

##### **6.1.4.1.4.1 General**

The AR ASE conveys FAL APDUs using the capabilities of the underlying data-link layer. Several characteristics are used to describe these capabilities. Subclause 6.1.4.1.4 provides a description of each. These characteristics are specific to the data-link mapping and are defined in IEC 61158-6-13. Their precise specification can be found there.

##### **6.1.4.1.4.2 Connection-oriented services**

The underlying layer does not support AR endpoints by providing connection-oriented services. Thus connections have to be established by means of the Application Layer.

##### **6.1.4.1.4.3 Buffered and queued services**

The underlying layer may support AR endpoints by providing buffered or queued services. These services can be used to implement buffers or queues required by certain classes of endpoints.

##### **6.1.4.1.4.4 Cyclic and acyclic transfers**

The underlying layer may support AR endpoints by providing cyclic or acyclic services.

##### **6.1.4.1.4.5 Segmentation**

The FAL has the capability of segmenting user data conveyed by it.

##### **6.1.4.1.5 AR establishment**

For an AR endpoint to be used by an application process, the corresponding AR has to be active. When an AR is activated, it is referred to as "established".

AR establishment can occur in one of two ways. First, ARs can be pre-established. Pre-established means that the AE that maintains the endpoint context is created when the AP is connected to the network. In this case, communications among the applications involved in the AR may take place without first having to explicitly establish the AR.

Second, ARs can require dynamic definition and establishment. In this case, definitions have to be created for each of the AREPs by means of the QUB-COS (CmdL) ARPM. Details are given in IEC 61158-6-13.

##### **6.1.4.1.6 Application relationship classes**

AREPs are defined with a combination of characteristics to form different classes of ARs.

#### **6.1.4.2 Application relationship endpoint class specifications**

##### **6.1.4.2.1 Formal model**

The AR endpoint formal model defines the characteristics common to all AR endpoints. This class is not capable of being instantiated. It is present only for the inheritance of its attributes and services by its subclasses, each specified in a separate subclause of this standard.

<b>FAL ASE:</b>		<b>AR ASE</b>
<b>CLASS:</b>		<b>AR ENDPOINT</b>
<b>CLASS ID:</b>		<b>not used</b>
<b>PARENT CLASS:</b>		<b>TOP</b>
<b>SYSTEM MANAGEMENT ATTRIBUTES:</b>		
1	(m)	Attribute: Local AP
2	(o)	Attribute: List of supported attributes
<b>SERVICES:</b>		
1	(o)	OpsService: AR-Ident
2	(o)	OpsService: AR-Status
3	(o)	OpsService: AR-NMT-req-invite
4	(o)	OpsService: AR-NMT-state-command
5	(o)	OpsService: AR-NMT-info
6	(o)	OpsService: AR-SDO-write
7	(o)	OpsService: AR-SDO-write-mult
8	(o)	OpsService: AR-SDO-read
9	(o)	OpsService: AR-SDO-read-mult
10	(o)	OpsService: AR-SDO-abort
11	(m)	OpsService: AR-PDO-transfer

**6.1.4.2.2 System management attributes**

**Local AP**

This attribute identifies the AP attached or configured to use the AREP using a local reference.

**List of supported attributes**

This optional attribute specifies the attributes supported by the object. This list contains, at a minimum, the mandatory attributes for the class of the object.

**6.1.4.2.3 Services**

The functionalities of all services listed above correspond to the services of the same name (without prefix "AR") listed in 6.1.1 through 6.1.3. The services are repeated her for formal reasons. In the further context of IEC 61158-5-13 and IEC 61158-6-13 they are referenced without the added prefix.

**6.1.4.3 Application relationship ASE service specifications**

**6.1.4.3.1 Supported services**

See 6.1.1 through 6.1.3.

**6.2 ARs**

**6.2.1 Buffered network-scheduled bi-directional pre-established connection (BNB-PEC) AR endpoint class specification**

**6.2.1.1 Class overview**

This class is defined to support the cyclic exchange of unconfirmed services between application processes using buffers. The cycle is given by the MN.

The services provided by this AREP permit a direct communication between CNs, the so called cross traffic

The behavior of this type of AR can be described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP for distribution. This activity is always located on the MN. It writes the APDU into the internal buffer, completely replacing the existing contents of the buffer. The data-link layer transfers the buffer contents at the next scheduled transfer opportunity.

If the AREP receives another APDU before the buffer contents are transmitted, the buffer contents will be replaced with the new APDU, and the previous APDU will be lost. When the buffer contents are transmitted, the AR ASE notifies the user of transmission.

The receiving endpoint is always located on a CN. Here the APDU is received from the network and is written immediately into the buffer, completely overwriting the existing contents of the buffer. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface.

The following response from the remote AR ASE user is no confirmation or acknowledgement of the received ADPU but contains data independent from it. It is written by the AREP into the data-link layer buffer, completely replacing the existing contents of the buffer. The data-link layer transfers the buffer contents as soon as possible. The delay time between incoming data indication and the following response only depends on the capabilities of the node hosting the AREP.

At the requesting endpoint (on the MN again), the APDU containing the response received from the network is written to the buffer, completely replacing the existing contents of the buffer. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface as the confirmation.

In addition to that any other CN may receive and interpret this response, depending on its configuration.

The following summarizes the characteristics of this AREP class.

Role	Push publisher / Push subscriber
Cardinality	1-to-1 (for the request) 1-to-n (for the response)

#### 6.2.1.2 Formal model

<b>ASE:</b>	<b>AR</b>
<b>CLASS:</b>	<b>Buffered network-scheduled bi-directional pre-established connection AR endpoint</b>
<b>CLASS ID:</b>	<b>not used</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>ATTRIBUTES:</b>	
1	(m) Attribute: Role
2	(m) Attribute: AREP-state
3	(m) Attribute: DL-mapping-reference
<b>SERVICES:</b>	
1	(m) OpsService: PDO-transfer

#### 6.2.1.3 Network management attributes

##### Role

This attribute specifies the role of the AREP. The valid values are:

- Push-publisher: Endpoints of this type publish their data issuing unconfirmed service request-APDUs.

- Push-subscriber: Endpoints of this type receive data from service request-APDUs released by a push-publisher AREP.

**AREP-state**

This attribute specifies the state of the AREP. The values for this attribute are specified in IEC 61158-6-13.

**DL-mapping-reference**

For publisher AREPs, this attribute specifies the mapping to the transmit conveyance path. For subscriber AREPs, this attribute specifies the mapping to the receive conveyance path. DL mapping attributes for the data-link layer are specified in IEC 61158-6-13.

**6.2.1.4 Services**

**PDO-transfer**

This service is used to transfer process data objects between nodes.

**6.2.2 Buffered network-scheduled uni-directional pre-established connection (BNU-PEC) AR endpoint class specification**

**6.2.2.1 Class overview**

This class is defined to support the “push” model for scheduled, buffered distribution of unconfirmed services to one or more application processes.

The behavior of this type of AR can be described as follows.

An AR ASE user wishing to convey a request or response APDU submits it as an AR ASE Service Data Unit to its AREP for distribution. This activity is always located on the MN. The AREP sending the request or response APDU writes it into the data-link layer buffer, completely replacing the existing contents of the buffer. The data-link layer transfers the buffer contents at the next scheduled transfer opportunity.

If the AREP that sent the APDU receives another APDU before the buffer contents are transmitted, the buffer contents will be replaced with the new APDU, and the previous APDU will be lost. When the buffer contents are transmitted, the AR ASE notifies the user of the transmission.

One or more CN(s) are the receiving endpoint(s). Here the APDU is received from the network and is immediately written into the buffer, completely overwriting the existing contents of the buffer. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface. If the APDU has not been delivered before the next APDU arrives, it will be overwritten by the next APDU and lost.

The following summarizes the characteristics of this AREP class.

Roles	Push publisher Push subscriber
Cardinality	1-to-n

### 6.2.2.2 Formal model

<b>ASE:</b>	<b>AR</b>
<b>CLASS:</b>	<b>Buffered network-scheduled uni-directional pre-established connection AR endpoint</b>
<b>CLASS ID:</b>	<b>not used</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>ATTRIBUTES:</b>	
1	(m) Attribute: Role
2	(m) Attribute: AREP-state
3	(m) Attribute: DL-mapping-reference
<b>SERVICES:</b>	
1	(m) OpsService: PDO transfer

### 6.2.2.3 Network management attributes

#### Role

This attribute specifies the role of the AREP. The valid values are:

- Push-publisher: Endpoints of this type publish their data issuing unconfirmed service request-APDUs.
- Push-subscriber: Endpoints of this type receive data from service request-APDUs released by a push-publisher AREP.

#### AREP-state

This attribute specifies the state of the AREP. The values for this attribute are specified in IEC 61158-6-13.

#### DL-mapping-reference

For the Publisher AREP on the MN, this attribute specifies the mapping to the transmit conveyance path. For the Subscriber AREPs on the CNs, this attribute specifies the mapping to the receive conveyance path. DL mapping attributes for the data-link layer are specified in IEC 61158-6-13.

### 6.2.2.4 Services

#### PDO-transfer

This service is used to transfer process data objects between nodes.

## 6.2.3 Queued user-triggered uni-directional (QUU) AR endpoint class specification

### 6.2.3.1 Class overview

This class is defined to support the on-demand queued distribution of unconfirmed services to one or more application processes. The behavior of this type of AR can be described as follows.

An AR ASE user wishing to convey a request APDU submits an AR ASE Service Data Unit to the sending endpoint of the AR. The AREP sending the request APDU submits it to its underlying layer for transfer. The underlying layer sends it at its next opportunity. The AREP receiving the request APDU from its underlying layer delivers it to the AR ASE user in the order that it was received.

The following summarizes the characteristics of this AREP class.

Roles	Master
	Slave
Cardinality	1-to-n

### 6.2.3.2 Formal model

<b>FAL ASE:</b>	<b>AR ASE</b>
<b>CLASS:</b>	<b>Queued user-triggered uni-directional AR endpoint</b>
<b>CLASS ID:</b>	<b>not used</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>NETWORK MANAGEMENT ATTRIBUTES:</b>	
1	(m) Attribute: Role
2	(m) Attribute: AREP State
3	(m) Attribute: DL-mapping-reference
<b>SERVICES:</b>	
1	(o) OpsService: NMT-state-command
2	(o) OpsService: NMT-info

### 6.2.3.3 Network management attributes

#### Role

This attribute specifies the role of the AREP. The valid values are

- Master: Endpoints of this type release their data issuing unconfirmed service request-APDUs.
- Slave: Endpoints of this type receive their data from service request-APDUs released by a master AREP.

#### AREP-state

This attribute specifies the state of the AREP. The values for this attribute are specified in IEC 61158-6-13.

#### DL-mapping-reference

This attribute provides a reference to the underlying Data Link Layer mapping for the conveyance path for this AREP. DL mapping attributes for the data-link layer are specified in IEC 61158-6-13.

### 6.2.3.4 Services

#### NMT-state-command

The MN uses NMT-state-command services to control the state machine(s) of the CN(s).

#### NMT-info

This service is used to transmit complex information from the MN to the CNs.

## 6.2.4 Queued user-triggered bi-directional connectionless (QUB-CL) AR endpoint class specification

### 6.2.4.1 Class overview

This class is defined to support the on-demand exchange of unconfirmed services between two or more application processes. This class uses connectionless data link services for the exchanges. The behavior of this class is described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP. The AREP sending the request APDU queues it to its underlying layer for transfer at the next available opportunity.

The AREP receiving the request APDU from its underlying layer, queues it for delivery to its AR ASE user in the order that it was received.

For a confirmed service request the AREP receiving the request APDU accepts the corresponding response APDU from its AR ASE user and queues it to the underlying layer for transfer.



The AREP that issued the request APDU receives the response APDU from its underlying layer and queues it for delivery to its AR ASE user in the order that it was received.

The following summarizes the characteristics of this AREP class.

Roles	Client Server
Cardinality	1-to-1

#### 6.2.4.2 Formal model

<b>FAL ASE:</b>	<b>AR ASE</b>
<b>CLASS:</b>	<b>Queued user-triggered bi-directional connectionless AR Endpoint</b>
<b>CLASS ID:</b>	<b>not used</b>
<b>PARENT CLASS:</b>	<b>AR Endpoint</b>
<b>NETWORK MANAGEMENT ATTRIBUTES:</b>	
1	(m) Attribute: Role
2	(m) Attribute: AREP State
3	(m) Attribute: DL-mapping-reference
<b>SERVICES:</b>	
1	(o) OpsService: Ident
2	(o) OpsService: Status
3	(o) OpsService: Sync
4	(o) OpsService: NMT-req-invite

#### 6.2.4.3 Network management attributes

##### Role

This attribute specifies the role of the AREP. The valid values are:

- Client: Endpoints of this type issue unconfirmed service request-APDUs to servers and receive unconfirmed service response-APDUs.
- Server: Endpoints of this type receive unconfirmed service request-APDUs from clients and issue unconfirmed service response-APDUs to them.

##### AREP-state

This attribute specifies the state of the AREP. The values for this attribute are specified in IEC 61158-6-13.

##### DL-mapping-reference

This attribute provides a reference to the underlying Data Link Layer mapping for the conveyance path for this AREP. DL mapping attributes for the data-link layer are specified in IEC 61158-6-13.

#### 6.2.4.4 Services

##### Ident

The Ident service is used by the MN to identify CNs.

##### Status

The Status service is used by the MN to query status information from the CN.

##### Sync

The Sync service is used by the MN to configure/synchronize a CN with a time-triggered PRes.

##### NMT-req-invite

The MN uses this service to find out what service the CN wants to be performed next by the MN.

**6.2.5 Queued user-triggered bi-directional connection-oriented with segmentation (QUB-COS) AR endpoint class specification**

**6.2.5.1 Class overview**

This class is defined to support the on-demand exchange of confirmed services between two application processes. Unconfirmed services are not supported by this type of AR. For the handling of connections it uses the connection-related services provided by the AR ASE. It supports segmentation of APDUs within the Application Layer. The data link priority for the transfers is specified separately for each transfer.

The behavior of this class is described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP. The AREP sending the request APDU queues it to its underlying layer for transfer at the next available opportunity.

The AREP receiving the request APDU from its underlying layer, queues it for delivery to its AR ASE user in the order that it was received.

For a confirmed service request the AREP receiving the request APDU accepts the corresponding response APDU from its AR ASE user and queues it to the underlying layer for transfer.

The AREP that issued the request APDU receives the response APDU from its underlying layer and queues it for delivery to its AR ASE user in the order that it was received.

For both directions an AR ASE service data unit can be split into several APDUs if its size is greater than a single transferable element. The parameters for handling the segmentation are provided by the FAL ASE wishing to send the affected data. The underlying layer sends each segment at the next opportunity. The AREP receiving the APDU segments from its underlying layer reassembles the APDU and delivers it to the AR ASE.

The following summarizes the characteristics of this AREP class.

Roles	Client
	Server
Cardinality	1-to-1

**6.2.5.2 Formal model**

<b>ASE:</b>	<b>AR</b>
<b>CLASS:</b>	<b>Queued user-triggered bi-directional connection-oriented with segmentation AR endpoint</b>
<b>CLASS ID:</b>	<b>not used</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>ATTRIBUTES:</b>	
1 (m) Attribute:	Role
2 (m) Attribute:	AREP-state
3 (m) Attribute:	DL-mapping-reference
<b>SERVICES:</b>	
1 (m) OpsService:	SDO-write
2 (o) OpsService:	SDO-write-mult
3 (m) OpsService:	SDO-read
4 (o) OpsService:	SDO-read-mult
5 (m) OpsService:	SDO-abort

### 6.2.5.3 Network management attributes

#### Role

This attribute specifies the role of the AREP. The valid values are

- Client: Endpoints of this type issue confirmed service Request-APDUs to servers and receive confirmed service Response-APDUs.
- Server: Endpoints of this type receive confirmed service Request-APDUs from clients and issue confirmed service Response-APDUs to them.

#### AREP-state

This attribute specifies the state of the AREP. The values for this attribute are specified in IEC 61158-6-13.

#### DL-mapping-reference

This attribute provides a reference to the underlying Data Link Layer mapping for the conveyance path for this AREP. DL mapping attributes for the data-link layer are specified in IEC 61158-6-13.

### 6.2.5.4 Services

#### SDO-write

This service is used to write data to one entry of the server's object dictionary.

#### SDO-write-mult

This service is used to write data to multiple entries of the server's object dictionary.

#### SDO-read

This service is used to read data from one entry of the server's object dictionary.

#### SDO-read-mult

This service is used to read data from multiple entries of the server's object dictionary.

#### SDO-abort

The service is used to abort a SDO communication.

