

INTERNATIONAL STANDARD

IEC
61158-5

Second edition
2000-01

**Digital data communications for
measurement and control –
Fieldbus for use in industrial control systems –**

**Part 5:
Application Layer Service definition**



Reference number
IEC 61158-5:2000(E)

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Validity of this publication

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

Information relating to the date of the reconfirmation of the publication is available in the IEC catalogue.

Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is to be found at the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
Published yearly with regular updates
(On-line catalogue)*
- **IEC Bulletin**
Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary* (IEV).

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment*. *Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams*.

* See web site address on title page.

INTERNATIONAL STANDARD

IEC
61158-5

Second edition
2000-01

Digital data communications for measurement and control – Fieldbus for use in industrial control systems –

Part 5: Application Layer Service definition

© IEC 2000 – Copyright – all rights reserved

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

International Electrotechnical Commission
Telefax: +41 22 919 0300

3, rue de Varembé Geneva, Switzerland
e-mail: inmail@iec.ch IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE **XH**

For price, see current catalogue

CONTENTS

Page

FOREWORD	9
INTRODUCTION	11
Clause	
1 SCOPE	11
2 NORMATIVE REFERENCES	12
3 DEFINITIONS	13
3.1 ISO/IEC 7498-1 terms	13
3.2 ISO/IEC 8822 terms	13
3.3 ISO/IEC 9545 terms	13
3.4 ISO/IEC 8824 terms	13
3.5 Fieldbus Data Link Layer terms	13
3.6 FAL Specific Definitions	14
3.6.1 Access Protection Definition	14
3.6.2 Address Assignment Table Definition	14
3.6.3 Allocate Definition	14
3.6.4 Application Layer Interoperability Definition	14
3.6.5 Application Object Definition	14
3.6.6 Application Process Definition	14
3.6.7 Application Process Identifier Definition	14
3.6.8 Application Process Object Class Definition	15
3.6.9 Application Relationship Definition	15
3.6.10 Application Relationship Application Service Element Definition	15
3.6.11 Application Relationship Endpoint Definition	15
3.6.12 Application Definition	15
3.6.13 Application Objects Definition	15
3.6.14 Attribute Definition	15
3.6.15 Behaviour Definition	15
3.6.16 Bit-no Definition	15
3.6.17 Channel Definition	15
3.6.18 Channel Related Diagnosis Definition	15
3.6.19 Class Definition	15
3.6.20 Class Attributes Definition	16
3.6.21 Class Code Definition	16
3.6.22 Class Specific Service Definition	16
3.6.23 Client Definition	16
3.6.24 Communication Objects Definition	16
3.6.25 Connection Definition	16
3.6.26 Connection ID (CID) Definition	16
3.6.27 Connection Path Definition	16
3.6.28 Connection Point Definition	16
3.6.29 Consume Definition	16
3.6.30 Consumer Definition	16
3.6.31 Consuming Application Definition	16
3.6.32 Configuration Check Definition	17
3.6.33 Configuration Fault Definition	17
3.6.34 Configuration Identifier Definition	17
3.6.35 Control Commands Definition	17
3.6.36 Conveyance Path Definition	17
3.6.37 Cyclic Definition	17
3.6.38 Data Consistency Definition	17

3.6.39	Dedicated AR Definition	17
3.6.40	Default DL-Address Definition.....	17
3.6.41	Device Definition	17
3.6.42	Device Profile Definition	17
3.6.43	Diagnosis Information Definition	17
3.6.44	Diagnosis Information Collection Definition	17
3.6.45	DP-Master (Class 1) Definition	18
3.6.46	DP-Master (Class 2) Definition	18
3.6.47	DP-Slave Definition	18
3.6.48	Dynamic AR Definition	18
3.6.49	End Node Definition	18
3.6.50	Endpoint Definition	18
3.6.51	Error Definition	18
3.6.52	Error Class Definition	18
3.6.53	Error Code Definition	18
3.6.54	FAL Subnet Definition	18
3.6.55	First-In-First-Out Variable Definition	18
3.6.56	Frame Definition.....	18
3.6.57	Freeze Definition	18
3.6.58	Group Definition	19
3.6.59	Invocation Definition.....	19
3.6.60	I/O Data Definition.....	19
3.6.61	Ident Number Definition	19
3.6.62	Identifier Related Diagnosis Definition	19
3.6.63	Index Definition	19
3.6.64	Instance Definition	19
3.6.65	Instance Attributes Definition	19
3.6.66	Instantiated Definition	19
3.6.67	Lpacket Definition	19
3.6.68	Management Information Definition.....	19
3.6.69	Manufacturer ID Definition	19
3.6.70	Master Parameter Set Definition.....	20
3.6.71	Member Definition.....	20
3.6.72	Message Router Definition.....	20
3.6.73	Multicast Connection Definition.....	20
3.6.74	Module Definition	20
3.6.75	Network Definition.....	20
3.6.76	Object Definition	20
3.6.77	Object Specific Service Definition	20
3.6.78	Originator Definition	20
3.6.79	Peer Definition	20
3.6.80	Point-to-point Connection Definition	20
3.6.81	Pre-defined AR Endpoint Definition	21
3.6.82	Pre-established AR Endpoint Definition.....	21
3.6.83	Process Data Definition	21
3.6.84	Produce Definition.....	21
3.6.85	Producer Definition	21
3.6.86	Publisher Definition	21
3.6.87	Publishing Manager Definition	21
3.6.88	Pull Publisher Definition	21
3.6.89	Pull Publishing Manager Definition	21
3.6.90	Push Publisher Definition.....	21
3.6.91	Push Publishing Manager Definition	21
3.6.92	Pull Subscriber Definition.....	22
3.6.93	Push Subscriber Definition.....	22
3.6.94	Real Configuration Definition	22
3.6.95	Resource Definition.....	22
3.6.96	Route Endpoint	22
3.6.97	Serial Number Definition	22
3.6.98	Server Definition	22
3.6.99	Server Definition	22

3.6.100	Service Definition	22
3.6.101	Slot Definition	22
3.6.102	Subscriber Definition	22
3.6.103	Sync Definition	22
3.6.104	Target Definition	23
3.6.105	Unconnected Message Manager (UCMM) Definition	23
3.6.106	Unconnected Service Definition	23
3.7	Abbreviations and Symbols	23
3.8	Conventions	24
3.8.1	General Conventions	24
3.8.2	Conventions for Class Definitions	25
3.8.3	Conventions for Service Definitions	26
4	CONCEPTS	28
4.1	Architectural relationships	28
4.1.1	Overview	28
4.1.2	Relationship to the Application Layer of the OSI Reference Model	28
4.1.3	Relationships to other fieldbus standards	29
4.2	Fieldbus Application Layer structure	30
4.2.1	Overview	30
4.2.2	Fundamental concepts	30
4.2.3	Fieldbus Application processes	31
4.2.4	Application objects	34
4.2.5	Application entities	36
4.2.6	Fieldbus Application Service Elements	37
4.2.7	Application relationships	40
4.3	Fieldbus Application Layer Naming and Addressing	43
4.3.1	Identifying Objects accessed through the FAL	43
4.3.2	Addressing APs accessed through the FAL	44
4.4	Architecture Summary	44
4.5	FAL Service Procedures and Time Sequence Diagrams	45
4.5.1	FAL Confirmed Service Procedures	45
4.5.2	Confirmed Service Time Sequence Diagram	46
4.5.3	FAL Unconfirmed Service Procedures	47
4.5.4	Unconfirmed Service Time Sequence Diagram	47
4.6	Common FAL attributes	47
4.7	Common FAL Service Parameters	48
4.8	APDU size	50
5	DATA TYPE ASE	51
5.1	Overview	51
5.1.1	Overview Of Basic Types	52
5.1.2	Overview of Constructed Types	52
5.1.3	Specification of User Defined Data Types	53
5.1.4	Transfer of User Data	53
5.2	Formal Definition of Data Type Objects	53
5.2.1	Data Type Class	53
5.3	FAL Defined Data Types	55
5.3.1	Fixed Length Types	55
5.3.2	String Types	69
5.3.3	Structure Types	71
5.4	Data type ASE Service Specification	73
5.5	Summary of data types	73
6	TYPES OF COMMUNICATION MODELS	74
7	COMMUNICATION MODEL TYPE 1 SPECIFICATION	74
7.1	Type 1 Concepts	74
7.2	Type 1 ASEs	74
7.2.1	Object Management ASE	74
7.2.2	Application Process ASE	84