## 6.3 Summary of FAL classes

Subclause 6.3 contains a summary of the defined FAL Classes. The Class ID values have been assigned to be compatible with existing standards. Table 16 provides a summary of the classes.

**Table 16 – FAL class summary**

FAL ASE	Class
	AP
Data type	Fixed length & String data type
Application Relationship	AREP
	BNU-PEC
	BNB-PEC
	QUU
	QUB-CL
	QUB-COS
Service data object	SDO
Process data object	PDO

**6.4 Permitted FAL services by AREP role**

Table 17 below defines the valid combinations of services and AREP roles (which service APDUs and AREP with the specified role can send or receive). The Unc and Cnf columns indicate whether the service listed in the left-hand column is unconfirmed (Unc) or confirmed (Cnf).

**Table 17 – Services by AREP role**

	Unconfirmed	Confirmed	Client	Server	Push Publisher	Push Subscriber	Master	Slave
FAL Services			req-rcv	req-rcv	req-rcv	req-rcv	req-rcv	req-rcv
AP ASE								
Ident	x		x x	x x				
Status	x		x x	x x				
Sync	x						x x	x x
NMT-req-invite	x		x x	x x				
NMT-state-command	x						x	x
NMT-info	x						x	x
SDO ASE								
SDO-write		x	x x	x x				
SDO-write-mult		x	x x	x x				
SDO-read		x	x x	x x				
SDO-read-mult		x	x x	x x				
PDO ASE								
PDO-transfer	x				x	x		
AR ASE								
(See equivalent services above)								

## Bibliography

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

EPSS DS 301 V1.2.0, *Ethernet POWERLINK Communication Profile Specification, Draft Standard Version 1.2.0, EPSS 2013, available at <http://www.ethernet-powerlink.org/>*

---

## SOMMAIRE

AVANT-PROPOS.....	62
INTRODUCTION.....	64
1 Domaine d'application .....	65
1.1 Généralités.....	65
1.2 Spécifications.....	66
1.3 Conformité .....	66
2 Références normatives.....	66
3 Termes, définitions, symboles, abréviations et conventions .....	67
3.1 Termes de l'ISO/CEI 7498-1 .....	67
3.2 Termes de l'ISO/CEI 8822 .....	67
3.3 Termes de l'ISO/CEI 9545 .....	67
3.4 Définitions spécifiques à la couche application de bus de terrain.....	68
3.5 Abréviations et symboles.....	74
3.6 Conventions .....	76
4 Concepts.....	78
4.1 Concepts typiques.....	78
4.2 Concepts spécifiques à un type .....	79
5 Data type ASE.....	80
5.1 Généralités.....	80
5.2 Définition formelle des objets de data type .....	80
5.3 Types de données de FAL définis.....	80
5.4 Spécification du service de l'ASE Data type .....	86
6 Spécification du modèle de communication de Type 13 .....	86
6.1 ASE.....	86
6.2 AR.....	109
6.3 Résumé sur les classes de FAL.....	116
6.4 Services de FAL autorisés par le rôle d'AREP .....	117
Bibliographie.....	119
Figure 1 – ASE de Type 13.....	80
Figure 2 – ASE d'AR achemine des APDU entre les AP .....	104
Tableau 1 – Structure globale de l'OD.....	79
Tableau 2 – Service Ident .....	89
Tableau 3 – Service Status .....	91
Tableau 4 – Service Sync .....	92
Tableau 5 – Service NMT-req-invite .....	93
Tableau 6 – Service NMT-state-command.....	94
Tableau 7 – Service NMT-info.....	94
Tableau 8 – SDO-write .....	97
Tableau 9 – SDO-write-mult.....	98
Tableau 10 – SDO-read .....	99
Tableau 11 – SDO-read-mult .....	100
Tableau 12 – SDO-abort .....	101

Tableau 13 – PDO-transfer .....	103
Tableau 14 – Acheminement des primitives de service par le rôle d'AREP .....	105
Tableau 15 – Combinaisons valides des rôles d'AREP impliqués dans une AR .....	105
Tableau 16 – Résumé sur les classes de FAL .....	117
Tableau 17 – Services par le rôle d'AREP .....	118

## COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

### RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

#### Partie 5-13: Définition des services de la couche application – Éléments de type 13

#### AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) 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, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). 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 de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

L'attention est attirée sur le fait que l'utilisation du type de protocole associé est restreinte par les détenteurs des droits de propriété intellectuelle. En tout état de cause, l'engagement de renonciation partielle aux droits de propriété intellectuelle pris par les détenteurs de ces droits autorise l'utilisation d'un type de protocole de couche avec les autres protocoles de couche du même type, ou dans des combinaisons avec d'autres types autorisées explicitement par les détenteurs des droits de propriété intellectuelle pour ce type.

NOTE Les combinaisons de types de protocoles sont spécifiées dans la CEI 61784-1 et la CEI 61784-2.

La Norme internationale CEI 61158-5-13 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.



Cette deuxième édition annule et remplace la première édition parue en 2007. Cette édition constitue une révision technique.

Les modifications majeures par rapport à l'édition précédente sont énumérées ci-dessous:

- ajout d'une fonction de synchronisation,
- corrections, et
- améliorations rédactionnelles.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65C/763/FDIS	65C/773/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.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61158, publiées sous le titre général *Réseaux de communication industriels – Spécifications des bus de terrain*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous <http://webstore.iec.ch> dans les données relatives à la publication recherchée. À cette date, la publication sera:

- reconduite;
- supprimée;
- remplacée par une édition révisée, ou
- amendée

## INTRODUCTION

La présente partie de la norme CEI 61158 s'inscrit dans une série créée pour faciliter l'interconnexion des composants de systèmes d'automatisation. Elle renvoie aux autres normes de l'ensemble défini par le modèle de référence de bus de terrain "à trois couches" décrit dans la CEI 61158-1.

Le Service d'Application est fourni par le Protocole d'Application en utilisant les services disponibles issus de la couche liaison de données d'une autre couche immédiatement inférieure. La présente norme définit les caractéristiques du service d'application qui peuvent être utilisées par les applications de bus de terrain et/ou la gestion de systèmes.

Dans toute la série de normes relatives aux bus de terrain, le terme «service» renvoie à la capacité abstraite fournie par une couche du Modèle de Référence de Base pour l'Interconnexion des Systèmes Ouverts (OSI) à la couche immédiatement supérieure. Ainsi, le service de la couche application défini dans la présente norme est un service architectural conceptuel, indépendant des divisions administratives et de mise en œuvre.

## RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

### Partie 5-13: Définition des services de la couche application – Éléments de type 13

#### 1 Domaine d'application

##### 1.1 Généralités

La couche application de bus de terrain (FAL «Fieldbus Application Layer») fournit aux programmes d'utilisateur un moyen d'accéder à l'environnement de communication du bus de terrain. À cet égard, la FAL peut être vue comme une "fenêtre entre des programmes d'application correspondants".

La présente norme fournit des éléments communs pour les communications par messagerie de base prioritaire et non prioritaire entre des programmes d'application dans un environnement d'automatisation et des matériaux spécifiques aux bus de terrain de Type 13. Le terme «prioritaire» est utilisé pour montrer la présence d'une fenêtre temporelle, dans les limites de laquelle une ou plusieurs actions spécifiées sont tenues d'être parachevées avec un niveau défini de certitude. Le manquement à parachever les actions spécifiées dans les limites de la fenêtre temporelle risque d'entraîner la défaillance des applications qui demandent les actions, avec le risque concomitant pour l'équipement, l'installation et éventuellement pour la vie humaine.

La présente norme définit de manière abstraite le comportement visible de l'extérieur fourni par les différents Types de la couche application de bus de terrain en termes

- a) d'un modèle abstrait pour la définition des ressources (objets) d'application capables d'être employées par les utilisateurs à l'aide du service de FAL.
- b) des actions et événements primitifs du service;
- c) des paramètres associés à chaque action et événement primitif et la forme qu'ils prennent; et
- d) de l'interrelation entre ces actions et événements, et leurs séquences valides.

La présente norme a pour objectif de définir les services fournis à

- 1) l'utilisateur de FAL à la frontière entre l'utilisateur et la couche application du modèle de référence de bus de terrain, et
- 2) la gestion des systèmes à la frontière entre la couche application et la gestion des systèmes selon le modèle de référence de bus de terrain.

La présente norme spécifie la structure et les services de la couche application de bus de terrain de la CEI en conformité avec le Modèle de référence de base pour l'interconnexion des systèmes ouverts (ISO/CEI 7498-1) et la structure de la couche application OSI (ISO/CEI 9545).

Des services et protocoles de FAL sont fournis par les entités d'application (AE) de FAL contenues au sein des processus d'application. Une AE de FAL est composée d'un ensemble d'éléments de services d'application (ASE) orientés objet et d'une entité de gestion de couche (LME) qui gère l'AE. Les ASE fournissent des services de communication qui fonctionnent sur un ensemble des classes connexes d'objets des processus d'application (APO). L'un des ASE de FAL est un ASE de gestion qui fournit un ensemble typique de services pour la gestion des instances des classes de FAL.

Bien que ces services spécifient, du point de vue des applications, comment les demandes et les réponses sont émises et livrées, ils ne comprennent pas la spécification de ce que les applications de demande et de réponse sont tenues d'en faire. Cela veut dire que les aspects comportementaux des applications ne sont pas spécifiés; seule la définition des demandes et des réponses, qu'ils peuvent envoyer/recevoir, est spécifiée. Ainsi, les utilisateurs de FAL sont dotés d'une plus grande flexibilité pour la normalisation d'un tel comportement d'objet. Outre ces services, des services de support, également définis dans la présente norme, donnent accès à la FAL pour le contrôle de certains aspects de son fonctionnement.

## 1.2 Spécifications

L'objectif principal de la présente norme est de spécifier les caractéristiques des services conceptuels de la couche application qui sont adaptés à la communication prioritaire, et de compléter ainsi le Modèle de référence de base OSI en guidant le développement des protocoles de couche application pour les communications prioritaires.

Un autre objet consiste à assurer des trajets de migration à partir de protocoles de communications industrielles préexistants. C'est ce dernier objectif qui donne lieu à la diversité des services normalisés comme les divers Types de la CEI 61158, et les protocoles correspondants normalisés dans les sous-parties de la CEI 61158-6.

La présente spécification peut servir de base pour les interfaces formelles de programmation d'application. Toutefois, ce n'est pas une interface de programmation formelle, et il sera nécessaire pour toute interface de ce type de traiter des questions de mise en œuvre qui ne sont pas couvertes par cette spécification, y compris

- a) les dimensions et l'ordonnancement des octets pour les divers paramètres de service multi-octets, et
- b) la corrélation de primitives appariées "request-confirm" ou "indication-response".

## 1.3 Conformité

La présente norme ne spécifie pas de mises en œuvre individuelles ou de produits individuels ni ne contraint les mises en œuvre d'entités de couche application au sein des systèmes d'automatisation industriels.

Il n'y a pas de conformité d'équipement à la présente norme de définition des services de couche application. Au contraire, la conformité est obtenue par la mise en œuvre des protocoles de couche application correspondants qui satisfont aux services de couche application de tous les Types donnés tels que définis dans la présente norme.

## 2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

NOTE Toutes les parties de la série CEI 61158, ainsi que la CEI 61784-1 et la CEI 61784-2 font l'objet d'une maintenance simultanée. Les références croisées à ces documents dans le texte se rapportent par conséquent aux éditions datées dans la présente liste de références normatives.

CEI 61158-1:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 1: Présentation et lignes directrices des séries CEI 61158 et CEI 61784*

CEI 61158-3-13, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 3-13: Définition des services de la couche liaison de données – Éléments de type 13*

CEI 61158-4-13, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 4-13: Spécification du protocole de la couche liaison de données – Eléments de type 13*

CEI 61158-6 (toutes les parties), *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6: Spécification du protocole de la couche application*

CEI 61158-6-13, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-13: Spécification du protocole de la couche application – Eléments de type 13*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange* (disponible en anglais seulement)

ISO/CEI 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

ISO/CEI 8822, *Technologies de l'information – Interconnexion de systèmes ouverts – Définition du service de présentation*

ISO/CEI 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche Application*

ISO/CEI 10731, *Technologies de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Conventions pour la définition des services OSI*

IEEE 754, *IEEE Standard for Floating-Point Arithmetic* (disponible en anglais seulement)

### **3 Termes, définitions, symboles, abréviations et conventions**

Pour les besoins du présent document, les termes, définitions, symboles, abréviations et conventions suivants s'appliquent.

#### **3.1 Termes de l'ISO/CEI 7498-1**

##### **3.1.1 entité d'application**

##### **3.1.2 processus d'application**

##### **3.1.3 unité de données de protocole d'application**

##### **3.1.4 élément de service d'application**

##### **3.1.5 invocation d'entités d'application**

##### **3.1.6 invocation de processus d'application**

##### **3.1.7 transaction d'application**

##### **3.1.8 système ouvert réel**

##### **3.1.9 syntaxe de transfert**

#### **3.2 Termes de l'ISO/CEI 8822**

##### **3.2.1 syntaxe abstraite**

##### **3.2.2 contexte de présentation**

#### **3.3 Termes de l'ISO/CEI 9545**

##### **3.3.1 association d'application**

##### **3.3.2 contexte d'application**

- 3.3.3 nom de contexte d'application**
- 3.3.4 invocation d'entités d'application**
- 3.3.5 type d'entité d'application**
- 3.3.6 invocation de processus d'application**
- 3.3.7 type de processus d'application**
- 3.3.8 élément de service d'application**
- 3.3.9 élément de service de contrôle d'application**

#### **3.4 Définitions spécifiques à la couche application de bus de terrain**

Pour les besoins de la présente partie de la CEI 61158, les termes et définitions suivants s'appliquent.

##### **3.4.1 application**

fonction ou structure de données pour laquelle des données sont consommées ou produites

##### **3.4.2 objets d'application**

classes d'objets multiples qui gèrent et assurent un échange de messages à travers le réseau et au sein de l'appareil interconnecté

##### **3.4.3 processus d'application**

partie d'une application distribuée sur un réseau, située sur un appareil et associée à une adresse non ambiguë

##### **3.4.4 identificateur de processus d'application**

distingue plusieurs processus d'application utilisés dans un appareil

##### **3.4.5 objet de processus d'application**

composant d'un processus d'application qui peut être identifié et accessible à travers une relation entre applications de FAL

Note 1 à l'article: Les définitions d'objets des processus d'application sont composées d'un ensemble des valeurs pour les attributs de leur classe (voir la définition de la classe d'objets des processus d'application). Les définitions d'objets des processus d'application peuvent être accessibles à distance en utilisant les services d'ASE de gestion d'objets de FAL. Les services de gestion d'objets de FAL peuvent être utilisés pour charger et mettre à jour les définitions d'objets, lire les définitions d'objets, créer de façon dynamique et supprimer les objets d'application et leurs définitions correspondantes.

##### **3.4.6 classe d'objets de processus d'application**

classe d'objets de processus d'application définis par un ensemble de services et d'attributs accessibles par le réseau

##### **3.4.7 relation entre applications**

association coopérative entre deux ou plusieurs invocations d'entités d'application dans le but d'échanger des informations et de coordonner leur opération conjointe

Note 1 à l'article: Cette relation est activée soit par l'échange des unités de données du protocole d'application, soit en tant que résultat des activités de préconfiguration.

### 3.4.8

#### **élément de service d'application des relations entre applications**

élément de service d'application qui fournit un moyen exclusif pour établir et terminer toutes les relations entre applications

### 3.4.9

#### **point d'extrémité de relations entre applications**

contexte et comportement d'une relation entre applications tels que vus et maintenus par l'un des processus entre applications impliqués dans la relation entre applications

Note 1 à l'article: Chaque processus d'application impliqué dans la relation d'application maintient son propre point d'extrémité de relations d'application.

### 3.4.10

#### **attribut**

description d'une caractéristique ou d'une particularité d'un objet visible de l'extérieur

Note 1 à l'article: Les attributs d'un objet contiennent des informations sur les portions variables d'un objet. Ils fournissent généralement des informations sur le statut ou dirigent l'opération d'un objet. Les attributs peuvent également affecter le comportement d'un objet. Ils sont divisés en attributs de classes et attributs d'instances.

### 3.4.11

#### **comportement**

indique comment un objet répond à des événements particuliers

### 3.4.12

#### **canal**

liaison unique physique ou logique d'un objet d'application d'entrée ou de sortie d'un serveur vers le processus

### 3.4.13

#### **classe**

ensemble d'objets qui représentent tous le même type de composant de système

Note 1 à l'article: Une classe est une généralisation d'un objet; un modèle pour définir des variables et des méthodes. Tous les objets dans une classe sont identiques selon la forme et le comportement, mais ils contiennent généralement des données différentes dans leurs attributs.

### 3.4.14

#### **attributs de classe**

attribut qui est partagé par tous les objets au sein de la même classe

### 3.4.15

#### **code de classe**

identificateur unique attribué à chaque classe d'objets

### 3.4.16

#### **service spécifique à une classe**

service défini par une classe d'objets particulière pour effectuer une fonction demandée qui n'est pas effectuée par un service commun

Note 1 à l'article: Un objet spécifique à la classe est unique par rapport à la classe d'objets qui le définit

### 3.4.17

#### **client**

- a) objet qui utilise les services d'un autre objet (serveur) pour effectuer une tâche
- b) initiateur d'un message auquel un serveur réagit

### 3.4.18

#### **nœud commandé**

tout nœud, autre que le nœud gérant, dans un réseau de Type 13 est de ce type

#### **3.4.19**

##### **connexion**

liaison logique entre des objets d'application qui peuvent avoir lieu au sein des appareils qui sont les mêmes ou différents

Note 1 à l'article: Les connexions peuvent être soit point à point, soit multipoint.

#### **3.4.20**

##### **canal de connexion**

description d'une connexion entre un récepteur et une source d'éléments de données

#### **3.4.21**

##### **trajet de connexion**

flux d'octets qui définit l'objet d'application auquel une instance de connexion s'applique

#### **3.4.22**

##### **point de connexion**

tampon qui est représenté comme une sous-instance d'un objet d'assemblage

#### **3.4.23**

##### **consommation**

acte de réception des données issues d'un producteur

#### **3.4.24**

##### **consommateur**

nœud ou récepteur qui reçoit des données issues d'un producteur

#### **3.4.25**

##### **application consommatrice**

application qui consomme des données

#### **3.4.26**

##### **commandes de contrôle**

invocations d'action transférées d'un client vers un serveur pour supprimer les sorties, fixer les entrées et/ou synchroniser les sorties

#### **3.4.27**

##### **trajet d'acheminement**

flux unidirectionnel des APDU à travers une relation entre applications

#### **3.4.28**

##### **cyclique**

répétitif d'une manière régulière

#### **3.4.29**

##### **cohérence des données**

moyens pour assurer une transmission cohérente et l'accès d'un objet de données d'entrée ou de sortie entre un client et un serveur, deux clients, ou deux serveurs

#### **3.4.30**

##### **appareil**

connexion matérielle physique à la liaison

Note 1 à l'article: Un appareil peut contenir plusieurs nœuds.

#### **3.4.31**

##### **profil d'appareil**

ensemble des informations et des fonctionnalités dépendantes de l'appareil assurant la cohérence entre les appareils similaires du même type d'appareil



**3.4.32****informations de diagnostic**

toutes les données disponibles sur le serveur pour assurer la maintenance

**3.4.33****nœud d'extrémité**

nœud consommateur ou producteur

**3.4.34****point d'extrémité**

une des entités en communication impliquées dans une connexion

**3.4.35****erreur**

divergence entre une valeur ou une condition calculée, observée ou mesurée et une valeur ou une condition spécifiée ou correcte en théorie

**3.4.36****classe d'erreurs**

groupement général pour les définitions d'erreurs connexes et les codes d'erreurs correspondants

**3.4.37****code d'erreur**

identification d'un type d'erreur spécifique au sein d'une classe d'erreur

**3.4.38****événement**

instance d'un changement de conditions

**3.4.39****variable FIFO**

classe "Variable Object" composée d'un ensemble d'éléments typés de façon homogène, où le premier élément écrit est le premier élément qui peut être lu

Note 1 à l'article: Sur le bus de terrain, il n'y a qu'un seul élément complet qui peut être transféré comme le résultat d'une invocation de services.

**3.4.40****trame**

synonyme déconseillé de DLPU

**3.4.41****groupe**

a) <général> terme général pour un ensemble d'objets

Utilisations spécifiques:

b) <adressage> en décrivant une adresse, celle qui identifie plusieurs entités

**3.4.42****invocation**

acte d'utilisation d'un service ou d'une autre ressource d'un processus d'application

Note 1 à l'article: Chaque invocation représente un fil de contrôle séparé qui peut être décrit par son contexte. Une fois le service terminé ou l'utilisation de la ressource libérée, l'invocation cesse d'exister. Pour les invocations de services, un service initié, mais qui n'est pas encore terminé, est dénommé comme une invocation de service en cours. Dans les invocations de services, un Invoke ID peut être utilisé pour une identification non ambiguë de l'invocation de services et pour sa différenciation par rapport aux autres invocations de services en cours.

### **3.4.43**

#### **index**

adresse d'un objet dans un processus d'application

### **3.4.44**

#### **instance**

apparition physique actuelle d'un objet au sein d'une classe qui identifie un ou plusieurs objets au sein de la même classe d'objets

EXEMPLE California est une instance de la classe d'objet US-state.

Note 1 à l'article: Les termes objet, instance et instance d'objets sont utilisés pour s'appliquer à une instance spécifique.

### **3.4.45**

#### **attributs d'instance**

attribut qui est unique à une instance d'objets et n'est pas partagé par une classe d'objets

### **3.4.46**

#### **instancié**

objet qui a été créé dans un appareil

### **3.4.47**

#### **appareil logique**

une certaine classe de FAL qui extrait un composant logiciel ou un composant de microprogramme sous forme d'un appareil intégré et autonome au sein d'un appareil d'automatisme

### **3.4.48**

#### **ID de fabricant**

identification de chaque fabricant de produits par un numéro unique

### **3.4.49**

#### **informations de gestion**

informations accessibles par le réseau qui soutiennent la gestion de l'opération du système de bus de terrain, y compris la couche application

Note 1 à l'article: La gestion comprend des fonctions telles que commande, surveillance et diagnostic

### **3.4.50**

#### **routeur de messages**

objet dans un nœud qui diffuse des demandes de messagerie à des objets d'application appropriés

### **3.4.51**

#### **nœud de gestion**

Un nœud d'un réseau de type 13 doit agir en tant que nœud gérant

### **3.4.52**

#### **connexion multipoint**

connexion à partir d'un nœud vers plusieurs nœuds

Note 1 à l'article: Les connexions multipoint permettent des messages issus d'un seul producteur et reçus par plusieurs nœuds consommateurs.

### **3.4.53**

#### **réseau**

ensemble de nœuds connectés par un support de communication, y compris tous les répéteurs, ponts, routeurs et passerelles de couche inférieure

**3.4.54****nœud**

hôte d'un AP, identifié par une adresse de DLCEP unique

**3.4.55****objet**

représentation abstraite d'un composant particulier au sein d'un appareil, généralement un ensemble de données connexes (sous forme des variables) et de méthodes (procédures) pour exploiter ces données qui ont nettement défini une interface et un comportement

**3.4.56****dictionnaire d'objets**

ensemble de définitions, de paramètres et d'attributs spécifiques à la communication, et de données dépendant de l'application

**3.4.57****service spécifique à un objet**

service unique à une classe d'objets qui le définit

**3.4.58****origine**

client responsable de l'établissement d'un trajet de connexion vers la cible

**3.4.59****appareil physique**

appareil d'automation ou un autre appareil de réseau

**3.4.60****connexion point à point**

connexion qui existe entre deux objets d'application

**3.4.61****point d'extrémité d'AR préétabli**

point d'extrémité d'AR dans un état établi au cours de la configuration des AE qui commandent ses points d'extrémités

**3.4.62****données de traitement**

objet(s) qui est (sont) déjà prétraité(s) et transféré(s) cycliquement dans le but d'informations ou d'un traitement ultérieur

**3.4.63****production**

acte d'envoi des données à destination d'un consommateur

**3.4.64****producteur**

nœud responsable de l'envoi des données

**3.4.65****propriété**

terme général pour les informations descriptives relatives un objet

**3.4.66****fournisseur**

source d'une connexion de données

**3.4.67****éditeur**

rôle d'un point d'extrémité d'AR transmettant des APDU sur le bus de terrain pour être consommées par un ou plusieurs abonnés

Note 1 à l'article: Un éditeur peut ne pas connaître l'identité ou le nombre des abonnés.

**3.4.68****gestionnaire d'édition**

rôle d'un point d'extrémité d'AR dans lequel il émet une ou plusieurs APDU de demande de service confirmé vers un éditeur pour lui demander d'éditer un objet spécifié

Note 1 à l'article: Deux types de gestionnaires sont définis par la présente norme: les abonnés par extraction (pull) et les abonnés par émission (push); les deux font l'objet d'une définition distincte

**3.4.69****éditeur par émission**

type d'un éditeur qui édite un objet dans une APDU de demande de service non confirmé

**3.4.70****gestionnaire d'édition par émission**

type de gestionnaire d'édition qui demande qu'un objet spécifié soit édité à l'aide d'un service non confirmé

**3.4.71****abonné par émission**

type d'abonné qui reconnaît des APDU de demande de service non confirmé comme des données d'objets éditées

**3.4.72****serveur**

- a) rôle d'un AREP dans lequel il renvoie une APDU de réponse de service confirmé vers le client ayant initié la demande
- b) objet qui fournit des services à un autre objet (client)

**3.4.73****service**

opération ou fonction qu'un objet et/ou une classe d'objets effectue sur demande d'un autre objet et/ou d'une classe d'objets

**3.4.74****abonné**

rôle d'un AREP dans lequel il reçoit des APDU produites par un éditeur

**3.4.75****cible**

nœud d'extrémité sur lequel une connexion est établie

**3.5 Abréviations et symboles**

AE	Application entity (Entité d'application)
AL	Application layer (Couche application)
ALME	Application layer management entity (Entité de gestion de couche application)
ALP	Application layer protocol (Protocole de couche application)
APO	Application objet (Objet d'application)
AP	Application process (Processus d'application)

APDU	Application protocol data unit (Unité de données de protocole d'application)
AR	Application relationship (Relation entre applications)
AREP	Application relationship end point (Point d'extrémité de relations entre applications)
ASCII	American Standard Code for Information Interchange (Code de normalisation américain pour l'échange d'informations)
ASE	Application service element (Élément de service d'application)
BNB-PEC	Buffered network-scheduled bi-directional pre-established connection (Connexion préétablie bidirectionnelle tamponnée programmée sur le réseau)
BNU-PEC	Buffered network-scheduled uni-directional pre-established connection (Connexion préétablie unidirectionnelle tamponnée programmée sur le réseau)
CID	Connection ID (ID de connexion)
CN	Controlled node (Nœud commandé)
Cnf	Confirmation (Confirmation)
DL-	Préfixe désignant la liaison de données
DLCEP	Data-link connection end point (Point d'extrémité de connexion de liaison de données)
DLL	Data-link layer (Couche liaison de données)
DLM	Data-link-management (Gestion de liaison de données)
DLSAP	Data-link service access point (Point d'accès au service de liaison de données)
DLSDU	DL-service-data-unit (Unité de données du service de DL)
DNS	Domain name service (Service de noms de domaine)
FAL	Fieldbus application layer (Couche application de bus de terrain)
Ind	Indication (Indication)
IP	Internet protocol (Protocole Internet) (voir RFC 791)
MN	Managing node (Nœud gérant)
NMT	Network management (Gestion de réseau)
OD	Object dictionary (Dictionnaire d'objets)
PDO	Process data object (Objet de données de traitement)
PReq	PollRequest (Type 13 frame type) (Demande d'interrogation (type de trame de Type 13))
Pres	PollRequest (Type 13 frame type) (Réponse d'interrogation (type de trame de Type 13))
QUB-CL	Queued user-triggered bi-directional connectionless (En file d'attente, déclenché par un utilisateur, bidirectionnel et sans connexion)
QUB-COS	Queued user-triggered bi-directional connection-oriented with segmentation (En file d'attente, déclenché par un utilisateur, bidirectionnel orienté connexion avec segmentation)
QUU	Queued user-triggered uni-directional (En file d'attente, déclenché par un utilisateur et unidirectionnel)
Req	Request (Demande)
RPDO	Receive process data object (Objet de données de traitement de réception)
Rsp	Response (Réponse)
SDO	Service data object (Objet de données de service)
TPDO	Transmit process data object (Objet de données de traitement de

transmission)

### 3.6 Conventions

#### 3.6.1 Vue d'ensemble

La FAL est définie comme un ensemble des ASE orientés objet. Chaque ASE est spécifié dans un paragraphe séparé. Chaque spécification d'ASE est divisée en deux parties: la spécification des classes et la spécification des services.

La spécification de la classe définit les attributs de la classe. L'accès à ces attributs n'est pas traité dans le présent document à l'exception des spécifications précises. La spécification des services définit les services qui sont fournis par l'ASE.

#### 3.6.2 Conventions générales

La présente norme utilise les conventions descriptives données dans l'ISO/CEI 10731.

#### 3.6.3 Conventions relatives aux définitions de classe

Les définitions des classes sont décrites à l'aide des modèles. Chaque modèle est composé d'une liste d'attributs pour une classe. La forme générale du modèle est représentée ci-dessous:

<b>FAL ASE:</b>		<b>Nom d'ASE</b>
<b>CLASS:</b>	<b>Nom de la classe</b>	
<b>CLASS ID:</b>	<b>#</b>	
<b>PARENT CLASS:</b>	<b>Nom de la classe parente</b>	
<b>ATTRIBUTES:</b>		
1	(o) Attribut clé:	identificateur numérique
2	(o) Attribut clé:	nom
3	(m) Attribut:	nom d'attribut(valeurs)
4	(m) Attribut:	nom d'attribut(valeurs))
4.1	(s) Attribut:	nom d'attribut(valeurs)
4.2	(s) Attribut:	nom d'attribut(valeurs)
4.3	(s) Attribut:	nom d'attribut(valeurs)
5.	(c) Contrainte:	expression de la constante
5.1	(m) Attribut:	nom d'attribut(valeurs)
5.2	(o) Attribut:	nom d'attribut(valeurs)
6	(m) Attribut:	nom d'attribut(valeurs)
6.1	(s) Attribut:	nom d'attribut(valeurs)
6.2	(s) Attribut:	nom d'attribut(valeurs)
<b>SERVICES:</b>		
1	(o) OpsService:	nom de service
2	(c) Contrainte:	expression de la contrainte
2.1	(o) OpsService:	nom de service
3	(m) MgtService:	nom de service

- (1) "FAL ASE": l'entrée est le nom de l'ASE de FAL qui fournit les services pour la classe spécifiée.
- (2) "CLASS": l'entrée est le nom de la classe spécifiée. Tous les objets définis, qui utilisent ce modèle, font l'objet d'une instance de cette classe. La classe peut être spécifiée par la présente norme, ou par un utilisateur de la présente norme.
- (3) "CLASS ID": l'entrée est un numéro qui identifie la classe spécifiée. Ce numéro n'est pas utilisé pour les éléments de Type 13.

- (4) "PARENT CLASS": l'entrée est le nom de la classe parente pour la classe spécifiée. Tous les attributs définis pour la classe parente et hérités par elle font l'objet d'un héritage pour la classe définie. Par conséquent, ils ne doivent pas être redéfinis dans le modèle pour cette classe.

NOTE La classe parente "TOP" indique que la classe définie fait l'objet d'une définition de la classe initiale. La classe parente TOP est utilisée comme un point de départ pour la définition de toutes les autres classes. L'utilisation de TOP est réservée pour des classes définies par la présente norme.

- (5) L'étiquette "ATTRIBUTES" (ATTRIBUTS) indique que les entrées suivantes sont des attributs définis pour la classe.
- Chacune des entrées d'attributs contient un numéro de la ligne dans la colonne 1, "mandatory" (obligatoire) (m) / "optional" (facultatif) (o) / "conditional" conditionnel (c) / "selector indicator" (indicateur de sélecteur) (s) dans la colonne 2, étiquette de type d'attributs dans la colonne 3, nom ou expression conditionnelle dans la colonne 4, et une liste facultative des valeurs énumérées dans la colonne 5. Dans la colonne qui suit la liste des valeurs, la valeur par défaut pour l'attribut peut être spécifiée.
  - Des objets sont généralement identifiés par un identificateur numérique ou par un nom d'objet, ou par les deux. Dans les modèles des classes, ces attributs clé sont définis comme l'attribut clé.
  - Le numéro de la ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication est utilisée pour spécifier
    - des champs d'un attribut structuré (4.1, 4.2, 4.3),
    - des attributs conditionnels sur un énoncé de contrainte (5). Les attributs peuvent être obligatoires (5.1) ou facultatifs (5.2), si la contrainte est vraie. Tous les attributs ne nécessitent pas tous les énoncés de contrainte comme c'est le cas de l'attribut défini en (5.2).
    - des champs de sélection d'un attribut de type de choix (6.1 et 6.2).
- (6) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe.
- Une (m) dans la colonne 2 indique que le service est obligatoire pour la classe, alors qu'une (o) indique qu'il est optionnel. Une (c) dans cette colonne indique que le service est conditionnel. Lorsque tous les services déterminés pour une classe sont considérés comme facultatifs, au moins l'un d'entre eux est à sélectionner quand une instance de la classe est définie.
  - L'étiquette "OpsService" désigne un service opérationnel (1).
  - L'étiquette "MgtService" désigne un service de gestion (2).
  - Le numéro de la ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication dans la liste de services est utilisée pour spécifier des services conditionnels sur un énoncé de contrainte.