7.2.3	Application Relationship ASE.....	108
7.2.4	Variable ASE.....	133
7.2.5	Event ASE.....	163
7.2.6	Load Region ASE.....	192
7.2.7	Function Invocation ASE.....	214
7.3	Type 1 ARs.....	232
7.3.1	Queued User-triggered Uni-directional (QUU) AR Endpoint Formal Model.....	232
7.3.2	Queued User-triggered Bi-directional Connection-Oriented (QUB-Co) AR Endpoint Formal Model.....	233
7.3.3	Queued User-triggered Bi-directional Connectionless (QUB-CL) AR Endpoint Formal Model.....	235
7.3.4	Queued User-triggered Bi-directional with Flow Control (QUB-FC) AR Endpoint Formal Model.....	238
7.3.5	Queued User-Triggered Bi-directional with Segmentation(QUB-Seg) AR Endpoint Formal Model.....	241
7.3.6	Buffered User-triggered Bi-directional (BUB) AR Endpoint Formal Model.....	244
7.3.7	Buffered Network-Scheduled Uni-directional (BNU) AR Endpoint Formal Model.....	247
7.3.8	Buffered Network-scheduled Bi-directional (BNB) AR Endpoint Formal Model.....	251
7.3.9	Buffered Network-Scheduled And Unscheduled Uni-directional (BNU-Mp) AR Endpoint Formal Model.....	254
7.4	Description.....	254
L.1	Formal Class Definition.....	257
7.5	Summary of Type 1 FAL Classes.....	259
7.6	Permitted FAL Type 1 Services by AREP Role.....	260
8	COMMUNICATION MODEL TYPE 2 SPECIFICATION.....	262
8.1	Type 2 Concepts.....	262
8.1.1	General.....	262
8.1.2	General concepts.....	262
8.1.3	Relationships between ASE's.....	263
8.1.4	Naming and Addressing.....	264
8.1.5	Type 2 Data Types.....	264
8.2	Type 2 ASEs.....	269
8.2.1	Type 2 Object Management ASE.....	269
8.2.2	Connection Manager ASE.....	317
8.3	Type 2 AR's.....	328
8.3.1	Overview.....	328
8.3.2	UCMM AR Formal Model.....	338
8.3.3	Transport AR Formal Model.....	340
8.3.4	AR ASE Services.....	357
8.4	Summary of FAL Type 2 Classes.....	365
8.5	Permitted FAL Type 2 Services by AR Type.....	366
9	COMMUNICATION MODEL TYPE 3 SPECIFICATION.....	367
9.1	Type 3 Concepts.....	367
9.1.1	User Requirements of the Type 3 DP System.....	367
9.1.2	Features of Type 3 DP.....	367
9.1.3	DP Communication Model.....	368
9.1.4	Dynamical Behaviour of Type 3 DP.....	379
9.1.5	Type 3 Status Values.....	380
9.2	Type 3 ASEs.....	381
9.2.1	Process Data ASE.....	381
9.2.2	I/O Data ASE.....	392
9.2.3	Diagnosis ASE.....	412
9.2.4	Alarm ASE.....	432
9.2.5	Context ASE.....	439
9.2.6	Management ASE.....	472
9.2.7	AR ASE.....	493
9.3	Type 3 ARs.....	498
9.3.1	MS0 Application Relationship.....	498
9.3.2	MS1 Application Relationship.....	501

9.3.3	MS2 Application Relationship.....	502
9.3.4	MM1 Application Relationship.....	506
9.3.5	MM2 Application Relationship.....	507
9.4	Summary of Type 3 FAL Classes.....	598
9.5	Permitted FAL Type 3 Services by AREP Role.....	599
9.6	Conformance Classes.....	601
9.7	Application Characteristics.....	602
9.7.1	Time Behaviour.....	602
9.7.2	Ident Number.....	603
10	COMMUNICATION MODEL TYPE 4 SPECIFICATION	603
10.1	Type 4 Concepts.....	603
10.1.1	Overview.....	603
10.1.2	Application Entities.....	603
10.1.3	Gateway and Routing.....	605
10.1.4	Architecture Summary.....	606
	FAL Service Procedures and Time Sequence Diagrams.....	607
10.2	Type 4 Variable ASE.....	609
10.2.1	Overview.....	609
10.2.2	Variable Model Class Specification.....	611
10.2.3	Basic Variable Type Specifications.....	612
10.2.4	Constructed Variable Type Specifications.....	617
10.2.5	Services.....	621
10.3	Type 4 Route Endpoint ASE.....	621
10.3.1	Overview.....	621
10.3.2	Route Endpoint Model.....	622
10.3.3	Route Endpoint ASE Service Specification.....	624
10.4	Type 4 Application Relationship ASE.....	629
10.4.1	Overview.....	629
10.4.2	Application Relationship Class Specification.....	630
10.4.3	Application Relationship ASE Service Specifications.....	632
11	COMMUNICATION MODEL TYPE 5 SPECIFICATION	636
11.1	Type 5 Concepts.....	636
11.1.1	Objectives of the Type 5 FAL AE.....	636
11.1.2	Type 5 Devices.....	636
11.1.3	Data Type ASE.....	637
11.1.4	APO ASEs.....	637
11.1.5	Application Relationships.....	639
11.1.6	Support for Configuration Management.....	640
11.2	Type 5 ASEs.....	640
11.2.1	Simple Virtual Field Device ASE.....	640
11.2.2	System Management Kernel ASE.....	647
11.2.3	LAN Redundancy ASE.....	664
11.3	Type 5 ARs.....	668
11.3.1	Publisher/Subscriber Type 5 AR Endpoint Formal Model.....	668
11.3.2	Client / Server AR Endpoint Formal Model.....	669
11.4	Summary of FAL Type 1 and Type 5 Classes.....	671
11.5	Permitted FAL Type 1 and Type 5 Services by AREP Role.....	672
12	COMMUNICATION MODEL TYPE 6 SPECIFICATION	673
13	COMMUNICATION MODEL TYPE 7 SPECIFICATION	673
13.1	Type 7 Concepts.....	673
13.1.1	AL Type 7 environment.....	673
13.1.2	Application services.....	673
13.1.3	Application process.....	674
13.1.4	Application entity.....	675
13.1.5	AE model.....	676
13.1.6	AEI model.....	677
13.1.7	Application context.....	679

13.1.8	Coordination of the AEs	683
13.1.9	Concepts for Sub_MMS	687
13.2	Type 7 ASEs	689
13.2.1	MPS ASE (Periodic/Aeriodic Manufacturing Services)	689
13.2.2	Virtual Model of a Device (VMD) ASE	783
13.2.3	Domain ASE	803
13.2.4	Program Invocation (PI) ASE	820
13.2.5	Variable ASE	833
13.2.6	Event ASE	851
13.2.7	Directory ASE	864
13.3	Type 7 ARs	864
13.3.1	Messaging Common Service (MCS) AR ASE	864
14	COMMUNICATION MODEL TYPE 8 SPECIFICATION	885
14.1	Type 8 Concepts	885
14.1.1	Overview	885
14.1.2	Mapping of functional capabilities	885
14.1.3	Fundamental concept restrictions	885
14.1.4	Common FAL Attributes	887
14.2	Type 8 ASEs	887
14.2.1	Overview	887
14.2.2	Object Management ASE	887
14.2.3	Application Process ASE	888
14.2.4	Application Relationship ASE	892
14.2.5	Variable ASE	895
14.2.6	Function Invocation ASE	899
14.3	Type 8 Application Relationships	903
14.3.1	Queued User-triggered Bi-directional with Flow Control (QUB-FC) AR Endpoint Formal Model	903
14.1.1	Buffered Network-Scheduled Uni-directional (BNU) AR Endpoint Formal Model	903
14.2	Permitted FAL Services by AREP Role	904
Annex A. (informative)	Model for Service Error Reporting	905
A.1	General	905
A.2	Error Handling Procedure	906
Bibliography		908

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**DIGITAL DATA COMMUNICATIONS FOR MEASUREMENT AND CONTROL –
FIELDBUS FOR USE IN INDUSTRIAL CONTROL SYSTEMS –**

Part 5: Application Layer Service definition

FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international cooperation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents published have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this standards may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

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

This second edition cancels and replaces the first edition which was issued as a technical specification in 1999. It constitutes a technical revision and now has the status of an International Standard.

This second edition adds seven distinct sets of services, each with a corresponding protocol, to the set of services and protocols of the first edition.

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

FDIS	Report on voting
65C/225/FDIS	65C/230/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 the ISO/IEC Directives, Part 3.

IEC 61158 consists of the following parts, under the general title *Digital data communications for measurement and control — Fieldbus for use in industrial control systems*:

- Part 1: Introductory guide (under preparation)
- Part 2: Physical layer specification and service definition
- Part 3: Data Link Service definition
- Part 4: Data Link Protocol specification
- Part 5: Application layer service definition
- Part 6: Application layer protocol specification
- Part 7: System management (under consideration)
- Part 8: Conformance testing (under consideration)

Annexe A is for information only.

The committee has decided that the contents of this publication will remain unchanged until 2006. At this date, the publication will be:

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

Withdrawn

INTRODUCTION

This standard describes the Fieldbus Application Layer services intended to support the information interchange and the interactions between application processes.

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

This application layer standard does not contain test specifications used to demonstrate compliance with IEC 61158-5 and IEC 61158-6.

Withdrawn

DIGITAL DATA COMMUNICATIONS FOR MEASUREMENT AND CONTROL – FIELDBUS FOR USE IN INDUSTRIAL CONTROL SYSTEMS –

Part 5: Application Layer Service definition

1 Scope

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”.

The FAL is an Application Layer Communication Standard designed to support the conveyance of time-critical and non-time-critical application requests and responses among devices in an automation environment. The term “time-critical” is used to represent the presence of an application time-window, within which one or more specified actions are required to be completed with some defined level of certainty.

This standard specifies the structure and services of the IEC Fieldbus Application Layer (FAL). It is specified in conformance with the OSI Basic Reference Model (ISO/IEC 7498) 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.

This part of IEC 61158 specifies interactions between remote applications in terms of

- an abstract model for defining application resources (objects) capable of being manipulated by users via the use of FAL Services;
- the primitives (interactions between the FAL and the FAL user) associated with each FAL Service;
- the parameters associated with each primitive;
- the interrelationship between and the valid sequences of the primitives for each service.

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.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 61158. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

IEC 61131-1: 1992, *Programmable controllers – Part 1: General information*

IEC 61131-3:1993, *Programmable controllers — Part 3: Programming languages*

IEC 61158–3:2000, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems– Part 3: Data Link Layer service definition*

IEC 61158–4: 2000, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 4: Data Link Layer Protocol Specification*

IEC 61158–6: 2000, *Digital data communications for measurement and control – Fieldbus for use in industrial control systems – Part 6: Application Layer Protocol Specification*

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

ISO/IEC 7498 (all parts), *Information technology – Open Systems Interconnection – Basic Reference Model*

ISO/IEC 7498–1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO 7498–3:1997, *Information technology – Open Systems Interconnection – Basic Reference Model – Part 3: Naming and addressing*

ISO/IEC 8822:1994, *Information technology – Open Systems Interconnection – Presentation service definition*

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

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

ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

ANSI/IEEE 754:– *IEEE Standard for Binary Floating–Point arithmetic*

IEEE 754:1985 (R1990), *Binary floating-point arithmetic*

3 Definitions

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

3.1 ISO/IEC 7498-1 terms

For the purposes of this part of IEC 61158, the following terms as defined in ISO/IEC 7498-1 apply:

- a) application entity;
- b) application process;
- c) application protocol data unit;
- d) application service element;
- e) application entity invocation;
- f) application process invocation;
- g) application transaction;
- h) real open system; and
- i) transfer syntax.

3.2 ISO/IEC 8822 terms

For the purposes of this part of IEC 61158, the following terms as defined in ISO/IEC 8822 apply:

- a) abstract syntax; and
- b) presentation context.

3.3 ISO/IEC 9545 terms

For the purposes of this part of IEC 61158, the following terms as defined in ISO/IEC 9545 apply:

- a) application-association;
- b) application-context;
- c) application context name;
- d) application-entity-invocation;
- e) application-entity-type;
- f) application-process-invocation;
- g) application-process-type;
- h) application-service-element; and
- i) application control service element.

3.4 ISO/IEC 8824 terms

For the purposes of this part of IEC 61158, the following terms as defined in ISO/IEC 8824 apply:

- a) object identifier; and
- b) type.

3.5 Fieldbus Data Link Layer terms

For the purposes of this part of IEC 61158, the following terms as defined in IEC 61158-3 and IEC 61158-4 apply.

- a) DL-Time
- b) DL-Scheduling-policy
- c) DLCEP
- d) DLC
- e) DL-connection-oriented mode
- f) DLSDU
- g) DLSAP
- h) link
- i) network address / node address / MAC ID
- j) node
- k) tag (fixed tag, generic tag)
- l) scheduled, unscheduled

3.6 FAL Specific Definitions

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

3.6.1 Access Protection Definition

The access to certain functions of a server is restricted to one client.

3.6.2 Address Assignment Table Definition

Mapping of the client's internal I/O-addresses to the decentralised inputs and outputs.

3.6.3 Allocate Definition

to take a resource from a common area and assign that resource for the exclusive use of a specific entity.

3.6.4 Application Layer Interoperability Definition

capability of application entities to perform coordinated and cooperative operations using the services of the FAL

3.6.5 Application Object Definition

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

3.6.6 Application Process Definition

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

3.6.7 Application Process Identifier Definition

If multiple application processes are used in a device these application processes are addressed with an Application Process Identifier.

3.6.8 Application Process Object Class Definition

description of a class of application process objects in terms of the set of its network-accessible attributes and services.

3.6.9 Application Relationship Definition

cooperative relationship between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities

3.6.10 Application Relationship Application Service Element Definition

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

3.6.11 Application Relationship Endpoint Definition

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

3.6.12 Application Definition

function or data structure for which data is consumed or produced.

3.6.13 Application Objects Definition

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

3.6.14 Attribute Definition

a description of an externally visible characteristic or feature of an object. 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 behaviour of an object. Attributes are divided into class attributes and instance attributes.

3.6.15 Behaviour Definition

indication of how the object responds to particular events. Its description includes the relationship between attribute values and services.

3.6.16 Bit-no Definition

A Bit-no designates the number of a bit in a bitstring or an octet.

3.6.17 Channel Definition

A single physical or logical connection of inputs or outputs of a server to the process.

3.6.18 Channel Related Diagnosis Definition

This diagnosis information refers to a single input or output channel of the server.

3.6.19 Class Definition

a set of objects, all of which represent the same kind of system component. A class is a generalisation of the object; a template for defining variables and methods. All objects in a class are identical in form and behaviour, but usually contain different data in their attributes.

3.6.20 Class Attributes Definition

an attribute that is shared by all objects within the same class.

3.6.21 Class Code Definition

a unique identifier assigned to each object class.

3.6.22 Class Specific Service Definition

a service defined by a particular object class to perform a required function which is not performed by a common service. A class specific object is unique to the object class which defines it.

3.6.23 Client Definition

(1) an object which uses the services of another (server) object to perform a task.

(2) an initiator of a message to which a server reacts, such as the role of an AR endpoint in which it issues confirmed service request APDUs to a single AR endpoint acting as a server

3.6.24 Communication Objects Definition

components that manage and provide run time exchange of messages across the network such as the Connection Manager object, the unconnected message manager (UCMM), and the Message Router object.

3.6.25 Connection Definition

a logical binding between two application objects. These application objects may be within the same or different devices.

3.6.26 Connection ID (CID) Definition

identifier assigned to a transmission that is associated with a particular connection between producers and consumers that identifies a specific piece of application information.

3.6.27 Connection Path Definition

the attribute is made up of a byte stream which defines the application object to which a connection instance applies.

3.6.28 Connection Point Definition

a buffer which is represented as a subinstance of an assembly object.

3.6.29 Consume Definition

the act of receiving data from a producer.

3.6.30 Consumer Definition

a node that is receiving data from a producer.

3.6.31 Consuming Application Definition

the application that consumes data.

3.6.32 Configuration Check Definition

In the start-up phase, the client transmits the expected configuration for comparison with the real configuration to the server.

3.6.33 Configuration Fault Definition

If the server detects a not acceptable difference between the expected configuration and the real configuration, then a configuration fault is indicated in the diagnosis information.

3.6.34 Configuration Identifier Definition

For each input- and/or output-module of a server at least one Configuration Identifier is defined in the configuration string (either a Cfg Identifier or Special Cfg Identifier).

3.6.35 Control Commands Definition

Control commands will be transferred from the client to the server. With the control commands it is possible to clear the outputs, or to freeze the inputs and/or synchronise the outputs of the server.

3.6.36 Conveyance Path Definition

unidirectional flow of APDUs across an application relationship

3.6.37 Cyclic Definition

term used to describe events which repeat in a regular and repetitive manner.

3.6.38 Data Consistency Definition

The input- or output-data-range shall always be transmitted consistent between the client and the server.

3.6.39 Dedicated AR Definition

AR used directly by the FAL User. On Dedicated ARs, only the FAL Header and the user data are transferred

3.6.40 Default DL-Address Definition

The default DL-address of a server is 126. This address will be used as initial value if address setting will be done via Type 3 DP.

3.6.41 Device Definition

a physical hardware connection to the link. A device may contain more than one node.

3.6.42 Device Profile Definition

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

3.6.43 Diagnosis Information Definition

This term is used to identify the whole diagnosis data of a server.

3.6.44 Diagnosis Information Collection Definition

A system diagnosis information that is collected at the client.

3.6.45 DP-Master (Class 1) Definition

The DP-Master (Class 1) is a controlling device which controls several DP-Slaves (field devices); usually a programmable controller or distributed control system.

3.6.46 DP-Master (Class 2) Definition

The DP-Master (Class 2) is a controlling device which manages configuration data (parameter sets) and diagnosis data of a DP-Master (Class 1). Additionally the DP-Master (Class 2) can perform all communication capabilities of a DP-Master (Class 1).

3.6.47 DP-Slave Definition

A field device that is assigned to one DP-Master (Class 1) as a provider for cyclic I/O data exchange. In addition acyclic functions and alarms could be provided.

3.6.48 Dynamic AR Definition

AR that requires the use of the AR establishment procedures to place it into an established state

3.6.49 End Node Definition

a producing or consuming node.

3.6.50 Endpoint Definition

one of the communicating entities involved in a connection.

3.6.51 Error Definition

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

3.6.52 Error Class Definition

general class for error definitions. Error codes for specific errors are defined within an error class

3.6.53 Error Code Definition

identification of a specific type of error within an error class

3.6.54 FAL Subnet Definition

networks composed of one or more data link segments. They are permitted to contain bridges, but not routers. FAL subnets are identified by a subset of the network address.

3.6.55 First-In-First-Out Variable Definition

A FIFO (First In First Out) is a Type 4 Variable Object class, composed of a set of homogeneously typed elements. The first written element will be the first element that can be read. On the Fieldbus only one, complete element can be transferred as a result of one service invocation.

3.6.56 Frame Definition

single data transfer on a link.

3.6.57 Freeze Definition

The input data of several DP-Slaves will be simultaneously frozen.

3.6.58 Group Definition

A set of DP-Slaves which perform a Freeze or Sync operation.

3.6.59 Invocation Definition

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

3.6.60 I/O Data Definition

Data which have to be transferred cyclically for the purpose of processing.

3.6.61 Ident Number Definition

The Ident Number specifies a DP-Master (Class 1) or DP-Slave device type.

3.6.62 Identifier Related Diagnosis Definition

This diagnosis information is related to the input- or output-range for which an Identifier is defined in the configuration string.

3.6.63 Index Definition

Address of an object within an application process.

3.6.64 Instance Definition

the actual physical occurrence of an object within a class. Identifies one of many objects within the same object class. For example: California is an instance of the object class state. The terms object, instance, and object instance are used to refer to a specific instance.

3.6.65 Instance Attributes Definition

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

3.6.66 Instantiated Definition

an object that has been created in a device.

3.6.67 Lpacket Definition

the type 2 Lpacket (or link packet) is a piece of application information that contains a size, control byte, tag, and link data. Peer Data Link Layers use Lpackets to send and receive service data units from higher layers in the OSI stack.

3.6.68 Management Information Definition

network-accessible information that supports the management of the Fieldbus environment, including the application layer

3.6.69 Manufacturer ID Definition

identification of each manufacturer of Type 2 products by number. Manufacturer IDs uniquely identify a particular manufacturer of Type 2 products.

3.6.70 Master Parameter Set Definition

The Master parameter set includes the configuration and parameterisation data of all DP-Slaves that are assigned to the corresponding DP-Master and the bus parameters.

3.6.71 Member Definition

a piece of an attribute that is structured as an array.

3.6.72 Message Router Definition

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

3.6.73 Multicast Connection Definition

a connection from one node to many. Multicast connections allow messages from a single producer to be received by many consumer nodes.

3.6.74 Module Definition

Addressable unit inside the DP-Slave.

3.6.75 Network Definition

a series of nodes connected by some type of communication medium. The connection paths between any pair of nodes can include repeaters, routers and gateways.