### 3.6.4 Conventions relatives aux définitions de service

#### 3.6.4.1 Généralités

Le modèle de service, les primitives de service, et les diagrammes de séquence temporelle utilisés sont des descriptions totalement abstraites; ils ne constituent pas une spécification pour une mise en œuvre.

#### 3.6.4.2 Paramètre du service

Les primitives de services sont utilisées pour représenter des interactions entre un utilisateur de service et un fournisseur de service (ISO/CEI 10731). Elles acheminent des paramètres qui indiquent des informations disponibles dans l'interaction entre un utilisateur et un fournisseur. Dans n'importe quelle interface particulière, il n'est pas indispensable d'énoncer tous les paramètres de façon explicite.

Les spécifications de service de la présente norme utilisent un format tabulaire pour décrire les paramètres de composants des primitives de service d'ASE. Les paramètres qui s'appliquent à chaque groupe des primitives de services sont établis dans les tableaux. Chaque tableau est divisé en cinq colonnes:

- 1) nom du paramètre,
- 2) primitive "request",
- 3) primitive "indication",
- 4) primitive "response", et
- 5) primitive "confirm".

Un paramètre (ou un composant de celui-ci) est énuméré dans chaque ligne de chaque tableau. Dans les colonnes appropriées de la primitive de service, un code est utilisé pour spécifier le type d'usage du paramètre sur la primitive spécifiée dans la colonne:

- M le paramètre est obligatoire pour la primitive.
- U le paramètre est une option d'utilisateur et peut être fourni ou pas en fonction de l'utilisation dynamique de l'utilisateur du service. Lorsqu'il n'est pas fourni, une valeur par défaut est supposée pour le paramètre.
- C le paramètre est conditionnel en fonction d'autres paramètres ou de l'environnement de l'utilisateur du service.
- (blanc/vide) le paramètre n'est jamais présent.
- S le paramètre est un élément sélectionné.

Certaines entrées sont caractérisées par des symboles entre parenthèses. Ceux-ci peuvent être

- a) une contrainte spécifique au paramètre:
  - "(=)" indique que le paramètre est équivalent sémantiquement au paramètre dans la primitive de service située immédiatement à sa gauche dans le tableau.
- b) indique qu'une certaine note s'applique à l'entrée:
  - "(n)" indique que la note "n" suivante contient des informations supplémentaires se rapportant au paramètre et à son utilisation.

#### Procédures du service

Les procédures sont définies en termes

- des interactions entre les entités d'application à travers l'échange des unités de données du protocole d'application (APDU) de bus de terrain, et
- des interactions entre un fournisseur du service de couche application et un utilisateur du service de couche application dans le même système à travers l'invocation des primitives du service de couche application.

Ces procédures peuvent être appliquées à des instances de communication entre les systèmes qui prennent en charge des services de communications à contrainte de temps au sein de la couche application de bus de terrain.

## 4 Concepts

### 4.1 Concepts typiques

L'Article 9 de la CEI 61158-1 est incorporé par référence, sauf dans les cas d'annulation spécifique en 4.2.



## 4.2 Concepts spécifiques à un type

### 4.2.1 Nœud gérant et Nœuds commandés

Le nœud qui gère l'autorisation d'envoyer des messages vers Ethernet est appelé "Managing Node" (Nœud gérant). Seul le MN peut envoyer des messages indépendamment, c'est-à-dire, pas comme une réponse à un message reçu. Tous les nœuds disponibles dans le réseau sont configurés sur le MN. Un seul MN actif est autorisé dans le réseau.

Tous les autres nœuds transmettent seulement au sein des créneaux de communication attribués par le MN. Ils sont appelés "Controlled Nodes" (CN) (Nœuds commandés). Les CN sont des nœuds de bus passifs qui n'envoient que lorsqu'ils sont demandés par le MN. Il existe deux types de CN:

- Les CN isochrones sont capables de communiquer via les AR déclenchées par l'utilisateur et celles programmées par le réseau.
- Les CN asynchrones ne communiquent que via les AR déclenchées par l'utilisateur.

La séquence et l'attribution des créneaux temporels pour des interactions déclenchées par l'utilisateur et programmées par le réseau sont spécifiées dans la CEI 61158-3-13 et la CEI 61158-4-13.

### 4.2.2 Nœud, AP et Dictionnaire d'objets

Chaque nœud accueille exactement un AP. Tous les APO pour cet AP sont rassemblés dans le dénommé Dictionnaire d'objets (OD). La structure globale de l'OD est décrite dans le Tableau 1.

**Tableau 1 – Structure globale de l'OD**

Zone	Contenu
Zone du type de données	Définition des types de données
Zone du profil de communication	Contient des paramètres spécifiques à la communication pour le réseau de Type 13. Ces entrées sont typiques pour tous les appareils
Zone spécifique à un fabricant	Définition des variables spécifiques à un fabricant
Zone du profil de l'appareil	Définition des variables définies dans un profil de l'appareil (qui ne relève pas de la présente norme)
Zone réservée	Réservée pour une future utilisation.

### 4.2.3 ASE de gestion d'objets

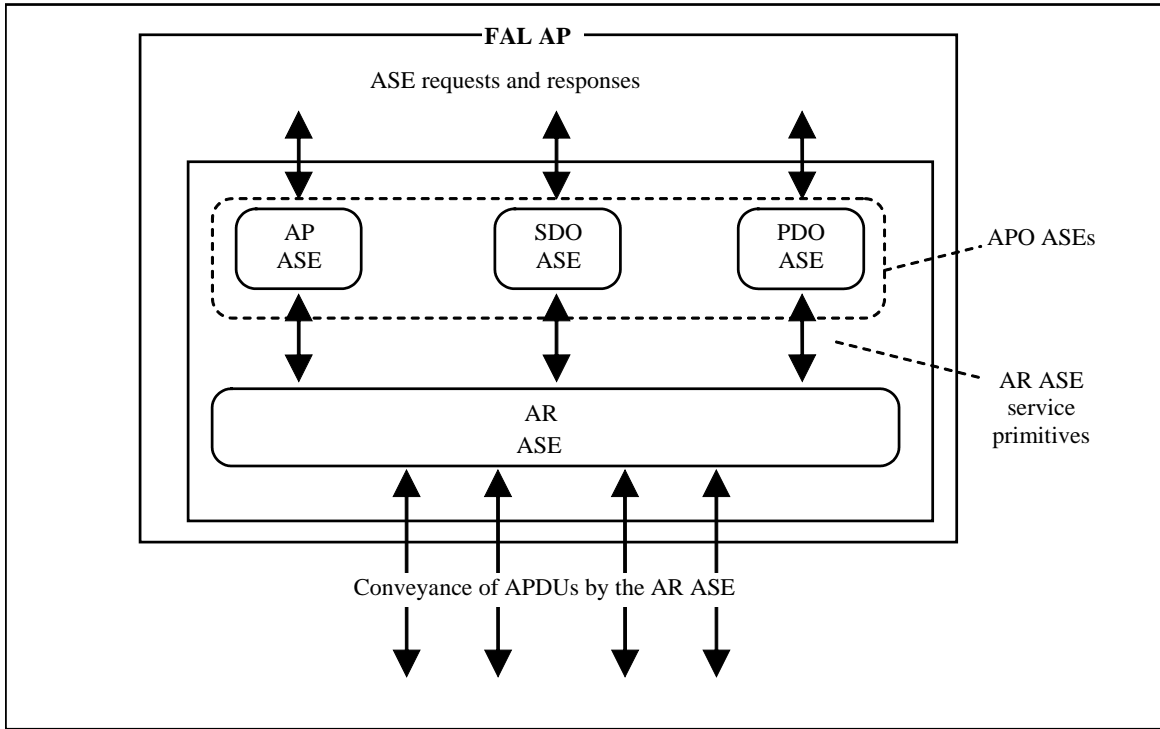
Il n'existe pas d'ASE de gestion d'objets défini pour le Type 13.

### 4.2.4 Opportunité

La fonction "timeliness" n'est pas mise en œuvre dans la FAL de Type 13. La surveillance et le contrôle de la synchronisation de cycles sont gérés par la DLL, voir la CEI 61158-3-13 et la CEI 61158-4-13.

### 4.2.5 ASE d'APO

Les ASE de FAL de Type 13 et leurs interrelations sont décrits dans la Figure 1.



**Légende**

Anglais	Français
ASE requests and responses	Demandes et réponses d'ASE
AR ASE service primitives	Primitives du service d'ASE d'AR
Conveyance of APDUs by the AR ASE	Acheminement des APDU par l'ASE d'AR

**Figure 1 – ASE de Type 13**

**5 Data type ASE**

**5.1 Généralités**

Le Paragraphe 10.1 de la CEI 61158-1 est incorporé par référence avec une restriction: Seul le niveau d'imbrication de 1 (voir 10.1.4.4 de la CEI 61158-1) est pris en charge.

**5.2 Définition formelle des objets de data type**

Le Paragraphe 10.2 de la CEI 61158-1 est incorporé par référence.

**5.3 Types de données de FAL définis**

**5.3.1 Types de longueur fixe**

**5.3.1.1 Types booléens**

<b>CLASS:</b>	<b>Data type</b>
<b>ATTRIBUTES:</b>	
1 Data type numeric identifier	= 1
2 Data type name	= Boolean
3 Format	= FIXED LENGTH
3.1 Octet length	= 1

Ce type de données exprime un type de données booléen avec les valeurs "TRUE" et "FALSE".

### 5.3.1.2 Types Date/Time

#### 5.3.1.2.1 TimeOfDay

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 12
2	Data type name	= TimeOfDay
3	Format	= FIXED LENGTH
3.1	Octet length	= 6

Composé de deux éléments des valeurs non signées, ce type de données exprime l'heure d'un jour et la date. Le premier élément est le type de données Unsigned32 qui donne l'heure après minuit en millisecondes. Le deuxième élément est le type de données Unsigned16 qui donne la date en comptant les jours à partir du 1 janvier 1984.

#### 5.3.1.2.2 TimeDifference

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 13
2	Data type name	= TimeDifference
3	Format	= FIXED LENGTH
3.1	Octet length	= 4 ou 6

Ce type de données est composé de deux éléments des valeurs non signées qui expriment la différence de temps. Le premier élément est le type de données Unsigned32 qui fournit la partie fractionnaire d'un jour en millisecondes. Le deuxième élément facultatif est un type de données Unsigned16 qui fournit la différence en jours.

### 5.3.1.3 Types numériques

#### 5.3.1.3.1 Real32

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 8
2	Data type name	= Real32
3	Format	= FIXED LENGTH
3.1	Octet length	= 4

Ce type a une longueur de quatre octets. Le format pour Real32 est celui défini par l'IEEE 754 comme «simple précision».

#### 5.3.1.3.2 Real64

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 17
2	Data type name	= Real64
3	Format	= FIXED LENGTH
3.1	Octet length	= 8

Ce type a une longueur de huit octets. Le format pour Real64 est celui défini par l'IEEE 754 comme «double précision».

### 5.3.1.3.3 Types entiers

#### 5.3.1.3.3.1 Integer8

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 2
2	Data type name	= Integer8
3	Format	= FIXED LENGTH
4.1	Octet length	= 1

Ce type entier est un nombre binaire en complément à deux avec une longueur d'un octet.

#### 5.3.1.3.3.2 Integer16

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 3
2	Data type name	= Integer16
3	Format	= FIXED LENGTH
3.1	Octet length	= 2

Ce type entier est un nombre binaire en complément à deux avec une longueur de deux octets.

#### 5.3.1.3.3.3 Integer24

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 16
2	Data type name	= Integer24
3	Format	= FIXED LENGTH
3.1	Octet length	= 3

Ce type entier est un nombre binaire en complément à deux avec une longueur de trois octets.

#### 5.3.1.3.3.4 Integer32

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 4
2	Data type name	= Integer32
3	Format	= FIXED LENGTH
3.1	Octet length	= 4

Ce type entier est un nombre binaire en complément à deux avec une longueur de quatre octets.

#### 5.3.1.3.3.5 Integer40

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 18
2	Data type name	= Integer40
3	Format	= FIXED LENGTH
3.1	Octet length	= 5

Ce type entier est un nombre binaire en complément à deux avec une longueur de cinq octets.

**5.3.1.3.3.6 Integer48**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 19
2	Data type name	= Integer48
3	Format	= FIXED LENGTH
3.1	Octet length	= 6

Ce type entier est un nombre binaire en complément à deux avec une longueur de six octets.

**5.3.1.3.3.7 Integer56**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 20
2	Data type name	= Integer56
3	Format	= FIXED LENGTH
3.1	Octet length	= 7

Ce type entier est un nombre binaire en complément à deux avec une longueur de sept octets.

**5.3.1.3.3.8 Integer64**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 21
2	Data type name	= Integer64
3	Format	= FIXED LENGTH
3.1	Octet length	= 8

Ce type entier est un nombre binaire en complément à deux avec une longueur de huit octets.

**5.3.1.3.4 Types non signés****5.3.1.3.4.1 Unsigned8**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 5
2	Data type name	= Unsigned8
3	Format	= FIXED LENGTH
3.1	Octet length	= 1

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type a une longueur d'un octet.

**5.3.1.3.4.2 Unsigned16**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 6
2	Data type name	= Unsigned16
3	Format	= FIXED LENGTH
3.1	Octet length	= 2

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de deux octets.

**5.3.1.3.4.3 Unsigned24**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	22
2	Data type name	=	Unsigned24
3	Format	=	FIXED LENGTH
3.1	Octet length	=	3

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de trois octets.

**5.3.1.3.4.4 Unsigned32**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	7
2	Data type name	=	Unsigned32
3	Format	=	FIXED LENGTH
3.1	Octet length	=	4

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de quatre octets.

**5.3.1.3.4.5 Unsigned40**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	24
2	Data type name	=	Unsigned40
3	Format	=	FIXED LENGTH
3.1	Octet length	=	5

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de cinq octets.

**5.3.1.3.4.6 Unsigned48**

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	25
2	Data type name	=	Unsigned48
3	Format	=	FIXED LENGTH
3.1	Octet length	=	6

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de six octets.

**5.3.1.3.4.7 Unsigned56**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 26
2	Data type name	= Unsigned56
3	Format	= FIXED LENGTH
3.1	Octet length	= 7

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de sept octets.

**5.3.1.3.4.8 Unsigned64**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 27
2	Data type name	= Unsigned64
3	Format	= FIXED LENGTH
3.1	Octet length	= 8

Ce type représente un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort d'un nombre binaire; aucun bit de signe n'est compris. Ce type non signé a une longueur de huit octets.

**5.3.2 Types de chaîne de caractères****5.3.2.1 OctetString**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 10
2	Data type name	= OctetString
3	Format	= STRING
3.1	Octet length	= 1 to n

OctetString est une séquence d'octets ordonnée numérotée de 1 à n. Pour les besoins du débat, l'octet 1 de la séquence se réfère au premier octet.

NOTE La CEI 61158-6-13 définit l'ordre de la transmission.

**5.3.2.2 VisibleString**

<b>CLASS:</b>		<b>Data type</b>
<b>ATTRIBUTES:</b>		
1	Data type numeric identifier	= 9
2	Data type name	= VisibleString
3	Format	= STRING
3.1	Octet length	= 1 to n

Ce type est défini comme le type de chaîne de caractères de l'ISO/IEC 646.

### 5.3.2.3 UnicodeString

<b>CLASS:</b>		<b>Data type</b>	
<b>ATTRIBUTES:</b>			
1	Data type numeric identifier	=	11
2	Data type name	=	UnicodeString
3	Format	=	STRING
3.1	Octet length	=	1 to n

Ce type est défini comme le type string (chaîne de caractères) UNICODE.

### 5.4 Spécification du service de l'ASE Data type

Il n'existe pas de services opérationnels définis pour l'objet type.

## 6 Spécification du modèle de communication de Type 13

### 6.1 ASE

#### 6.1.1 ASE de processus d'application

##### 6.1.1.1 Vue d'ensemble

La présente norme modélise un processus d'application (AP) de bus de terrain. Les processus d'application de bus de terrain représentent les ressources de traitement et d'informations d'un système qui peut être évalué via les services de FAL.

L'élément de service d'application (ASE) dans la FAL, qui fournit ces services, s'appelle Application Process ASE (ASE de processus application). Dans l'ASE d'AP, l'AP est modélisé et accessible comme un APO avec un identificateur normalisé et prédéfini.

##### 6.1.1.2 Spécification de la classe AP

###### 6.1.1.2.1 Modèle formel

La classe AP spécifie les attributs et les services définis pour des processus d'application. Sa classe parente "top" indique le sommet de l'arbre des classes de FAL.



<b>ASE:</b>		<b>ASE AP</b>
<b>CLASS:</b>		<b>AP</b>
<b>CLASS ID:</b>		<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>		<b>TOP</b>
<b>ATTRIBUTES:</b>		
1	(m) Attribut:	NMT-status
2	(m) Attribut:	Type 13 fieldbus-version
3	(m) Attribut:	feature-flags
4	(m) Attribut:	MTU
5	(m) Attribut:	cycle-timing
6	(m) Attribut:	identity
7	(m) Attribut:	verify-configuration
8	(o) Attribut:	application-software-version
9	(c) Attribut:	IP-address
10	(c) Attribut:	host-name
11	(o) Attribut:	vendor-specific-extension
12	(m) Attribut:	static-error
13	(o) Attribut:	error-history
14	(m) Attribut:	sync-control
15	(m) Attribut:	sync-status
<b>SERVICES:</b>		
1	(m) OpsService:	Ident
2	(m) OpsService:	Statut
3	(m) OpsService:	Sync
4	(m) OpsService:	NMT-req-invite
5	(m) OpsService:	NMT-state-command
6	(m) OpsService:	NMT-info

#### 6.1.1.2.2 Attributs

##### **NMT-status**

Cet attribut rapporte le statut actuel du diagramme d'états de NMT des CN.

##### **Type 13 fieldbus-version**

Cet attribut indique la version de bus de terrain de Type 13 à laquelle le CN correspond.

##### **feature-flags**

Cet attribut contient des informations sur le comportement de l'appareil dans le cycle de bus de terrain de Type 13, la mise en œuvre des fonctions de NMT et sa capacité d'agir en tant que routeur. La spécification détaillée de cet attribut est incluse dans la CEI 61158-6-13.

##### **MTU**

Cet attribut rapporte la taille de la plus grande trame IP qui peut être transmise sur le réseau, y compris la taille de l'en-tête de transport.

##### **cycle-timing**

Cet attribut spécifie la performance de synchronisation du CN au cours de la phase isochrone du cycle de bus de terrain de Type 13.

##### **identity**

Cet attribut spécifie le type du CN et le fournisseur ID. Cet attribut, comme un supplément facultatif, peut également contenir le code de produit, le numéro de révision et le numéro de série du CN.

##### **verify-configuration**

Cet attribut contient la date de la configuration et l'heure du nœud.

**application-software-version**

Cet attribut facultatif contient la date et l'heure de la dernière mise à jour pour le logiciel d'application du CN.

**IP-address**

Cet attribut fournit des informations sur l'adresse IP du CN. Il n'est utilisé que si IP est pris en charge par l'appareil.

**host-name**

Cet attribut rapporte le nom d'hôte DNS actuel du CN. Il n'est utilisé que si IP est pris en charge par l'appareil.

**vendor-specific-extension**

Cet attribut facultatif est utilisé pour un objectif spécifique à un fournisseur.

**static-error**

Cet attribut indique une erreur statique sur le CN et son type.

**error-history**

Cet attribut documente des événements d'erreur antécédents. Il contient le type d'erreur, l'heure de son apparition, et des informations supplémentaires.

**sync-control**

Cet attribut contient des données nécessaires pour la configuration/synchronisation d'un CN avec une PRes en temps déclenché.

**sync-status**

Cet attribut contient des données nécessaires pour la synchronisation d'un CN avec une PRes en temps déclenché.

**6.1.1.2.3 Services****Ident**

Ce service est utilisé pour récupérer des caractéristiques pertinentes des CN.

**Status**

Ce service est utilisé pour interroger le statut actuel et les données d'erreurs issus des CN.

**Sync**

Ce service est utilisé pour configurer un CN avec une PRes en temps déclenché.

**NMT-req-invite**

À l'aide de ce service, le MN invite un CN à envoyer une demande de commande de NMT.

**NMT-state-command**

Ce service est utilisé pour contrôler le statut opérationnel des CN.

**NMT-info**

Ce service est utilisé pour éditer des informations pertinentes de système vers les CN dans le réseau.

**6.1.1.3 Spécification des services d'AP ASE****6.1.1.3.1 Services pris en charge**

Le Paragraphe 6.1.1.3 contient la définition des services uniques à cet ASE. Les services définis pour cet ASE sont

- Ident
- Status
- Sync
- NMT-req-invite

- NMT-state-command
- NMT-info services

### 6.1.1.3.2 Service Ident

#### 6.1.1.3.2.1 Vue d'ensemble

Le service Ident est utilisé par le MN pour identifier des CN configurés, mais non reconnus, au démarrage du système ou après la perte de communication. Le service peut être utilisé après le démarrage pour demander des informations d'installation du CN.

#### 6.1.1.3.2.2 Primitives du service

Les paramètres de ce service sont représentés dans le Tableau 2.

**Tableau 2 – Service Ident**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
Result(+)			S	S (=)
NMT-status			M	M (=)
Type 13 fieldbus-version			M	M (=)
feature-flags			M	M (=)
cycle-timing			M	M (=)
Identity			M	M (=)
verify-configuration			M	M (=)
application-software-version			M	M (=)
IP-address			M	M (=)
host-name			M	M (=)
vendor-specific-extensions			M	M (=)
Result(-)			S	S (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

#### Argument

L'argument achemine les paramètres spécifiques au service de la demande de service.

#### Result(+)

Ce paramètre du type de sélection indique que la demande de service a réussi.

#### NMT-status

Ce paramètre spécifie la valeur de l'attribut NMT-status de l'AP.

#### Type 13 fieldbus-version

Ce paramètre spécifie la valeur de l'attribut fieldbus-version de Type 13 de l'AP.

#### feature-flags

Ce paramètre spécifie la valeur de l'attribut feature-flags de l'AP.

**cycle-timing**

Ce paramètre spécifie la valeur de l'attribut cycle-timing de l'AP.

**identity**

Ce paramètre spécifie la valeur de l'attribut identity de l'AP.

**verify-configuration**

Ce paramètre spécifie la valeur de l'attribut verify-configuration de l'AP.

**application-software-version**

Ce paramètre spécifie la valeur de l'attribut application-software-version de l'AP. Ce paramètre est à remplir de zéros, si l'attribut n'est pas utilisé.

**IP-address**

Ce paramètre spécifie la valeur de l'attribut IP-address de l'AP. Ce paramètre est à remplir de zéros, si l'attribut n'est pas utilisé.

**host-name**

Ce paramètre spécifie la valeur de l'attribut host-name de l'AP. Ce paramètre est à remplir de zéros, si l'attribut n'est pas utilisé.

**vendor-specific-extensions**

Ce paramètre spécifie la valeur de l'attribut vendor-specific-extensions de l'AP. Ce paramètre est à remplir de zéros, si l'attribut n'est pas utilisé.

**Result(-)**

Ce paramètre du type de sélection indique que la demande de service a échoué.

**6.1.1.3.2.3 Procédure du service**

Cette procédure du service est une séquence des deux services non confirmés consécutifs (tels que spécifiés dans la CEI 61158-1, 9.6.2) dans des directions opposées.

Des APDU contenant la réponse Ident (c'est-à-dire, Result(+)) sont envoyées via les AR de multidiffusion. Le MN reçoit des réponses Ident; les CN peuvent recevoir des réponses Ident, s'ils sont configurés pour cela.

Le MN vérifie la nouvelle réponse Ident avec ses propres informations sur le nœud demandé. En cas de différences, il tente de reconfigurer le nœud avec ses données ou, si impossible, d'envoyer un signal d'erreur à son utilisateur.

**6.1.1.3.3 Service Status****6.1.1.3.3.1 Vue d'ensemble**

Le service Status est utilisé par le MN pour demander des informations de statut issues du CN sur le statut actuel et les données d'erreurs.

**6.1.1.3.3.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 3.

**Tableau 3 – Service Status**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
Result(+)			S	S (=)
NMT-status			M	M (=)
static-error			M	M (=)
error-history			U	U (=)
Result(-)			S	S (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**Result(+)**

Ce paramètre du type de sélection indique que la demande de service a réussi.

**NMT-status**

Ce paramètre spécifie la valeur de l'attribut NMT-status de l'AP.

**static-error**

Ce paramètre spécifie la valeur de l'attribut static-error de l'AP.

**error-history**

Ce paramètre spécifie la valeur de l'attribut error-history de l'AP.

**Result(-)**

Ce paramètre du type de sélection indique que la demande de service a échoué.

**6.1.1.3.3.3 Procédure du service**

Cette procédure du service est une séquence des deux services non confirmés consécutifs (tels que spécifiés dans la CEI 61158-1, 9.6.2) dans des directions opposées.

Des APDU contenant la réponse Status (c'est-à-dire, Result(+)) sont envoyées via les AR de multidiffusion. Le MN reçoit des réponses Status; les CN peuvent recevoir des réponses Status, s'ils sont configurés pour cela.

**6.1.1.3.4 Service Sync****6.1.1.3.4.1 Vue d'ensemble**

Le service Sync est utilisé par le MN pour configurer/synchroniser un CN avec une PRes en temps déclenché.

**6.1.1.3.4.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 4.

**Tableau 4 – Service Sync**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
sync-control	M	M (=)		
Result				
sync-status			M	M (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**sync-control**

Ce paramètre spécifie la valeur de l'attribut sync-control de l'AP.

**Result**

Ce paramètre du type de sélection indique le résultat de la demande de service.

**Sync-status**

Ce paramètre spécifie la valeur de l'attribut sync-status de l'AP.

**6.1.1.3.4.3 Procédure du service**

Cette procédure du service est une séquence des deux services non confirmés consécutifs (tels que spécifiés dans la CEI 61158-1, 9.6.2) dans des directions opposées.

Des APDU contenant la réponse Sync (c'est-à-dire, Result) sont envoyées via les AR de multidiffusion. Le MN reçoit des réponses Sync; les CN peuvent recevoir des réponses Sync, s'ils prennent en charge l'envoi en temps déclenché des PRes.

**6.1.1.3.5 Service NMT-req-invite**

**6.1.1.3.5.1 Vue d'ensemble**

Le MN utilise ce service pour savoir quel service suivant, souhaité par le CN, doit être effectué par le MN.

**6.1.1.3.5.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 5.

**Tableau 5 – Service NMT-req-invite**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument AREP	M	M (=)		
Result(+)			S	S (=)
command-ID			M	M (=)
target-node			M	M (=)
Data			M	M (=)
Result(-)			S	S (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**Result(+)**

Ce paramètre du type de sélection indique que la demande de service a réussi.

**command-ID**

Ce paramètre spécifie l'ID de la commande demandée.

**target-node**

Ce paramètre spécifie le nœud cible de la commande de NMT demandée.

**data**

Ce paramètre spécifie des données spécifiques à la commande qui doivent être émises par le MN.

**Result(-)**

Ce paramètre du type de sélection indique que la demande de service a échoué.

**6.1.1.3.5.3 Procédure du service**

Ce service est déclenché par une notification précédente "Ready-to-send" qui est émise par un CN. Après cette notification, le MN adresse le CN en appel à la prochaine occasion.

Cette procédure du service est une séquence des deux services non confirmés consécutifs (tels que spécifiés dans la CEI 61158-1, 9.6.2) dans des directions opposées.

Ce service est toujours "unicast".

**6.1.1.3.6 Service NMT-state-command****6.1.1.3.6.1 Vue d'ensemble**

La gestion de réseau (NMT) est orientée nœud en suivant la relation entre un maître et un esclave.

La fonction du maître NMT est effectuée par le MN. Les CN sont administrés comme les esclaves NM par le maître.

Le MN utilise des services NMT-state-command pour contrôler le(s) diagramme(s) d'états du (des) CN.

### 6.1.1.3.6.2 Primitives du service

Les paramètres de ce service sont représentés dans le Tableau 6.

**Tableau 6 – Service NMT-state-command**

Nom du paramètre	Req	Ind
Argument		
AREP	M	M (=)
command-ID	M	M (=)
node-list	C	C (=)

#### Argument

L'argument achemine les paramètres spécifiques au service de la demande de service.

#### command-ID

Ce paramètre identifie la commande d'états de NMT à émettre. Une liste des commandes valides est représentée dans la CEI 61158-6-13.

#### node-list

Ce paramètre conditionnel spécifie les adresses des nœuds qui sont affectées par la commande. Il est utilisé seulement si l'adresse n'est ni "unicast" ni "broadcast".

### 6.1.1.3.6.3 Procédure du service

Le service est émis par le MN sur une demande intérieure ou sur une demande extérieure via le service NMT-req-invite. Les CN adressés modifient leur état à la réception de la commande d'états.

### 6.1.1.3.7 Service NMT-info

#### 6.1.1.3.7.1 Vue d'ensemble

Ce service est utilisé pour transmettre des informations de statut complexes sous forme de paquets ainsi que pour diffuser des informations d'installation relatives au système et issues du MN vers les CN.

#### 6.1.1.3.7.2 Primitives du service

Les paramètres de service pour ce service sont représentés dans le Tableau 7.

**Tableau 7 – Service NMT-info**

Nom du paramètre	Req	Ind
Argument		
AREP	M	M (=)
publish-node-list	C	C (=)
publish-time	C	C (=)

#### Argument

L'argument achemine les paramètres spécifiques au service de la demande de service.



**publish-node-list**

Ce paramètre spécifie une liste de nœuds qui sont dans le même état. Sinon, il fournit une liste de tous les nœuds avec leurs états respectifs indiqués. Les détails sont spécifiés dans la CEI 61158-6-13.

**publish-time**

Ce paramètre fournit le temps du système.

**6.1.1.3.7.3 Procédure du service**

Le service est émis par le MN sur une demande intérieure ou sur une demande extérieure via NMT-req-invite. Le service est soit "unicast", soit "broadcast".

**6.1.2 ASE d'objets de données du service****6.1.2.1 Vue d'ensemble**

Pour tous les types de transfert, c'est le client qui prend l'initiative d'un transfert. Le propriétaire du dictionnaire d'objets évalué est le serveur de l'objet de données du service (SDO). L'initiative d'annuler le transfert d'un SDO peut être prise soit par le client, soit par le serveur. Toutes les commandes sont confirmées. Le paramètre de résultat distant indique la réussite de la demande. En cas d'échec, il faut exécuter une demande d'abandon du transfert.

Il est possible de segmenter les données pour cause de leur taille supérieure à un seul élément transférable. La situation est gérée par l'ARPM QUB-COS (CMdL). Les détails sont présentés dans la CEI 61158-6-13.

**6.1.2.2 Spécification de la classe SDO****6.1.2.2.1 Modèle formel**

L'ASE SDO est spécifié par le modèle suivant:

<b>ASE:</b>	<b>ASE SDO</b>
<b>CLASS:</b>	<b>SDO</b>
<b>CLASS ID:</b>	<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>	<b>TOP</b>
<b>ATTRIBUTES:</b>	
1	(m) Attribut: index
2	(m) Attribut: sub-index
3	(o) Attribut: name
4	(m) Attribut: command-ID
5	(c) Attribut: segment-size
6	(c) Attribut: data-size
<b>SERVICES:</b>	
1	(m) OpsService: SDO-write
2	(o) OpsService: SDO-write-mult
3	(m) OpsService: SDO-read
4	(o) OpsService: SDO-read-mult
5	(m) OpsService: SDO-abort

**6.1.2.2.2 Attributs****index**

Cet attribut spécifie l'indice de l'entrée de l'OD à accéder.

**sub-index**

Cet attribut spécifie le sous-indice de l'entrée de l'OD à accéder.

**name**

Cet attribut spécifie le nom de l'entrée de l'OD à accéder.

**command-ID**

Cet attribut spécifie le type d'une opération de lecture / écriture.

**segment-size**

Ce paramètre conditionnel spécifie la longueur des données segmentées.

**data-size**

Cet attribut conditionnel spécifie la longueur du bloc de données transférées.

**6.1.2.2.3 Services****SDO-write**

Ce service est utilisé pour écrire des données dans une entrée du dictionnaire d'objets du serveur.

**SDO-write-mult**

Ce service est utilisé pour écrire des données dans des entrées multiples du dictionnaire d'objets du serveur.

**SDO-read**

Ce service est utilisé pour lire des données dans une entrée du dictionnaire d'objets du serveur.