3.6.76 Object Definition

an abstract representation of a particular component within a device, i.e. :

- (1) An abstract representation of a computer's capabilities. Objects can be composed of any or all of the following components:
 - a) data (information which changes with time),
 - b) configuration (parameters for behaviour),
 - c) methods (things that can be done using data and configuration).
- (2) A collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behaviour.

3.6.77 Object Specific Service Definition

a service defined by a particular object class to perform a required function which is not performed by a common service. An object specific service is unique to the object class which defines it.

3.6.78 Originator Definition

the client responsible for establishing a connection path to the target.

3.6.79 Peer Definition

role of an AR endpoint in which it is capable of acting as both client and server

3.6.80 Point-to-point Connection Definition

a connection that exists between two nodes only. Connections can be either point-to-point or multicast.

3.6.81 Pre-defined AR Endpoint Definition

AR endpoint that is defined locally within a device without use of the create service. Pre-defined ARs that are not pre-established must be established before being used

3.6.82 Pre-established AR Endpoint Definition

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

3.6.83 Process Data Definition

Data which are already pre-processed and transferred acyclically for the purpose of information or further processing.

3.6.84 Produce Definition

act of sending data to be received by a consumer.

3.6.85 Producer Definition

a node that is responsible for sending data.

3.6.86 Publisher Definition

role of an AR endpoint in which it transmits APDUs onto the Fieldbus for consumption by one or more subscribers. The publisher may not be aware of the identity or the number of subscribers and it may publish its APDUs using a dedicated AR. Two types of publishers are defined by this standard, Pull Publishers and Push Publishers, each of which is defined separately

3.6.87 Publishing Manager Definition

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. Two types of publishing managers are defined by this standard, Pull Publishing Managers and Push Publishing Managers, each of which is defined separately

3.6.88 Pull Publisher Definition

type of Publisher that publishes an object in confirmed service response APDUs that are responses to confirmed service request APDUs received from its Pull Publishing Manager. The AR used to support Pull Publishers conveys the response APDUs to one or more Pull Subscriber AR endpoints

3.6.89 Pull Publishing Manager Definition

type of Publishing Manager that sends confirmed service request APDUs to its Publisher to request that the specified object be published in the corresponding response APDU

3.6.90 Push Publisher Definition

type of Publisher that publishes an object in unconfirmed service request APDUs. These unconfirmed service request APDUs may be issued as a result of a confirmed service request APDU received on a separate AREP from its Push Publishing Manager. The AR used to support Push Publishers conveys the response APDUs to one or more Pull Subscriber AR endpoints.

3.6.91 Push Publishing Manager Definition

type of Publishing Manager that sends confirmed service requests to its Publisher to request that the specified object be published using an unconfirmed service on a separate Push Publisher AR endpoint

3.6.92 Pull Subscriber Definition

type of Subscriber that recognizes received confirmed service response APDUs as published object data

3.6.93 Push Subscriber Definition

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

3.6.94 Real Configuration Definition

The real configuration describes the input and output data structure of the DP-Slave. Additionally, the definition of the data consistency is included in the real configuration.

3.6.95 Resource Definition

resource is a processing or information capability of a subsystem

3.6.96 Route Endpoint

A Route Endpoint is a Type 4 object container, containing Variable Objects of a Type 4 variable class.

3.6.97 Serial Number Definition

a unique value (32 bit integer) assigned by each manufacturer to every device having Type 2 communication capabilities. Each manufacturer shall be responsible for guaranteeing the uniqueness of the serial number across all of its devices, but the number need only be unique with respect to the manufacturer. When used in conjunction with the Manufacturer ID, it shall form a unique identifier for each device.

3.6.98 Server Definition

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

3.6.99 Server Definition

an object which provides services to another (client) object.

3.6.100 Service Definition

operation or function that an object and/or object class performs upon request from another object and/or object class. A set of common services is defined and provisions for the definition of object-specific services are provided. Object-specific services are those which are defined by a particular object class to perform a required function which is not performed by a common service.

3.6.101 Slot Definition

Address of a module within a DP-Slave.

3.6.102 Subscriber Definition

role of an AREP in which it receives APDUs produced by a publisher. Two types of subscribers are defined by this standard, Pull Subscribers and Push Subscribers, each of which is defined separately

3.6.103 Sync Definition

The outputs of several DP-Slaves will be simultaneously updated.

3.6.104 Target Definition

the end-node to which a connection is established.

3.6.105 Unconnected Message Manager (UCMM) Definition

the component within a node that transmits and receives unconnected explicit messages and sends them directly to the Message Router object.

3.6.106 Unconnected Service Definition

the messaging service which does not rely on the set up of a connection between devices before allowing information exchanges.

3.7 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
API	Application Process Identifier
AR	Application Relationship
AREP	Application Relationship End Point
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
CID	Connection ID
CIM	Computer Integrated Manufacturing
CIP	The Control and Information Protocol corresponding to Type 2. CIP includes both connected and unconnected messaging.
CM_API	Actual Packet Interval
CM_RPI	Requested Packet Interval
Cnf	Confirmation
COR	Connection originator
CR	Communication Relationship
CREP	Communication Relationship End Point
DL-	(as a prefix) Data Link-
DLC	Data Link Connection
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
DP	Decentralised Peripherals
FAL	Fieldbus Application Layer

FIFO	First In First Out
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
ISO	International Organization for Standardization
LME	Layer Management Entity
O2T	Originator to target (connection characteristics)
O⇒T	Originator to target (connection characteristics)
OSI	Open Systems Interconnect
PDU	Protocol Data Unit
PL	Physical Layer
PLC	Programmable Logic Controller
REP	Route Endpoint
Req	Request
Rsp	Response
SAP	Service Access Point
SCL	Security Level
SDU	Service Data Unit
SEM	State event matrix
SMIB	System Management Information Base
SMK	System Management Kernel
STD	State transition diagram, used to describe object behaviour
S-VFD	Simple Virtual Field Device
T2O	Target to originator (connection characteristics)
T⇒O	Target to originator (connection characteristics)
VAO	Type 4 Variable Object

3.8 Conventions

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate clause. Each ASE specification is composed of two parts, its class specification, and its service specification. The protocol specification for each of the ASEs is defined in IEC 61158-6.

The class specification defines the attributes of the class. The attributes are accessible from instances of the class using the Object Management ASE services specified in clause 5 of this standard. The service specification defines the services that are provided by the ASE.

3.8.1 General Conventions

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

3.8.2 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:

FAL ASE:	ASE Name
CLASS:	Class Name
CLASS ID:	#
PARENT CLASS:	Parent Class Name

ATTRIBUTES:

- | | | | |
|-----|-----|----------------|------------------------|
| 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) |

SERVICES:

- | | | | |
|-----|-----|-------------|-----------------------|
| 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 unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2048 are allocated for identifying user defined classes.
- (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 an 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.

3.8.3 Conventions for Service Definitions

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

3.8.3.1 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 uses 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.

3.8.3.2 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.

Withdrawing

4 Concepts

This clause describes fundamentals of the FAL. Each of the ASEs specified for the FAL are described, including their relationships to each other. More detailed descriptive information about each of the FAL ASEs can be found in the “Overview” clause of their specifications.

4.1 Architectural relationships

4.1.1 Overview

The Fieldbus is intended to be used in factories and process plants to interconnect primary automation devices (e.g. sensors, actuators, local display devices, annunciators, programmable logic controllers, small single loop controllers, and stand-alone field controls) with control and monitoring equipment located in control rooms.

Primary automation devices are associated with the lowest levels of the industrial automation hierarchy and perform a limited set of functions within a definite time window. Some of these functions include diagnostics, data validation, and handling of multiple inputs and outputs.

These primary automation devices, also termed field devices, are located close to the process fluids, the fabricated part, the machine, the operator and the environment. This use positions the Fieldbus at the lowest levels of the Computer Integrated Manufacturing (CIM) architecture.

Some of the expected benefits in using Fieldbus are reduction in wiring, increase in amount of data exchanged, wider distribution of control between the primary automation devices and the control room equipment, and the satisfaction of time critical constraints.

4.1.2 Relationship to the Application Layer of the OSI Reference Model

The functions of the FAL have been described according to OSI layering principles. However, its architectural relationship to the lower layers is different, as shown in Figure 1.

- The FAL includes OSI functions together with extensions to cover time-critical requirements. The OSI Application Layer Structure standard (ISO/IEC 9545) was used as a basis for specifying the FAL.
- The Fieldbus Data Link layer directly supports the FAL.

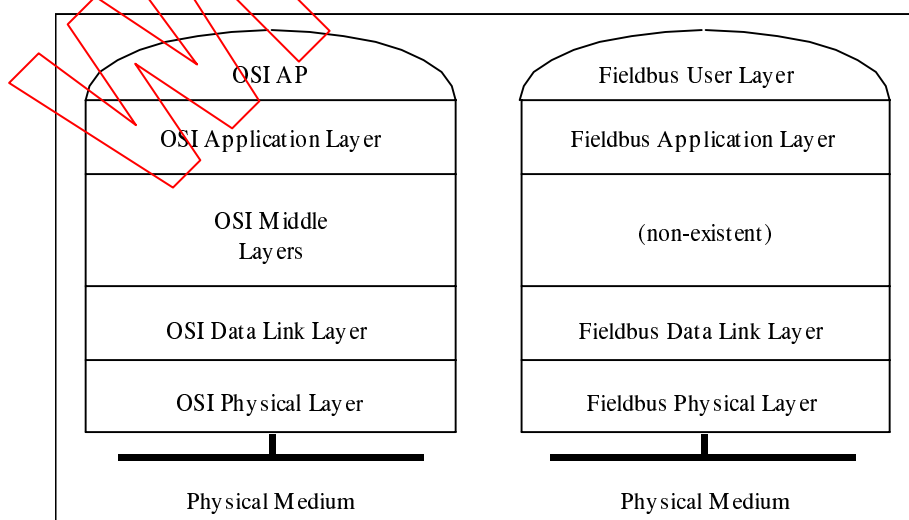


Figure 1 – Relationship to the ISO OSI Reference Model

4.1.3 Relationships to other fieldbus standards

The Fieldbus Application Layer (FAL) architectural relationships, as illustrated in Figure 2, have been designed to support the interoperability needs of time-critical systems distributed within the Fieldbus environment.

Within this environment, the FAL provides communications services to time-critical and non-time-critical applications located in fieldbus devices.

In addition, the FAL directly uses the Data Link Layer to transfer its application layer protocol data units. It does this using a set of data transfer services and a set of supporting services used to control the operational aspects of the Data Link Layer.

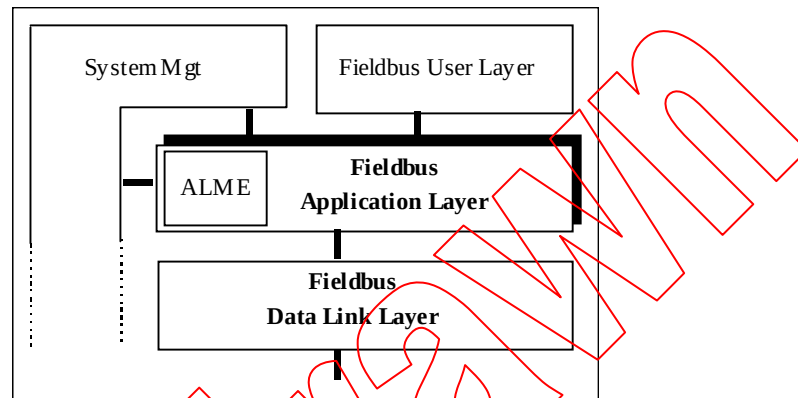


Figure 2 – Architectural Positioning of the Fieldbus Application Layer

4.1.3.1 Use of the Fieldbus Data Link Layer

The Fieldbus Application Layer (FAL) provides network access to Fieldbus APs. It interfaces directly to the Fieldbus Data Link Layer for transfer of its APDUs.

The Data Link Layer provides various types of services to the FAL for the transfer of data between Data Link Service access points (DLSAPs) and Data Link connection endpoints (DLCEPs). The Data Link Layer protocol includes scheduled and non-scheduled access to the physical medium.

4.1.3.2 Support for Fieldbus applications

Fieldbus applications are represented to the network as application processes (APs). APs are the components of a distributed system that may be individually identified and addressed.

Each AP contains an FAL application entity (AE) that provides network access for the AP. That is, each AP communicates with other APs through its AE. In this sense, the AE provides a window of visibility into the AP.

APs contain identifiable components that are also visible across the network. These components are represented to the network as Application Process Objects (APO). They may be identified by one or more key attributes. They are located at the address of the application process that contains them.

The services used to access them are provided by APO-specific application service elements (ASEs) contained within the FAL. These ASEs are designed to support user, function block, and management applications.

4.1.3.3 Support for System Management

The FAL services can be used to support various management operations, including management of fieldbus systems, applications, and the Fieldbus network.

NOTE System management will be specified in the future IEC 61158-7 (in preparation).

4.1.3.4 Access to FAL Layer Management Entities

One layer management entity (LME) is present in each FAL entity on the network. FALMEs provide access to the FAL for system management purposes.

The set of data accessible by the System Manager is referred to as the System Management Information Base (SMIB). Each Fieldbus Application Layer Management Entity (FALME) provides the FAL portion of the SMIB. How the SMIB is implemented is beyond the scope of this standard.

4.2 Fieldbus Application Layer structure

4.2.1 Overview

The structure of the FAL is a refinement of the OSI Application Layer Structure (ISO/IEC 9545). As a result, the organization of this subclause is similar to that of ISO/IEC 9545. Certain concepts presented here have been refined from ISO/IEC 9545 for the Fieldbus environment.

The FAL differs from the other layers of OSI in two principal aspects:

- OSI defines a single type of application layer communications channel, the association, to connect APs to each other. The FAL defines the Application Relationship (AR), of which there are several types, to permit application processes (APs) to communicate with each other.
- The FAL uses the DLL to transfer its PDUs and not the presentation layer. Therefore, there is no explicit presentation context available to the FAL. Between the same pair (or set) of data link service access points the FAL protocol may not be used concurrently with other application layer protocols.

4.2.2 Fundamental concepts

The operation of time critical real open systems is modeled in terms of interactions between time-critical APs. The FAL permits these APs to pass commands and data between them.

Cooperation between APs requires that they share sufficient information to interact and carry out processing activities in a coordinated manner. Their activities may be restricted to a single Fieldbus segment, or they may span multiple segments. The FAL has been designed using a modular architecture to support the messaging requirements of these applications.

Cooperation between APs also often requires that they share a common sense of time. The Data Link Layer specified in IEC 61158-3 and IEC 61158-4 provides for the distribution of time to all devices. It also defines local device services that can be used by APs to access the distributed data link time.

The remainder of this clause describes each of the modular components of the architecture and their relationships with each other. The components of the FAL are modeled as objects, each of which provides a set of FAL communication services for use by applications. The FAL objects and their relationships are described below. The detailed specifications of FAL objects and their services are provided in the following clauses of this standard. IEC 61158-6 specifies the protocols necessary to convey these object services between applications.

4.2.3 Fieldbus Application processes

4.2.3.1 Definition of the Fieldbus AP

In the Fieldbus environment, an application may be partitioned into a set of components and distributed across a number of devices on the network. Each of these components is referred to as a Fieldbus Application Process (AP). A Fieldbus AP is a variation of an Application Process as defined in ISO OSI Reference Model (ISO/IEC 7498). Fieldbus APs may be unambiguously addressed by at least one individual Data Link Layer service access point address. Unambiguously addressed, in this context, means that no other AP may simultaneously be located by the same address. This definition does not prohibit an AP from being located by more than one individual or group data link service access point address.

4.2.3.2 Communication services

Fieldbus APs communicate with each using confirmed and unconfirmed services (ISO/IEC 10731). The services defined in this standard for the FAL specify the semantics of the services as seen by the requesting and responding APs. The syntax of the messages used to convey the service requests and responses is defined in IEC 61158-6. The AP behavior associated with the services is specified by the AP.

Confirmed services are used to define request/response exchanges between APs, such that if a response is not returned to the requesting user within a specified time interval, the requesting user is notified and the service invocation is cancelled.

Unconfirmed services, in contrast, are used to define the unidirectional transfer of messages from one AP to one or more remote APs. From a communications perspective, there is no relationship between separate invocations of unconfirmed services as there is between the request and response of a confirmed service. Therefore, there is no associated time interval associated with the processing of an unconfirmed service by the receiver.

Dedicated ARs provide the AR ASE services directly to the FAL User. They support optimized data transfers between APs in which user data is transferred using only one byte FAL Protocol Control Information (PCI).

4.2.3.3 AP Interactions

Within the Fieldbus environment, APs may interact with other APs as necessary to achieve their functional objectives. No constraints are imposed by this standard on the organization of these interactions or the possible relationships that may exist between them. For example, in the Fieldbus environment, interactions may be based on request/response messages sent directly between APs, or on data/events published by one AP for use by others. These two models of interactions between APs are referred to as client/server and publisher/subscriber interactions.

The services supported by an interaction model are conveyed by application relationship endpoints (AREPs) associated with the communicating APs. The role that the AREP plays in the interaction (e.g. client, server, peer, publisher, subscriber) is defined as an attribute of the AREP. See clause 7 for the definition of the AREP and its attributes. See the annexes following annex B for the definition of each of the AREP types, including their roles. See annex C for a summary of AREP Roles and the FAL services supported by them.

4.2.3.3.1 Client/Server Interactions

Client/server interactions are characterized by a bi-directional data flow between a single client AP and a single server AP, as shown in Figure 3. In this type of interaction, the client may issue a confirmed or unconfirmed request to the server to perform some task. If the service is confirmed then the server will always return a response. If the service is unconfirmed, the server may return a response using an unconfirmed service defined for this purpose.

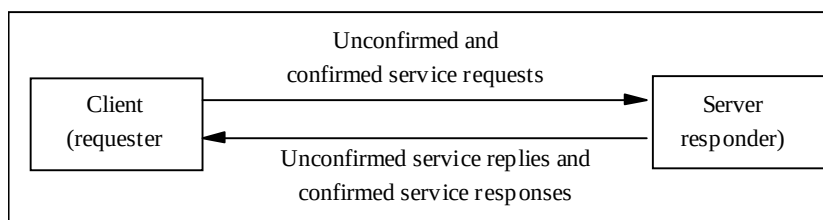


Figure 3 – Client/Server Interactions

4.2.3.3.2 Publisher / Subscriber Interactions

Publisher/subscriber interactions, on the other hand, involve a single publisher AP, and a group of one or more subscriber APs. This type of interaction has been defined to support two models of interaction between APs, the "pull" model and the "push" model. In both models, the setup of the publishing AP is performed by management and is outside the scope of this standard.

4.2.3.3.2.1 Pull Model Interactions

In the "pull" model, the publisher receives a request to publish from a remote publishing *manager*, and broadcasts (or multicasts) its response across the network. The publishing manager is responsible only for initiating publishing by sending a request to the publisher.

Subscribers wishing to receive the published data listen for responses transmitted by the publisher. In this fashion, data is "pulled" from the publisher by requests from the publishing manager.

Confirmed FAL services are used to support this type of interaction. Two characteristics of this type of interaction differentiate it from the other types of interaction. First, a typical confirmed request/response exchange is performed between publishing manager and the publisher. However, the underlying conveyance mechanism provided by the FAL returns the response not just to the publishing manager, but also to all subscribers wishing to receive the published information. This is accomplished by having the Data Link Layer transmit the response to a group address, rather than to the individual address of the publishing manager. Therefore, the response sent by the publisher contains the published data and is multicast to the publishing manager and to all subscribers.

The second difference occurs in the behavior of the subscribers. Pull model subscribers, referred to as pull subscribers, are capable of accepting published data in confirmed service responses without having issued the corresponding request. Figure 4 illustrates these concepts.