**SDO-read-mult**

Ce service est utilisé pour lire des données dans des entrées multiples du dictionnaire d'objets du serveur.

**SDO-abort**

Le service est utilisé pour abandonner une communication SDO.

**6.1.2.3 Spécification de l'ASE SDO****6.1.2.3.1 Services pris en charge**

Le Paragraphe 6.1.2.3 contient la définition des services uniques à cet ASE. Les services définis pour cet ASE sont

- SDO-write
- SDO-write-mult
- SDO-read
- SDO-read-mult
- SDO-abort

**6.1.2.3.2 SDO-write****6.1.2.3.2.1 Vue d'ensemble**

A l'aide du service SDO-write, le client télécharge des données vers le serveur (propriétaire du dictionnaire d'objets) en affectant une entrée de son dictionnaire d'objets.

**6.1.2.3.2.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 8.

**Tableau 8 – SDO-write**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
segment-size	C	C (=)		
data-size	C	C (=)		
OD-identifiant	M	M (=)		
payload-data	M	M (=)		
Result(+)			S	S (=)
Result(-)			S	S (=)
error-info			M	M (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**command-ID**

Ce paramètre indique comment identifier une entrée de l'OD affectée. L'identification est effectuée soit par indice et sous-indice, soit uniquement par indice. Dans le dernier cas, le service adresse tous les sous-indices (sauf sous-indice 0) de l'indice indiqué. Les détails sont présentés dans la CEI 61158-6-13. La longueur des données utiles doit confirmer la longueur des données pour tous les sous-indices; tous les sous-indices doivent être enregistrables.

**segment-size**

Ce paramètre conditionnel est utilisé si les données sont à segmenter pour cause de leur taille supérieure à un seul élément transférable. Le paramètre indique la taille d'un segment.

**data-size**

Ce paramètre conditionnel spécifie le nombre d'octets dans le bloc transféré en totalité. Il est fourni dans le premier segment d'un transfert segmenté.

**OD-identifiant**

Ce paramètre identifie l'entrée du dictionnaire d'objets cibles à lire par ce service. Le type d'identification utilisé est spécifié par le paramètre "command-ID".

**payload-data**

Ce paramètre spécifie les données à télécharger.

**Result(+)**

Ce paramètre du type de sélection indique que la demande de service a réussi.

**Result(-)**

Ce paramètre du type de sélection indique que la demande de service a échoué.

**6.1.2.3.2.3 Procédure du service**

La procédure du service confirmé, spécifiée dans la CEI 61158-1, 9.6.1, s'applique à ce service.

**6.1.2.3.3 SDO-write-mult**

**6.1.2.3.3.1 Vue d'ensemble**

En utilisant le service SDO-write-mult, le client télécharge des données vers le serveur (propriétaire du dictionnaire d'objets), en affectant des entrées multiples de son dictionnaire d'objets.

**6.1.2.3.3.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 9.

**Tableau 9 – SDO-write-mult**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
segment-size	C	C (=)		
data-size	C	C (=)		
OD-identifiant (n)	M	M (=)		
payload-data (n)	M	M (=)		
Result(+)			S	S (=)
Result(-)			S	S (=)
error-info (n)			M	M (=)

NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**command-ID**

Ce paramètre indique comment identifier des entrées de l'OD affectées. Avec le service SDO-write-mult, le seul type d'identification autorisé est par indice et sous-indice. Les détails sont présentés dans la CEI 61158-6-13.

**segment-size**

Ce paramètre est utilisé si les données sont à segmenter pour cause de leur taille étant supérieure à un seul élément transférable. Le paramètre indique la taille d'un segment.

**data-size**

Ce paramètre conditionnel spécifie le nombre d'octets dans le bloc transféré en totalité. Il est fourni dans le premier segment d'un transfert segmenté.

**OD-identifiant (n)**

Ce paramètre indique les entrées n du dictionnaire d'objets cibles qui doivent être inscrites par ce service.

**payload-data (n)**

Ce paramètre spécifie les ensembles de données n à télécharger.

**Result(+)**

Ce paramètre du type de sélection indique que la demande de service a réussi.

**Result(-)**

Ce paramètre de type sélection indique que la demande de service pour une ou plusieurs entrées a échoué.

**error-info (n)**

Ce paramètre indique les entrées pour lesquelles l'écriture a échoué. Le code d'erreur respectif est donné.

**6.1.2.3.3 Procédure du service**

La procédure du service confirmé, spécifiée dans la CEI 61158-1, 9.6.1, s'applique à ce service.

**6.1.2.3.4 SDO-read****6.1.2.3.4.1 Vue d'ensemble**

A l'aide du service SDO-read, le client télécharge des données à partir du serveur (propriétaire du dictionnaire d'objets) en affectant une entrée de son dictionnaire d'objets.

**6.1.2.3.4.2 Primitives du service**

Les paramètres de service pour ce service sont représentés dans le Tableau 10.

**Tableau 10 – SDO-read**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
OD-identifiant	M	M (=)		
Result(+)			S	S (=)
segment-size			C	C (=)
data-size			C	C (=)
payload-data			M	M (=)
Result(-)			S	S (=)
error-info			M	M (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**command-ID**

Ce paramètre indique comment une entrée de l'OD affectée est à identifier. L'identification est effectuée soit par indice et sous-indice, soit uniquement par indice. Dans le dernier cas, le service adresse tous les sous-indices (sauf sous-indice 0) de l'indice indiqué. Les détails sont présentés dans la CEI 61158-6-13. La longueur des données utiles doit confirmer la longueur des données pour tous les sous-index; tous les sous-index doivent être lisibles.

**OD-identifiant**

Ce paramètre identifie l'entrée du dictionnaire d'objets cibles à lire par ce service. Le type d'identification utilisé est spécifié par le paramètre "command-ID".

**Result(+)**

Ce paramètre des types de sélection indique que la demande de service a réussi.

**segment-size**

Ce paramètre est utilisé si les données sont à segmenter pour cause de leur taille étant supérieure à un seul élément transférable. Le paramètre indique la taille d'un segment.

**data-size**

Ce paramètre conditionnel spécifie le nombre d'octets dans le bloc transféré en totalité. Il est fourni dans le premier segment d'un transfert segmenté.

**payload-data**

Ce paramètre spécifie les données téléchargées.

**Result(-)**

Ce paramètre des types de sélection indique que la demande de service a échoué.

**6.1.2.3.4.3 Procédure du service**

La procédure du service confirmé, spécifiée dans la CEI 61158-1, 9.6.1, s'applique à ce service.

**6.1.2.3.5 SDO-read-mult**

**6.1.2.3.5.1 Vue d'ensemble**

A l'aide du service SDO-read-mult, le client télécharge des données à partir du serveur (propriétaire du dictionnaire d'objets) en affectant plusieurs entrées de son dictionnaire d'objets.

**6.1.2.3.5.2 Primitives du service**

Les paramètres de service pour ce service sont représentés dans le Tableau 11.

**Tableau 11 – SDO-read-mult**

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
command-ID	M	M (=)		
OD-identifiant (n)	M	M (=)		
Result				
segment-size			C	C (=)
data-size			C	C (=)
payload-data / error-info (n)			M	M (=)
NOTE La méthode par laquelle une primitive "confirm" est corrélée à sa primitive "request" précédente correspondante relève d'une initiative locale. Voir 1.2.				

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**command-ID**

Ce paramètre indique comment des entrées de l'OD affectées sont à identifier. Avec le service SDO-read-mult, le seul type d'identification autorisé est par indice et sous-indice. Les détails sont présentés dans la CEI 61158-6-13.

**OD entry (n)**

Ce paramètre indique les entrées n du dictionnaire d'objets cibles qui doivent être lues par ce service.

**Result**

Ce paramètre du type de sélection indique le Résultat de la demande de service.

**segment-size**

Ce paramètre est utilisé si les données sont à segmenter pour cause de leur taille étant supérieure à un seul élément transférable. Le paramètre indique la taille d'un segment.

**data-size**

Ce paramètre conditionnel spécifie le nombre d'octets dans le bloc transféré en totalité. Il est fourni dans le premier segment d'un transfert segmenté.

**payload-data / error-info (n)**

Pour chacune des entrées du dictionnaire d'objets n, ce paramètre spécifie soit les données téléchargées avec succès, soit en cas d'échec, le code d'erreur respectif.

**6.1.2.3.5.3 Procédure du service**

La procédure du service confirmé spécifiée dans la CEI 61158-1, 9.6.1 s'applique à ce service.

**6.1.2.3.6 SDO-abort****6.1.2.3.6.1 Vue d'ensemble**

Ce service peut être exécuté à tout moment soit par le client, soit par le serveur d'un transfert SDO. Si le client d'un SDO a un service confirmé en cours, l'indication de l'annulation est acceptée à titre de confirmation de ce service-là.

**6.1.2.3.6.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 12.

**Tableau 12 – SDO-abort**

Nom du paramètre	Req	Ind
Argument		C
AREP	M	C (=)
error-info	M	C (=)

**6.1.2.3.6.3 Procédure du service**

Le service lui-même est non confirmé malgré la réponse partielle de la fin de la communication confirmée.

**6.1.3 ASE d'objets de données de traitement****6.1.3.1 Vue d'ensemble**

Le transfert de données en temps réel est effectué à l'aide des objets de données de traitement (PDO). La communication PDO dans un réseau de Type 13 est toujours effectuée de façon isochrone via les trames PReq et/ou PRes (voir les détails dans la CEI 61158-3-13 et la CEI 61158-4-13). Les trames PRes sont envoyées comme "broadcast" ou "multicast" selon le schéma éditeur/abonné. Les trames PReq avec les adresses "unicast" se conforment à la relation maître/esclave.

Le type de transmission de PDO est continu. Le type de transmission "on event" ou "on change" n'est pas fourni.

Du point de vue de l'appareil, il existe deux types d'utilisation de PDO: transmission de données et réception de données. On distingue des PDO de transmission (TPDO) et des PDO de réception (RPDO). Des appareils qui prennent en charge les TPDO sont des éditeurs PDO. Des appareils capables de recevoir des PDO sont des abonnés PDO.

Le nombre de canaux pris en charge est fourni par l'application. Le nombre de canaux PDO est spécifique à l'application.

### 6.1.3.2 Spécification de la classe PDO

#### 6.1.3.2.1 Modèle formel

<b>ASE:</b>		<b>ASE PDO</b>
<b>CLASS:</b>		<b>PDO</b>
<b>CLASS ID:</b>		<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>		<b>TOP</b>
<b>ATTRIBUTES:</b>		
1	(m)	Attribut: list-of-RPDO-channels
1.1	(m)	Attribut: node-ID
1.2	(m)	Attribut: mapping-version
1.3	(m)	Attribut: list-of-mapped-OD-entries
1.3.1	(m)	Attribut: index
1.3.2	(m)	Attribut: sub-index
1.3.3	(m)	Attribut: offset
1.3.4	(m)	Attribut: length
2	(m)	Attribut: List of TPDO channels
2.1	(m)	Attribut: node-ID
2.2	(m)	Attribut: mapping-version
2.3	(m)	Attribut: list-of-mapped-OD-entries
2.3.1	(m)	Attribut: index
2.3.2	(m)	Attribut: sub-index
2.3.3	(m)	Attribut: offset
2.3.4	(m)	Attribut: length
<b>SERVICES:</b>		
1	(m)	OpsService: PDO-transfer

#### 6.1.3.2.2 Attributs

##### list-of-RPDO-channels

Les attributs suivants spécifient les canaux logiques pour des PDO en arrivée. Chaque type de nœud peut contenir jusqu'à 254 canaux RPDO.

##### node-ID

Cet attribut spécifie l'ID du nœud expéditeur.

##### mapping-version

Cet attribut assure la compatibilité des mappings au PDO.

##### list-of-mapped-OD-entries

Les attributs suivants spécifient les entrées de l'OD dont le mapping au PDO du canal respectif doit être assuré. Un PDO est composé de 256 entrées OD avec le mapping assuré.

##### index

Cet attribut spécifie l'indice de l'entrée de l'OD dont le mapping doit être assuré.



**sub-index**

Cet attribut spécifie le sous-indice de l'entrée de l'OD dont le mapping doit être assuré.

**offset**

Cet attribut fournit le décalage relatif au départ de la charge utile PDO (Nombre de bits).

**length**

Cet attribut fournit la longueur de l'objet avec le mapping assuré (Nombre de bits).

**Liste des canaux TPDO**

Les attributs suivants spécifient les canaux logiques pour des PDO en partance. Un MN peut contenir jusqu'à 254 canaux TPDO, un CN peut en avoir un, un CN asynchrone n'en a aucun.

Tous les attributs suivants correspondent à leurs équivalents sous "list-of-RPDO-channels" et ne sont donc pas répétés ici.

**6.1.3.2.3 Services****PDO-transfer**

Ce service est utilisé pour transférer des données d'objets de processus entre les nœuds.

**6.1.3.3 Spécification du service PDO ASE****6.1.3.3.1 Services pris en charge**

Le Paragraphe 6.1.3.3 contient la définition des services uniques à cet ASE. Le service défini pour cet ASE est:

- PDO-transfer

**6.1.3.3.2 PDO-transfer****6.1.3.3.2.1 Vue d'ensemble**

Ce service est utilisé pour transférer des données d'objets de processus entre les nœuds. Sur le nœud demandeur, le PDO en partance est composé selon les attributs TPDO appropriés; sur le nœud destinataire, le PDO en arrivée est traité comme sujet aux attributs RPDO correspondants.

**6.1.3.3.2.2 Primitives du service**

Les paramètres de ce service sont représentés dans le Tableau 13.

**Tableau 13 – PDO-transfer**

Nom du paramètre	Req	Ind
Argument		
AREP	M	M (=)
PDO	M	M (=)
PDO-version	M	M (=)

**Argument**

L'argument achemine les paramètres spécifiques au service de la demande de service.

**PDO**

Ce paramètre spécifie les données PDO.

**PDO-version**

Ce paramètre spécifie la version du mapping PDO utilisé.

**6.1.3.3.2.3 Procédure du service**

Ce service est effectué soit comme un service non confirmé (du MN vers les CN), soit comme une séquence des deux services non confirmés consécutifs dans des directions opposées.

**6.1.4 ASE de relations entre applications**

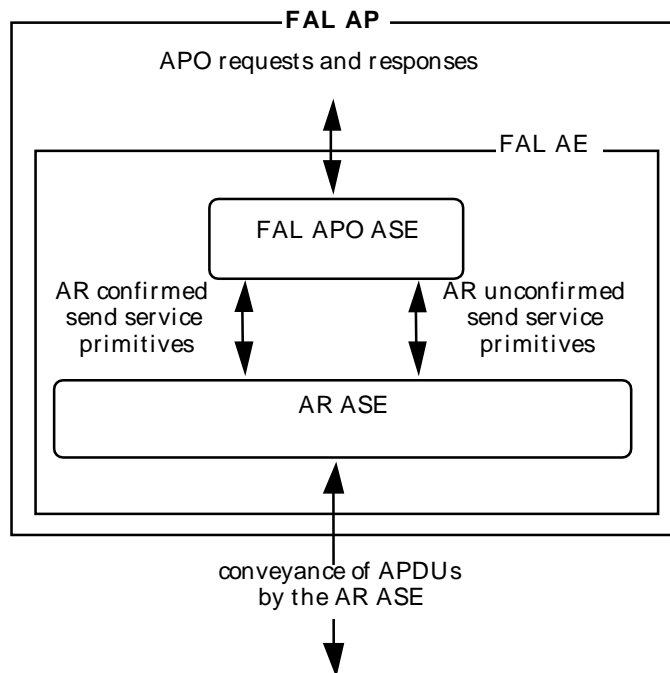
**6.1.4.1 Vue d'ensemble**

**6.1.4.1.1 Généralités**

Dans un système distribué, des processus d'application communiquent l'un avec l'autre en échangeant des messages de couche application à travers les canaux de communications de couche application bien définis. Ces canaux de communications sont modélisés dans la couche application de bus de terrain (FAL) comme des relations entre applications (AR).

Les AR sont responsables d'acheminer des messages entre les applications selon les caractéristiques de communications spécifiques exigées par les systèmes prioritaires. Des différentes combinaisons de ces caractéristiques conduisent à la définition des différents types d'AR. Les caractéristiques des AR sont définies formellement comme les attributs des classes de point d'extrémité d'AR.

Les messages qui sont acheminés par les AR sont des demandes et des réponses de service de FAL. Chacun d'entre eux est soumis à l'ASE d'AR pour le transfert par une ASE de FAL qui représente la classe de l'APO en cours d'évaluation. La Figure 2 représente ce concept.