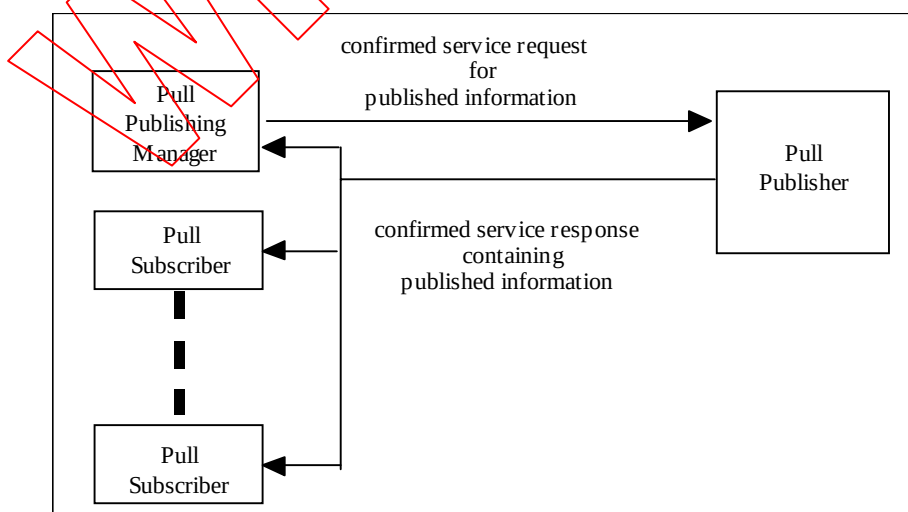


Figure 4 – Pull Model Interactions

4.2.3.3.2 Push Model Interactions

In the "push" model, two services are used, one confirmed and one unconfirmed. The confirmed service is used by the subscriber to request to join the publishing. The response to this request is returned to the subscriber, following the client/server model of interaction. This exchange is only necessary when the subscriber and the publisher are located in different APs.

The unconfirmed service used in the Push Model is used by the publisher to distribute its information to subscribers. In this case, the publisher is responsible for invoking the correct unconfirmed service at the appropriate time and for supplying the appropriate information. In this fashion, it is configured to "push" its data onto the network.

Subscribers for the Push Model receive the published unconfirmed services distributed by publishers. Figure 5 illustrates the concept of the Push Model.

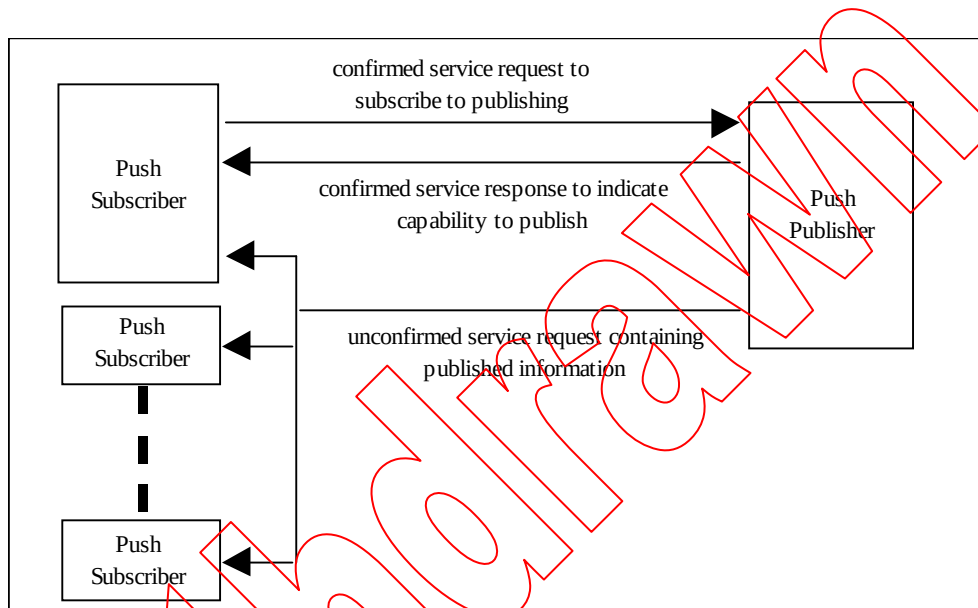


Figure 5 – Push Model Interactions

4.2.3.3.2.3 Timeliness of published information

To support the perishable nature of published information, the FAL supports four types of timeliness defined for publisher/subscriber interactions. Each make it possible for subscribers of published data to determine if the data they are receiving is up-to-date or "stale". These types are realized through mechanisms within the Data Link Layer (DLL). Each is described briefly below. For a more detailed description, refer to IEC 61158-3 and IEC 61158-4.

<u>Type</u>	<u>Description</u>
Transparent	This type of timeliness allows the user application process to determine the timeliness quality of data that it generates and have the timeliness quality accompany the information when it is transferred across the network. In this type of timeliness, the network provides no computation or measurement of timeliness. It merely conveys the timeliness quality provided with the data by the user application process.
Residence	When the FAL submits data from the publishing AP to the DLL for transmission, the DLL starts a timer. If the timer expires before the data has been transmitted, the DLL marks the buffer as "not timely" and conveys this timeliness information with the data.
Synchronized	This type of timeliness requires the coordination of two pieces of published information. One is the data to be published and the other is a special "sync mark". When the sync mark is received from the network a timer starts in each of the participating stations. Subsequently, when data is received for

transmission by the DLL at the publishing station, or when the transmitted data is received from the network at a subscribing station, the DLL timeliness attribute for the data is set to TRUE. It remains TRUE until the reception of the next sync mark or until the timer expires. Data received after the timer expires but before the next sync mark does not cause the timeliness attribute to be reset to TRUE. It is only reset to TRUE if data is received within the time window after receipt of the sync mark. Data transmitted by the publisher station with the timeliness attribute set to FALSE maintains the setting of FALSE at each of the subscribers, regardless of their timer operation.

Update This type of timeliness requires the coordination of the same two pieces of published information defined for *synchronized* timeliness. In this type, the sync mark also starts a timer in each of the participating stations. Like *synchronized* timeliness, expiration of the timer always causes the timeliness attribute to be set to FALSE. Unlike *synchronized* timeliness, receipt of new data at any time (not just within the time window started with the receipt of a sync mark) causes the timeliness attribute to be set to TRUE.

NOTE Time interval values are managed through system management (see the future IEC 61158-7).

4.2.3.4 AP Structure

The internals of APs may be represented by one, or more Application Process Objects (APOs) and accessed through one or more Application Entities (AEs). AEs provide the communication capabilities of the AP. For each Fieldbus AP, there is one and only one FAL AE. APOs are the network representation of application specific capabilities (user application objects) of an AP that are accessible through its AEs.

4.2.3.5 AP Class

An AP class is a definition of the attributes and services of an AP. The standard class definition for APs is defined in this standard. User defined classes also may be specified. Class identifiers (described in 3.8.2) are assigned from a set reserved for this purpose.

4.2.3.6 AP Type

AP types provide the mechanism for defining standard APs.

As described above in the previous subclauses, APs are defined by instantiating an AP class. Each AP definition is composed of the attributes and services selected for the AP from those defined by its AP class. In addition, an AP definition contains values for one or more of the attributes selected for it. When two APs share the same definition, that definition is referred to as an AP type. Thus, an AP type is a generic specification of an AP that may be used to define one or more APs.

The descriptor reference attribute included in AP class definitions may be used to identify the type of the AP.

4.2.4 Application objects

4.2.4.1 Definition of APO

An application object (APO) is a network representation of a specific aspect of an AP. Each APO represents a specific set of information and processing capabilities of an AP that are accessible through services of the FAL. APOs are used to represent these capabilities to other APs in a Fieldbus system.

From the perspective of the FAL, an APO is modeled as a network accessible object contained within an AP or within another APO (APOs may contain other APOs). APOs provide the network definition for objects contained within an AP that are remotely accessible. The definition of an APO includes an identification of the FAL services that can be used by remote APs for remote access. The FAL services, as shown in Figure 6, are provided by the FAL communications entity of the AP, known as the FAL Applications Entity (FAL AE).

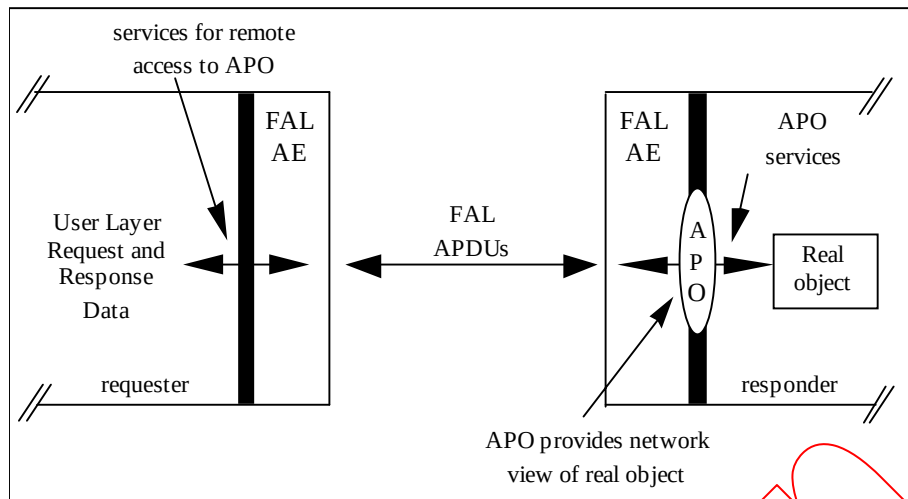


Figure 6 – APOs Services Conveyed by the FAL

In Figure 6, remote APs acting as clients may access the real object by sending requests through the APO that represents the real object. Local aspects of the AP convert between the network view (the APO) of the real object and the internal AP view of the real object.

To support the publisher / subscriber model of interaction, information about the real object can be published through its APO. Remote APs acting as subscribers see the APO view of the published information instead of having to know any of the real object specific details.

4.2.4.2 APO Classes

An APO Class is a generic specification for a set of APOs, each of which is described by the same set of attributes and accessed using the same set of services.

APO Classes provide the mechanism for standardizing network visible aspects of APs. Each standard APO Class definition specifies a particular set of network accessible AP attributes and services. IEC 61158-6 specifies the syntax and the procedures used by the FAL protocol to provide remote access to the attributes and services of an APO Class.

Standard APO Classes are specified by this standard for the purpose of standardizing remote access to APs. User defined classes may also be specified. Class identifiers for user defined classes, as described previously in 3.8.2, are assigned from a set reserved for this purpose.

User defined classes are defined as subclasses of standardized APO Classes or of other user-defined classes. They may be defined by identifying new attributes or by indicating that optional attributes for the parent class are mandatory for the subclass. The conventions for defining classes defined in 3.8.2 may be used for this purpose. The method for registering or otherwise making these new class definitions available for public use is beyond the scope of this standard.

4.2.4.3 APOs as Instances of APO Classes

APO Classes are defined in this standard using templates. These templates are used not only to define APO Classes, but also to specify the instances of a class.

Each APO defined for an AP is an instance of an APO Class. Each APO provides the network view of a real object contained in an AP. An APO is defined by

- (1) selecting the attributes from its APO Class template that are to be accessible from the real object,
- (2) assigning values to one or more attributes indicated as key in the template. Key attributes are used to identify the APO;
- (3) assigning values to zero, one, or more non-key attributes for the APO. Non-key attributes are used to characterize the APO;
- (4) selecting the services from the template that may be used by remote APs to access the real object.

Subclause 3.8.2 of this standard specifies the conventions for class templates. These conventions provide for the definition of mandatory, optional, and conditional attributes and services.

Mandatory attributes and services are required to be present in all APOs of the class. Optional attributes and services may be selected, on an APO by APO basis, for inclusion in an APO. Conditional attributes and services are defined with an accompanying constraint statement. Constraint statements specify the conditions that indicate whether or not the attribute is to be present in an APO.

4.2.4.4 APO Types

APO types provide the mechanism for defining standard APOs.

As described above in the previous subclauses, APOs are defined by instantiating an APO class. Each APO definition is composed of the attributes and services selected for the APO from those defined by its APO class. In addition, an APO definition contains values for one or more of the attributes selected for it. When two APOs share the same definition, except for the key attribute settings, that definition is referred to as an APO type. Thus, an APO type is a generic specification of an APO that may be used to define one or more APOs.

The descriptor reference attribute included in APO Class definitions may be used to identify the type of the APO.

4.2.5 Application entities

4.2.5.1 Definition of FAL AE

An application entity provides the communication capabilities for a single AP. An FAL AE provides a set of services and the supporting protocols to enable communications between APs in a Fieldbus environment. The services provided by FAL AEs are grouped into Application Service Elements (ASE), such that the FAL services provided to an AP are defined by the ASEs its FAL AE contains. Figure 7 illustrates this concept.

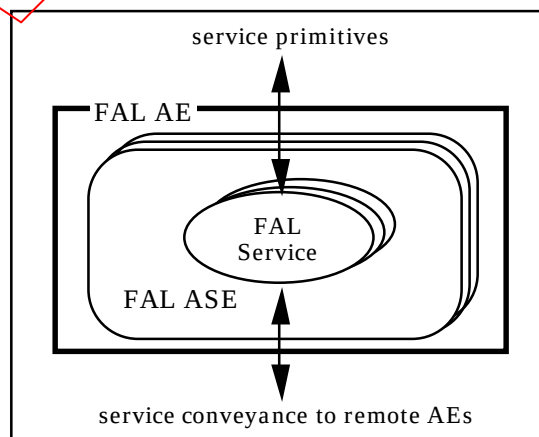


Figure 7 – Application entity structure

4.2.5.2 AE Type

Application entities that provide the same set of ASEs are of the same AE-type. Two AEs must share a common set of ASEs to be capable of communicating with each other.

4.2.6 Fieldbus Application Service Elements

4.2.6.1 General

An application service element (ASE), as defined in ISO/IEC 9545, is a set of application functions that provide a capability for the interworking of application-entity-invocations for a specific purpose. ASEs provide a set of services for conveying requests and responses to and from application processes and their objects. AEs, as defined above, are represented by a collection of ASE invocations within the AE.

4.2.6.2 FAL Services

FAL Services convey functional requests/responses between APs. Each FAL service is defined to convey requests and responses for access to a real object modeled as an FAL accessible object.

The FAL defines both confirmed and unconfirmed services. Confirmed service requests are sent to the AP containing the real object. Each invocation of a confirmed service request is identified by a user supplied Invoke ID. The same Invoke ID is returned in the response by the AP containing the real object. It is used by the requesting AP and its FAL AE to associate the response with the appropriate request.

The FAL AE uses context information associated with the request to help decode the response. If the responding AP fails to respond within a specified time period, the FAL AE uses the Invoke ID to indicate to the requesting AP that the response has timed out and the FAL AE will no longer wait for it.

Unconfirmed services may be sent from the AP containing the real object to publish information about the object. They also may be sent to the AP containing the real object to access the real object. Both types of unconfirmed services are defined for the FAL.

4.2.6.3 Definition of FAL ASEs

A modular approach has been taken in the definition of FAL ASEs. The ASEs defined for the FAL are also object-oriented. Each ASE, except for the Object Management ASE, provides a set of services designed for one specific object class or for a related set of classes. The Object Management ASE provides a common set of management services applicable to all classes of objects. The AP ASE provides a set of services that operates on the AP itself. The Variable, Event, Function Invocation, and Load Region ASEs provide services for accessing the APOs of an AP.

To support remote access to the AP, the Application Relationship ASE is defined. It provides services to the AP for defining and establishing communication relationships with other APs, and it provides services to the other ASEs for conveying their service requests and responses.

Each FAL ASE defines a set of services, APDUs, and procedures that operate on the classes that it represents. Only a subset of the ASE services may be provided to meet the needs of an application. Profiles may be used to define such subsets. Definition of profiles is beyond the scope of this standard.

APDUs are sent and received between FAL ASEs that support the same services. Each FAL AE contains, at a minimum, the AR ASE and at least one other ASE. Figure 8 illustrates a subset of the FAL ASEs and their architectural relationships. All APO ASEs follow this model.

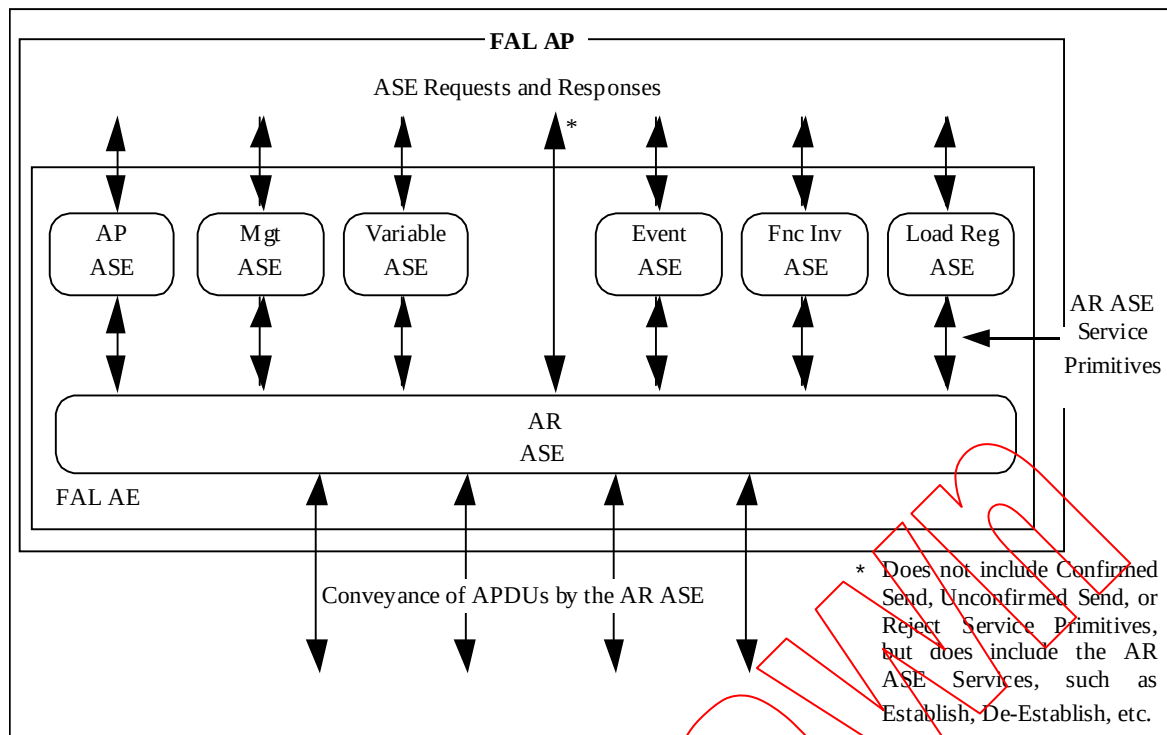


Figure 8 – FAL ASEs

4.2.6.3.1 Object Management ASE

A special object management ASE is specified for the FAL that provides services for the management of objects. Its services are used to access object attributes, and create and delete object instances. These services are used to manage network visible AP objects accessed through the FAL. The specific operational services that apply to each object type are specified in the definition of the ASE for the object type. Figure 9 illustrates the integration of management and operational services for an object within an AP.