**Légende**

Anglais	Français
APO requests and responses	Demandes et réponses d'APO
AR confirmed send service primitives	Primitives du service d'envoi confirmé d'AR
AR unconfirmed send service primitives	Primitives du service d'envoi non confirmé d'AR
Conveyance of APDUs by the AR ASE	Acheminement des APDU par l'ASE d'AR

**Figure 2 – ASE d'AR achemine des APDU entre les AP**

En fonction du type d'AR, des APDU peuvent être envoyées à un ou plusieurs processus d'application de destination connectés par l'AR. D'autres caractéristiques de l'AR déterminent comment des APDU sont à transférer. Ces caractéristiques sont décrites ci-dessous.

#### 6.1.4.1.2 Contexte du point d'extrémité

Chaque AP impliqué dans une AR contient un point d'extrémité d'AR. Chaque point d'extrémité d'AR est défini au sein d'une AE de l'AP. Étant combinée avec les définitions d'autres points d'extrémité, sa définition détermine une AR. Pour assurer la compatibilité de communications parmi ou entre des points d'extrémité, chaque définition du point d'extrémité contient un ensemble de caractéristiques relatives à la compatibilité. Ces caractéristiques ont besoin d'être configurées de façon appropriée pour chaque point d'extrémité afin que l'AR fonctionne correctement.

Les définitions du point d'extrémité contiennent également un ensemble de caractéristiques qui décrivent l'opération de l'AR. Ces caractéristiques, étant combinées avec celles utilisées pour spécifier la compatibilité, définissent le contexte du point d'extrémité. Le contexte du point d'extrémité est utilisé par l'ASE d'AR pour gérer l'opération du point d'extrémité et l'acheminement des APDU. Les caractéristiques, qui comprennent le contexte du point d'extrémité, sont décrites ultérieurement.

##### 6.1.4.1.2.1 Rôle du point d'extrémité

Le rôle d'un AREP détermine le comportement autorisé d'un AP sur l'AREP. Le rôle d'AREP peut être celui d'un client, serveur, éditeur par émission, abonné par émission, maître ou esclave.

Le Tableau 14 et le Tableau 15 résument les caractéristiques et les combinaisons de chacun des rôles d'AREP.

**Tableau 14 – Acheminement des primitives de service par le rôle d'AREP**

	Client	Serveur	Éditeur par émission	Abonné par émission	Maître	Esclave
Send Service Req	X		X		X	
Recv Service Req		X		X		X
Send Service Rsp		X				
Recv Service Rsp	X					

**Tableau 15 – Combinaisons valides des rôles d'AREP impliqués dans une AR**

Rôles d'AREP par le Modèle d'Interaction	Client	Serveur	Éditeur par émission	Abonné par émission	Maître	Esclave
Client/Serveur						
Client		Oui				
Serveur						
Éditeur/abonné						
Éditeur par émission				Oui		
Abonné par émission						
Maître						Oui
Esclave						

#### **6.1.4.1.2.2 Cardinalité**

La cardinalité d'une AR spécifie, du point de vue d'un client ou d'un point d'extrémité éditeur, combien de processus d'application distants sont impliqués dans une AR. Une cardinalité n'est jamais exprimée du point de vue d'un serveur ou d'un abonné.

Lorsqu'elle est exprimée du point de vue d'un client ou d'un point d'extrémité homologue, des AR sont toujours de type 1:1. Les clients ne sont jamais capables d'émettre une demande et d'attendre des réponses en provenance des serveurs multiples.

Lorsque la cardinalité est exprimée du point de vue d'un point d'extrémité abonné, les AR indiquent que des abonnés multiples sont pris en charge. Ces AR sont de type 1:n. Les AR qui sont de type 1:n assurent la communication entre une application ou un groupe d'applications (une ou plusieurs). Elles sont souvent dénommées "multicast" (multidiffusion).

#### **6.1.4.1.3 Modèle d'acheminement**

##### **6.1.4.1.3.1 Généralités**

Le modèle d'acheminement définit comment des APDU sont envoyées entre les points d'extrémité d'une AR. Pour définir ces transferts, on utilise trois caractéristiques:

- trajets d'acheminement,
- politique de déclenchement, et
- politique d'acheminement.

##### **6.1.4.1.3.2 Trajets d'acheminement**

L'objectif des ASE d'AR est de transférer des informations entre les points d'extrémité d'AR. Ce transfert d'informations a lieu sur les trajets d'acheminement d'une AR. Un trajet d'acheminement représente un trajet de communication à sens unique utilisé par un point d'extrémité pour l'entrée et la sortie.

Pour maintenir le rôle du processus d'application, le point d'extrémité est configuré avec un ou deux trajets d'acheminement. Des points d'extrémité qui ne font qu'envoyer ou recevoir sont configurés soit avec un trajet d'acheminement d'envoi, soit avec un trajet d'acheminement de réception; ceux qui font les deux sont configurés avec les deux. Des AR avec un seul trajet d'acheminement sont appelées unidirectionnelles, et celles avec deux trajets d'acheminement sont bidirectionnelles.

Des AR unidirectionnelles sont capables d'acheminer seulement des demandes de service. Une AR bidirectionnelle est nécessaire pour acheminer des réponses de service. C'est pourquoi les AR unidirectionnelles prennent en charge le transfert des services non confirmés seulement dans une direction, alors que les AR bidirectionnelles prennent en charge le transfert des services non confirmés et confirmés initiés seulement par un point d'extrémité, ou par les deux points d'extrémité.

##### **6.1.4.1.3.3 Politique de déclenchement**

La politique de déclenchement indique quand des APDU sont transmises par la couche liaison de données sur le réseau.

Le premier type est dénommé "user-triggered" (déclenché par un utilisateur). Des APDU déclenchées par l'utilisateur soumettent des APDU de FAL à la couche liaison de données pour la transmission à la prochaine occasion.

Le deuxième type est dénommé "network-scheduled" (programmé par un réseau). Des APDU programmées par le réseau soumettent des APDU de FAL à la couche liaison de données pour une transmission selon un programme configuré par la gestion.

Ce mécanisme de programmation réseau est cyclique.

#### **6.1.4.1.3.4 Politique d'acheminement**

La politique d'acheminement indique si les APDU sont transférées selon un modèle de tampon ou un modèle de file d'attente. Ces modèles décrivent la méthode d'acheminement des APDU d'un expéditeur vers un destinataire.

Les AR tamponnées contiennent des trajets d'acheminement qui possèdent un seul tampon sur chaque point d'extrémité. Des mises à jour du tampon de source sont acheminées vers le tampon destinataire selon la politique de déclenchement de l'AR. Des mises à jour de n'importe quel autre tampon remplacent son contenu par de nouvelles données. Dans les AR tamponnées, des données non acheminées ou non livrées, ayant été remplacées, sont perdues. En plus, des données contenues dans un tampon peuvent être lues à plusieurs reprises sans que leur contenu soit détruit.

Les AR en file d'attente contiennent des trajets d'acheminement qui sont représentés comme une file d'attente entre les points d'extrémité. Les AR en file d'attente acheminent des données à l'aide d'une file d'attente FIFO. Les AR ne sont pas écrasées, les nouvelles entrées sont mises en file d'attente jusqu'à ce qu'elles soient acheminées et livrées.

Si une file d'attente est pleine, un nouveau message ne sera pas mis en file d'attente.

**NOTE** Les services d'acheminement d'AR sont décrits de façon abstraite d'une telle manière qu'ils puissent être mis en œuvre et fonctionner à l'aide des tampons ou des files d'attente. Ces services peuvent être mis en œuvre de plusieurs manières. Par exemple, leur mise en œuvre peut être effectuée d'une manière d'assurer la capacité de charger le tampon/la file d'attente et de le/la poster, par la suite, pour un transfert par la couche liaison de données sous-jacente. Sinon, ces services peuvent être mis en œuvre d'une manière de combiner ces capacités pour que le tampon/la file d'attente puisse être chargé(e) et transféré(e) dans une seule demande. Du côté récepteur, ces services peuvent être mis en œuvre en livrant les données après leur réception, ou en indiquant leur réception et en permettant à l'utilisateur de les récupérer dans une opération séparée. Une autre option consiste à demander à l'utilisateur de détecter que le tampon ou la file d'attente est mis(e) à jour.

#### **6.1.4.1.4 Services de communications sous-jacents**

##### **6.1.4.1.4.1 Généralités**

Les ASE d'AR acheminent les APDU de FAL en utilisant les capacités de la couche liaison de données sous-jacente. Plusieurs caractéristiques sont utilisées pour décrire ces capacités. Le Paragraphe 6.1.4.1.4 présente la description pour chaque caractéristique. Ces caractéristiques sont spécifiques au mapping de liaison de données et sont définies dans la CEI 61158-6-13. Leur spécification précise peut y être consultée.

##### **6.1.4.1.4.2 Services orientés connexion**

La couche sous-jacente ne prend pas en charge des points d'extrémité d'AR en fournissant les services orientés connexion. Ainsi, les connexions doivent être établies à l'aide de la Couche Application.

##### **6.1.4.1.4.3 Services tamponnés et mis en file d'attente**

La couche sous-jacente peut prendre en charge des points d'extrémité d'AR en fournissant les services tamponnés ou mis en attente. Ces services peuvent être utilisés pour mettre en œuvre des tampons ou des files d'attente exigés par certaines classes de points d'extrémité.

##### **6.1.4.1.4.4 Transferts cycliques et acycliques**

La couche sous-jacente peut prendre en charge des points d'extrémité d'AR en fournissant des services cycliques et acycliques.

#### **6.1.4.1.4.5 Segmentation**

La FAL est capable de segmenter des données d'utilisateur qu'elle achemine.

#### **6.1.4.1.5 Établissement d'AR**

Pour qu'un point d'extrémité d'AR soit utilisé par un processus d'application, l'AR correspondante nécessite d'être active. Lorsqu'une AR est activée, elle est dite "established" (établie).

L'établissement d'AR peut avoir lieu de l'une de deux manières existantes. Dans la première, des AR peuvent être préétablies. "Pre-established" (préétablie) signifie que l'AE, qui prend en charge le contexte du point d'extrémité, est créée lorsque l'AP est connecté au réseau. Dans ce cas, les communications entre les applications impliquées dans l'AR peuvent avoir lieu sans devoir d'abord établir explicitement l'AR.

Dans la seconde façon, les AR peuvent nécessiter une définition et un établissement dynamiques. Dans ce cas, les définitions sont à créer pour chaque AREP à l'aide de l'ARPM QUB-COS (CmdL). Les détails sont présentés dans la CEI 61158-6-13.

#### **6.1.4.1.6 Classes des relations entre applications**

Les AREP sont définis avec une combinaison des caractéristiques pour former des différentes classes d'AR.

### **6.1.4.2 Spécifications des classes du point d'extrémité des relations entre applications**

#### **6.1.4.2.1 Modèle formel**

Le modèle formel du point d'extrémité d'AR définit les fonctions typiques pour tous les points d'extrémité d'AR. Cette classe ne peut pas être instanciée. Elle n'est présente que pour l'héritage de ses attributs et services par ses sous-classes, comme spécifié dans un paragraphe séparé de la présente norme.

**FAL ASE:** ASE AR  
**CLASS:** POINT D'EXTRÉMITÉ D'AR  
**CLASS ID:** n'est pas utilisé  
**PARENT CLASS:** TOP

#### ATTRIBUTS DE GESTION DES SYSTÈMES

1	(m)	Attribut:	Local AP
2	(o)	Attribut:	List of supported attributes

#### SERVICES:

1	(o)	OpsService:	AR-Ident
2	(o)	OpsService:	AR-Status
3	(o)	OpsService:	AR-NMT-req-invite
4	(o)	OpsService:	AR-NMT-state-command
5	(o)	OpsService:	AR-NMT-info
6	(o)	OpsService:	AR-SDO-write
7	(o)	OpsService:	AR-SDO-write-mult
8	(o)	OpsService:	AR-SDO-read
9	(o)	OpsService:	AR-SDO-read-mult
10	(o)	OpsService:	AR-SDO-abort
11	(m)	OpsService:	AR-PDO-transfer

#### 6.1.4.2.2 Attributs de gestion des systèmes

##### Local AP

Cet attribut définit l'AP attaché ou configuré pour l'utilisation de l'AREP à l'aide d'une référence locale.

##### List of supported attributes

Cet attribut facultatif spécifie tous les attributs pris en charge par l'objet. Cette liste contient, au minimum, les attributs obligatoires pour la classe de l'objet.

#### 6.1.4.2.3 Services

Les fonctionnalités de tous les services représentés ci-dessus correspondent aux services ayant le même nom (sans préfixe "AR") énumérés de 6.1.1 à 6.1.3. Ces services sont répétés pour des raisons formelles. Ils sont référencés sans préfixe ajouté dans le contexte ultérieur de la CEI 61158-5-13 et de la CEI 61158-6-13.

#### 6.1.4.3 Spécifications des services d'ASE des relations entre applications

##### 6.1.4.3.1 Services pris en charge

Voir 6.1.1 à 6.1.3.

## 6.2 AR

### 6.2.1 Spécification de la classe de point d'extrémité d'AR BNB-PEC

#### 6.2.1.1 Vue d'ensemble de la classe

Cette classe est définie pour prendre en charge l'échange cyclique des services non confirmés entre les processus d'application en utilisant des tampons. Le cycle est donné par le MN.

Les services fournis par cet AREP permettent une communication directe entre les CN, le so-disant trafic croisé.

Le comportement de ce type d'AR peut être décrit comme suit.

Un utilisateur d'ASE d'AR souhaitant acheminer une demande d'APDU soumet cette demande comme une unité de données du service d'ASE d'AR à son AREP pour la diffusion. Cette activité est toujours localisée sur le MN. Il inscrit des APDU dans le tampon intérieur, en remplaçant complètement le contenu existant du tampon. La couche liaison de données transfère le contenu du tampon au prochain transfert programmé.

Si l'AREP reçoit une autre APDU avant que le contenu du tampon soit transmis, le contenu du tampon sera remplacé par la nouvelle APDU et l'APDU précédente sera perdue. Lorsque le contenu du tampon est transmis, l'ASE d'AR alerte l'utilisateur de la transmission.

Le point d'extrémité destinataire est toujours localisé sur un CN. Ici, l'APDU est reçue à partir du réseau et est immédiatement inscrite dans le tampon, en écrasant complètement le contenu existant du tampon. Le point d'extrémité alerte l'utilisateur de l'arrivée de l'APDU et de sa livraison à l'utilisateur selon l'interface d'utilisateur local.

La réponse suivante issue de l'utilisateur d'ASE d'AR distant ne fait pas l'objet d'une confirmation ou d'un acquittement d'une APDU reçue, mais contient des données indépendantes de lui. Elle est inscrite par l'AREP dans le tampon de couche liaison de données, en remplaçant complètement le contenu existant du tampon. La couche liaison de données transfère le contenu du tampon le plus vite possible. Le délai d'attente entre une indication de données en arrivée et une réponse suivante ne dépend que des capacités du nœud hôte de l'AREP.

Sur le point d'extrémité en demande (de nouveau sur le MN), l'APDU contenant la réponse issue du réseau, est inscrite sur le tampon en remplaçant complètement le contenu existant du tampon. Le point d'extrémité alerte l'utilisateur de l'arrivée de l'APDU et de sa livraison à l'utilisateur selon l'interface d'utilisateur local à titre de confirmation.

En outre, tout CN peut recevoir et interpréter cette réponse en fonction de sa configuration.

Les caractéristiques de cette classe d'AREP sont résumées ci-dessous.

Rôle	Éditeur par émission / Abonné par émission
Cardinalité	1:1 (pour la demande) 1:n (pour la réponse)

### 6.2.1.2 Modèle formel

<b>ASE:</b>	<b>AR</b>
<b>CLASS:</b>	<b>Buffered network-scheduled bi-directional pre-established connection AR endpoint</b>
<b>CLASS ID:</b>	<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>ATTRIBUTES:</b>	
1 (m) Attribut:	Role
2 (m) Attribut:	AREP-state
3 (m) Attribut:	DL-mapping-reference
<b>SERVICES:</b>	
1 (m) OpsService:	PDO-transfer

### 6.2.1.3 Attributs de gestion du réseau

#### Role

Cet attribut spécifie le rôle de l'AREP. Les valeurs valides sont:



- **Push-publisher:** Les points d'extrémité de ce type éditent leurs données en émettant des APDU de demande de service non confirmé.
- **Push-subscriber:** Les points d'extrémité de ce type reçoivent des données issues des APDU de demande de service par un AREP éditeur par émission.

#### **AREP-state**

Cet attribut spécifie l'état de l'AREP. Les valeurs pour cet attribut sont spécifiées dans la CEI 61158-6-13.

#### **DL-mapping-reference**

Pour les AREP éditeurs, cet attribut spécifie le mapping vers le trajet d'acheminement de la transmission. Pour les AREP abonnés, cet attribut spécifie le mapping vers le trajet d'acheminement de la réception. Les attributs du mapping de DL pour la couche liaison de données sont spécifiés dans la CEI 61158-6-13.

### **6.2.1.4 Services**

#### **PDO-transfer**

Ce service est utilisé pour transférer des données d'objets de processus entre les nœuds.

### **6.2.2 Spécification de la classe de point d'extrémité d'AR BNU-PEC**

#### **6.2.2.1 Vue d'ensemble de la classe**

Cette classe est définie pour prendre en charge le modèle "push" pour la diffusion tamponnée et programmée des services non confirmés vers un ou plusieurs processus d'application.

Le comportement de ce type d'AR peut être décrit comme suit.

Un utilisateur d'ASE d'AR, qui souhaite acheminer une APDU de demande ou de réponse, la soumet comme une unité de données du service d'ASE d'AR vers son AREP pour la diffusion. Cette activité est toujours localisée sur le MN. L'AREP, qui envoie une APDU de demande ou de réponse, l'inscrit dans le tampon de couche liaison de données en remplaçant complètement le contenu existant du tampon. La couche liaison de données transfère le contenu du tampon au prochain transfert programmé.

Si l'AREP ayant envoyé l'APDU reçoit une autre APDU avant que le contenu du tampon soit transmis, le contenu du tampon sera remplacé par la nouvelle APDU et l'APDU précédente sera perdue. Lorsque le contenu du tampon est transmis, l'ASE d'AR avertit l'utilisateur de la transmission.

Un ou plusieurs CN sont des points d'extrémité destinataires. Ici, l'APDU est reçue à partir du réseau et est immédiatement inscrite dans le tampon, en écrasant complètement le contenu existant du tampon. Le point d'extrémité alerte l'utilisateur de l'arrivée de l'APDU et de sa livraison à l'utilisateur selon l'interface d'utilisateur local. Si l'APDU n'est pas livrée avant que la suivante APDU arrive, elle sera écrasée par l'APDU suivante avant d'être perdue.

Les caractéristiques de cette classe d'AREP sont résumées ci-dessous.

Rôles	Éditeur par émission Abonné par émission
Cardinalité	1:n

### 6.2.2.2 Modèle formel

**ASE:** AR

**CLASS:** Buffered network-scheduled uni-directional pre-established connection AR endpoint

**CLASS ID:** n'est pas utilisé

**PARENT CLASS:** AREP

**ATTRIBUTES:**

1	(m)	Attribut:	Role
2	(m)	Attribut:	AREP-state
3	(m)	Attribut:	DL-mapping-reference

**SERVICES:**

1	(m)	OpsService:	PDO transfer
---	-----	-------------	--------------

### 6.2.2.3 Attributs de gestion du réseau

#### Role

Cet attribut spécifie le rôle de l'AREP. Les valeurs valides sont:

- Push-publisher: Les points d'extrémité de ce type éditent leurs données en émettant des APDU de demande de service non confirmé.
- Push-subscriber: Les points d'extrémité de ce type reçoivent des données issues des APDU de demande de service par un AREP éditeur par émission.

#### AREP-state

Cet attribut spécifie l'état de l'AREP. Les valeurs pour cet attribut sont spécifiées dans la CEI 61158-6-13.

#### DL-mapping-reference

Pour les AREP éditeurs, cet attribut spécifie le mapping vers le trajet d'acheminement de la transmission. Pour les AREP abonnés, cet attribut spécifie le mapping vers le trajet d'acheminement de la réception. Les attributs du mapping de DL pour la couche liaison de données sont spécifiés dans la CEI 61158-6-13.

### 6.2.2.4 Services

#### PDO-transfer

Ce service est utilisé pour transférer des données d'objets de processus entre les nœuds.

## 6.2.3 Spécification de la classe de point d'extrémité d'AR QUU

### 6.2.3.1 Vue d'ensemble de la classe

Cette classe est définie pour prendre en charge la diffusion mise en file d'attente sur demande des services non confirmés vers un ou plusieurs processus d'application. Le comportement de ce type d'AR peut être décrit comme suit.

Un utilisateur d'ASE d'AR, qui souhaite acheminer une APDU de demande, soumet une unité de données du service d'ASE d'AR vers le point d'extrémité expéditeur de l'AR. L'AREP, qui envoie l'APDU de demande, la met en file d'attente sur sa couche sous-jacente pour le transfert. La couche sous-jacente l'envoie à la prochaine occasion. L'AREP, qui reçoit une APDU de demande issue de sa couche sous-jacente, la livre à son utilisateur d'ASE d'AR dans le même ordre qu'elle a été reçue.

Les caractéristiques de cette classe d'AREP sont résumées ci-dessous.

Rôles	Maître
	Esclave
Cardinalité	1:n

### 6.2.3.2 Modèle formel

<b>FAL ASE:</b>	<b>ASE AR</b>
<b>CLASS:</b>	<b>Queued user-triggered uni-directional AR endpoint</b>
<b>CLASS ID:</b>	<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>ATTRIBUTS DE GESTION DU RÉSEAU:</b>	
1	(m) Attribut: Role
2	(m) Attribut: AREP State
3	(m) Attribut: DL-mapping-reference
<b>SERVICES:</b>	
1	(o) OpsService: NMT-state-command
2	(o) OpsService: NMT-info

### 6.2.3.3 Attributs de gestion du réseau

#### Role

Cet attribut spécifie le rôle de l'AREP. Les valeurs valides sont:

- Master: Les points d'extrémité de ce type libèrent leurs données en émettant des APDU de demande de service non confirmé.
- Slave: Les points d'extrémité de ce type reçoivent leurs données en provenance des APDU de demande de service émises par un AREP maître.

#### AREP-state

Cet attribut spécifie l'état de l'AREP. Les valeurs pour cet attribut sont spécifiées dans la CEI 61158-6-13.

#### DL-mapping-reference

Cet attribut fournit une référence au mapping de couche liaison de données pour le trajet d'acheminement pour cet AREP. Les attributs du mapping de DL pour la couche liaison de données sont spécifiés dans la CEI 61158-6-13.

### 6.2.3.4 Services

#### NMT-state-command

Le MN utilise des services NMT-state-command pour contrôler le(s) diagramme(s) d'états du (des) CN(s).

#### NMT-info

Ce service est utilisé pour transmettre des informations complexes du MN vers les CN.

## 6.2.4 Spécification de la classe de point d'extrémité d'AR QUB-CL

### 6.2.4.1 Vue d'ensemble de la classe

Cette classe est définie pour prendre en charge l'échange sur demande des services non confirmés entre deux ou plusieurs processus d'application. Cette classe utilise des services de liaison de données sans connexion pour les échanges. Le comportement de cette classe est décrit comme suit.

Un utilisateur d'ASE d'AR, qui souhaite acheminer une APDU de demande, la soumet comme une unité de données du service d'ASE d'AR vers son AREP. L'AREP, qui envoie l'APDU de demande, la met en file d'attente sur sa couche sous-jacente pour le transfert à la prochaine occasion.

L'AREP, qui reçoit une APDU de demande issue de la couche sous-jacente, la met en file d'attente pour la livraison vers son utilisateur d'ASE d'AR dans le même ordre qu'elle a été reçue.

Pour une demande de service confirmé, l'AREP qui reçoit l'APDU de demande accepte l'APDU de réponse correspondante issue de son utilisateur d'ASE d'AR et la met en file d'attente sur sa couche sous-jacente pour le transfert.

L'AREP, qui a émis une APDU de demande, reçoit l'APDU de réponse issue de sa couche sous-jacente et la met en file d'attente pour la livraison à son utilisateur d'ASE d'AR dans le même ordre qu'elle a été reçue.

Les caractéristiques de cette classe d'AREP sont résumées ci-dessous.

Rôles	Client
	Serveur
Cardinalité	1:1

#### 6.2.4.2 Modèle formel

<b>FAL ASE:</b>	<b>ASE AR</b>
<b>CLASS:</b>	<b>Queued user-triggered bi-directional connectionless AR endpoint</b>
<b>CLASS ID:</b>	<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>	<b>Point d'extrémité d'AR</b>
<b>ATTRIBUTS DE GESTION DU RÉSEAU:</b>	
1	(m) Attribut: Role
2	(m) Attribut: AREP State
3	(m) Attribut: DL-mapping-reference
<b>SERVICES:</b>	
1	(o) OpsService: Ident
2	(o) OpsService: Status
3	(o) OpsService: Sync
4	(o) OpsService: NMT-req-invite

#### 6.2.4.3 Attributs de gestion du réseau

##### Role

Cet attribut spécifie le rôle de l'AREP. Les valeurs valides sont:

- Client: Les points d'extrémité de ce type émettent des APDU de demande de service non confirmé vers les serveurs et reçoivent des APDU de réponse de service non confirmé.
- Server: Les points d'extrémité de ce type reçoivent des APDU de demande de service non confirmé issues des clients et émettent des APDU de réponse de service non confirmé vers les clients.

##### AREP-state

Cet attribut spécifie l'état de l'AREP. Les valeurs pour cet attribut sont spécifiées dans la CEI 61158-6-13.

##### DL-mapping-reference

Cet attribut fournit une référence au mapping de couche liaison de données pour le trajet d'acheminement pour cet AREP. Les attributs du mapping de DL pour la couche liaison de données sont spécifiés dans la CEI 61158-6-13.

#### 6.2.4.4 Services

##### Ident

Le service Ident est utilisé par le MN pour identifier les CN.

##### Status

Le service Status est utilisé par le MN pour demander des informations de statut issues du CN.

**Sync**

Le service Sync est utilisé par le MN pour configurer/synchroniser un CN avec une PRes en temps déclenché.

**NMT-req-invite**

Le MN utilise ce service pour savoir quel service suivant, souhaité par le CN, doit être effectué par le MN.

**6.2.5 Spécification de la classe de point d'extrémité d'AR QUB-COS****6.2.5.1 Vue d'ensemble de la classe**

Cette classe est définie pour prendre en charge l'échange sur demande des services confirmés entre deux ou plusieurs processus d'application. Des services non confirmés ne sont pas pris en charge par ce type d'AR. Pour le traitement des connexions, il utilise les services relatifs à la connexion fournis par l'ASE d'AR. Elle prend en charge la segmentation des APDU au sein de la Couche Application. La priorité de la liaison de données pour les transferts est spécifiée séparément pour chaque transfert.

Le comportement de cette classe est décrit comme suit.

Un utilisateur d'ASE d'AR, qui souhaite acheminer une APDU de demande, la soumet comme une unité de données du service d'ASE d'AR vers son AREP. L'AREP, qui envoie l'APDU de demande, la met en file d'attente sur sa couche sous-jacente pour le transfert à la prochaine occasion.

L'AREP, qui reçoit une APDU de demande issue de la couche sous-jacente, la met en file d'attente pour la livraison vers son utilisateur d'ASE d'AR dans le même ordre qu'elle a été reçue.

Pour une demande de service confirmé, l'AREP qui reçoit l'APDU de demande accepte l'APDU de réponse correspondante issue de son utilisateur d'ASE d'AR et la met en file d'attente sur sa couche sous-jacente pour le transfert.

L'AREP, qui a émis une APDU de demande, reçoit l'APDU de réponse issue de sa couche sous-jacente et la met en file d'attente pour la livraison à son utilisateur d'ASE d'AR dans le même ordre qu'elle a été reçue.

Pour les deux directions, une unité de données du service d'ASE d'AR peut être divisée en plusieurs APDU, si sa taille est supérieure à son seul élément transférable. Les paramètres pour le traitement de la segmentation sont fournis par l'ASE de FAL qui souhaite envoyer les données affectées. La couche sous-jacente envoie chaque segment à la prochaine occasion. L'AREP, qui reçoit les segments d'APDU issus de sa couche sous-jacente, rassemble l'APDU et la livre à l'ASE d'AR.

Les caractéristiques de cette classe d'AREP sont résumées ci-dessous.

Rôles	Client Serveur
Cardinalité	1:1

**6.2.5.2 Modèle formel**

<b>ASE:</b>	<b>AR</b>
<b>CLASS:</b>	<b>Queued user-triggered bi-directional connection-oriented with segmentation AR endpoint</b>
<b>CLASS ID:</b>	<b>n'est pas utilisé</b>
<b>PARENT CLASS:</b>	<b>AREP</b>
<b>ATTRIBUTES:</b>	

1	(m)	Attribut:	Role
2	(m)	Attribut:	AREP-state
3	(m)	Attribut:	DL-mapping-reference

**SERVICES:**

1	(m)	OpsService:	SDO-write
2	(o)	OpsService:	SDO-write-mult
3	(m)	OpsService:	SDO-read
4	(o)	OpsService:	SDO-read-mult
5	(m)	OpsService:	SDO-abort

**6.2.5.3 Attributs de gestion du réseau****Role**

Cet attribut spécifie le rôle de l'AREP. Les valeurs valides sont:

- Client: Les points d'extrémité de ce type émettent des APDU de demande de service confirmé à des serveurs et reçoivent des APDU de réponse de service confirmé.
- Server: Les points d'extrémité de ce type reçoivent des APDU de demande de service confirmé issues des clients et émettent des APDU de réponse de service confirmé.

**AREP-state**

Cet attribut spécifie l'état de l'AREP. Les valeurs pour cet attribut sont spécifiées dans la CEI 61158-6-13.

**DL-mapping-reference**

Cet attribut fournit une référence au mapping de couche liaison de données pour le trajet d'acheminement pour cet AREP. Les attributs du mapping de DL pour la couche liaison de données sont spécifiés dans la CEI 61158-6-13.

**6.2.5.4 Services****SDO-write**

Ce service est utilisé pour inscrire des données sur une entrée du dictionnaire d'objets du serveur.

**SDO-write-mult**

Ce service est utilisé pour inscrire des données sur des entrées multiples du dictionnaire d'objets du serveur.

**SDO-read**

Ce service est utilisé pour la lecture des données à partir d'une entrée du dictionnaire d'objets du serveur.

**SDO-read-mult**

Ce service est utilisé pour la lecture des données à partir des entrées multiples du dictionnaire d'objets du serveur.

**SDO-abort**

Le service est utilisé pour annuler une communication SDO.

**6.3 Résumé sur les classes de FAL**

Le Paragraphe 6.3 contient un résumé sur les classes définies de FAL. Les valeurs Class ID sont attribuées pour être compatibles avec des standards existants. Le Tableau 16 représente un résumé sur les classes.

**Tableau 16 – Résumé sur les classes de FAL**

FAL ASE	Classe
	AP
Data type	Fixed length & String data type
Application Relationship	AREP
	BNU-PEC
	BNB-PEC
	QUU
	QUB-CL
	QUB-COS
Service data object	SDO
Process data object	PDO

#### 6.4 Services de FAL autorisés par le rôle d'AREP

Le Tableau 17 ci-dessous définit les combinaisons valides des services et des rôles d'AREP (quels AREP et APDU de service avec le rôle spécifié peuvent envoyer ou recevoir). Les colonnes Unc et Cnf indiquent si le service représenté dans la colonne à gauche est non confirmé (Unc) ou confirmé (Cnf).

**Tableau 17 – Services par le rôle d'AREP**

	Non confirmé	Confirmé	Client	Serveur	Éditeur par émission	Abonné par émission	Maître	Esclave
Services FAL			req-rcv	req-rcv	req-rcv	req-rcv	req-rcv	req-rcv
AP ASE								
Ident	x		x x	x x				
Status	x		x x	x x				
Sync	x						x x	x x
NMT-req-invite	x		x x	x x				
NMT-state-command	x						x	x
NMT-info	x						x	x
SDO ASE								
SDO-write		x	x x	x x				
SDO-write-mult		x	x x	x x				
SDO-read		x	x x	x x				
SDO-read-mult		x	x x	x x				
PDO ASE								
PDO-transfer	x				x		x	
AR ASE								
(Voir les services équivalents ci-dessus)								



## Bibliographie

CEI 61784-1, *Réseaux de communication industriels – Profils – Partie 1: Profils de bus de terrain*

CEI 61784-2, *Réseaux de communication industriels – Profils – Partie 2: Profils de bus de terrain supplémentaires pour les réseaux en temps réel basés sur l'ISO/CEI 8802-3*

EPSS DS 301 V1.2.0, *Ethernet POWERLINK Communication Profile Specification, Draft Standard Version 1.2.0, EPSS 2013, disponible à l'adresse <http://www.ethernet-powerlink.org/>*

---





INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

3, rue de Varembé  
PO Box 131  
CH-1211 Geneva 20  
Switzerland

Tel: + 41 22 919 02 11  
Fax: + 41 22 919 03 00  
[info@iec.ch](mailto:info@iec.ch)  
[www.iec.ch](http://www.iec.ch)