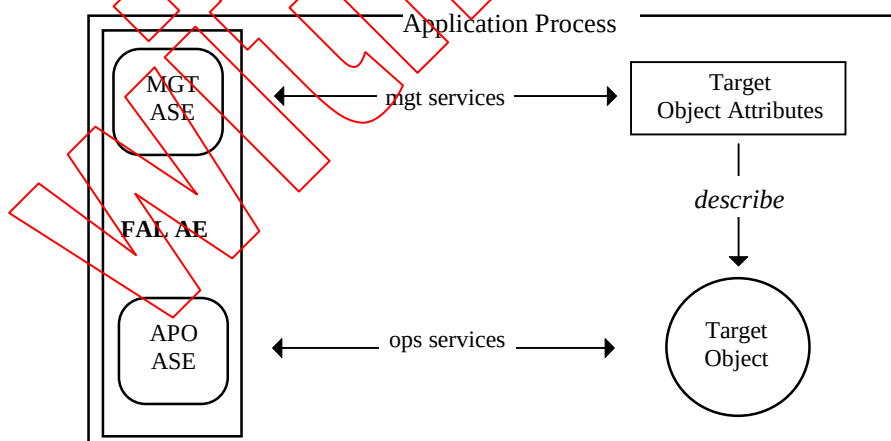


Figure 9 – FAL Management of objects

4.2.6.3.2 AP ASE

An AP ASE is specified for the identification and control of FAL APs. The attributes defined by the AP ASE specify characteristics of the AP about its manufacturer, and list its contents and capabilities. One service defined for the AP ASE supports the push model for publishing data. Using this service, subscribers are able to indicate their requirements for receiving published data. Another service is used to initialize communications with other APs. Other AP specific services are used to identify the manufacturer of the AP, to obtain its operational status.

4.2.6.3.3 APO ASEs

The FAL specifies a set of ASEs with services defined for accessing the APOs of an AP. The APO ASEs defined for the FAL are:

- Data Type ASE
- Variable ASE
- Event ASE
- Load Region ASE
- Function Invocation ASE

4.2.6.3.4 AR ASE

An AR ASE is specified to establish and maintain application relationships (ARs), that are used to convey FAL APDUs between/among APs. ARs represent application layer communication channels between APs. AR ASEs are responsible for providing services at the endpoints of ARs. AR ASE services are defined for establishing, terminating, and aborting ARs, for conveying APDUs for the AE, and for indicating the local status of the AR to the user. In addition, local services are defined for accessing certain aspects of AR endpoints.

Because ARs are communications objects defined and located within the FAL, they are managed by the System Management Agent instead of being managed using the FAL Object Management ASE services. The System Management Agent and the AR managed objects will be defined in the future IEC 61158-7.

4.2.6.4 FAL Service Conveyance

FAL AP, Management, and User APO ASEs provide services to convey requests and responses between service users and real objects. For push publishing services, service requests are conveyed *from* the real object, instead of to it. For all others, service requests are conveyed to the real object and service responses are returned from it.

To accomplish the task of conveying service requests and responses, three types of activities for the sending user and three corresponding types for the receiving user are defined. At the sending user, they accept service requests and responses to be conveyed. Secondly, they select the type of FAL APDU that will be used to convey the request or response and encode the service parameters into its body portion. Then they submit the encoded APDU body to the AR ASE for conveyance.

At the receiving user, they receive encoded APDU bodies from the AR ASE. They decode the APDU bodies and extract the service parameters conveyed by them. To conclude the conveyance, they deliver the service request or response to the user. Figure 10 illustrates these concepts.

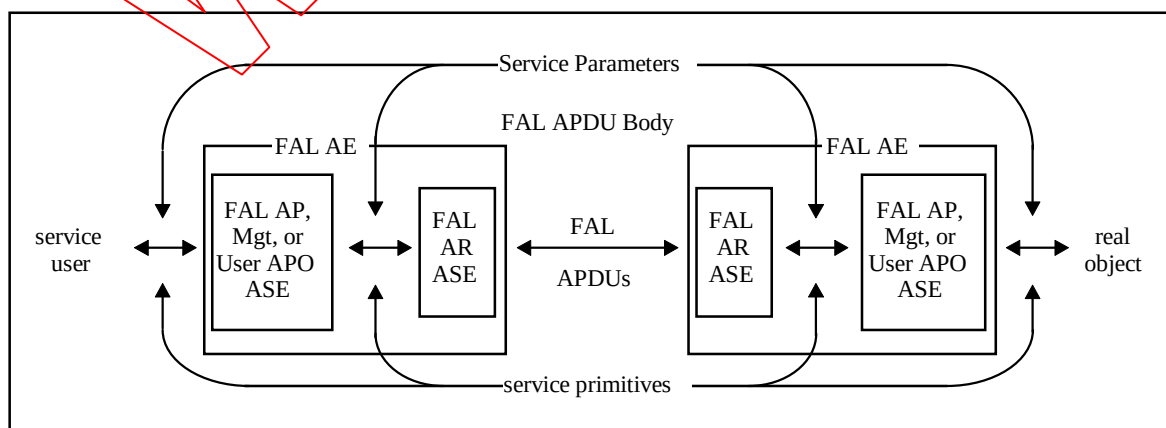


Figure 10 – ASE Service Conveyance

When a confirmed service request is received for conveyance the sending AR ASE starts a timer when it sends the request APDU and waits for the corresponding response APDU. If the timer expires before the response is received, it notifies the user and stops waiting for the response.

In addition, for service requests that operate on AR endpoints, such as establish and de-establish, the AR ASE will perform the processing required to maintain the appropriate context for the affected endpoint(s).

4.2.6.5 FAL Presentation Context

The presentation context in the OSI environment is used to distinguish the APDUs of one ASE from another, and to identify the transfer syntax rules used to encode each APDU. However, the Fieldbus communications architecture does not include the presentation layer. Therefore, an alternate mechanism is provided for the FAL by each of the specific types of Communication Models.

4.2.7 Application relationships

4.2.7.1 Definition of AR

ARs represent communication channels between APs. They define how information is communicated between APs. Each AR is characterized by how it conveys ASE service requests and responses from one AP to another. These characteristics are described below.

4.2.7.2 AR-Endpoints

ARs are defined as a set of cooperating APs. The AR ASE in each AP manages an endpoint of the AR, and maintains its local context. The local context of an AR endpoint is used by the AR ASE to control the conveyance of APDUs on the AR.

4.2.7.3 AR-Endpoint Classes

ARs are composed of a set of endpoints of compatible classes. AR endpoint classes are used to represent AR endpoints that convey APDUs in the same way. Through the standardization of endpoint classes, ARs for different models of interaction can be defined. Endpoint class definitions are contained as annexes to this standard. Additional classes may be added as necessary in annexes.

4.2.7.4 AR Cardinality

ARs characterize communications between APs. One of the characteristics of an AR is the number of AR endpoints in the AR. ARs that convey services between two APs have a cardinality of 1-to-1. Those that convey services from one AP to a number of APs have a cardinality of 1-to-many. ARs that represent a many-to-many conveyance among APs are not supported by the FAL.

4.2.7.5 Accessing Objects through ARs

ARs provide access to APs and the objects within them through the services of one or more ASEs. Therefore, one characteristic is the set of ASE services that may be conveyed to and from these objects by the AR. The list of services that can be conveyed by the AR are selected from those defined for the AE.

4.2.7.6 AR Conveyance Paths

ARs are modeled as one or two conveyance paths between AR endpoints. Each conveyance path conveys APDUs in one direction between one or more AR endpoints. Each receiving AR endpoint for a conveyance path receives all APDUs transmitting on the AR by the sending AR endpoint.

4.2.7.7 AREP Roles

Because APs interact with each other through endpoints, a basic determinant of their compatibility is the role that they play in the AR. The role defines how an AREP interacts with other AREPs in

the AR. Therefore, endpoint roles are defined to support the client/server and publisher/subscriber models of interaction.

For example, an AREP may operate as a client, a server, a publisher, or a subscriber. When an AREP interacts with another AREP on a single AR as both a client and a server, it is defined to have the role of “peer”. Other roles defined for AREPs are used in support of the publisher/subscriber model of interaction, such as publishing manager, “pull” publisher, and “push” publisher.

Certain roles are capable of initiating service requests, while others are only capable of responding to service requests. This part of the definition of a role identifies the requirement for an AR to be capable of conveying requests in either direction, or only in one direction.

4.2.7.8 AREP Buffers and Queues

AREPs may be modeled as a queue or as a buffer provided by the Data Link Layer. APDUs transferred over a queued AREP are delivered in the order received for conveyance. The transfer of APDUs over a buffered AREP is different. In this case, an APDU to be conveyed by the AR ASE is placed in a Data Link Layer buffer for transfer. When the Data Link Layer gains access to the network, it transmits the contents of the buffer.

When the AR ASE receives another conveyance request, it replaces the previous contents of the buffer whether or not they were transmitted. Once an APDU is written into a buffer for transfer, it is preserved in the buffer until the next APDU to be transmitted replaces it. While in the buffer, an APDU may be read more than once without deleting it from the buffer or changing its contents.

At the receiving end, the operation is similar. The receiving Data Link Layer places a received APDU into a buffer for access by the AR ASE. When a subsequent APDU is received, it overwrites the previous APDU in the buffer whether or not it was read by the AR ASE. Reading the APDU from the buffer is not destructive — it does not destroy or change the contents of the buffer, allowing the contents to be read from the buffer one or more times.

One variant of buffered ARs is that they can be dedicated to the FAL User, typically for the publishing of a single variable (or variable list). When they are, the identity of the variable (or variable list) is known by the context. ARs of this type are known as “dedicated”.

4.2.7.9 User-triggered and Network Scheduled Conveyance

Another characteristic of an AREP is when they convey service requests and responses. AREPs that convey them upon submission by the user are called user-triggered. Their conveyance is asynchronous with respect to network operation.

AREPs that convey requests and responses at predefined intervals, regardless of when they are received for transfer are termed network-scheduled. The time-intervals used to regulate conveyance are maintained by the Data Link Layer. Network-scheduled AREPs are capable of indicating when transferred data was submitted late for transmission, or when it was submitted on time, but transmitted late.

Network-scheduled AREPs may also convey requests and responses synchronously using the Data Link compel service if the Data Link layer has available bandwidth. These transfers occur “out-of-band” (that is, outside the cyclic portion of the network schedule) with respect to the scheduled transfers of these ARs.

4.2.7.10 AREP Timeliness

AREPs convey APDUs between applications using the services of the link layer. When the timeliness capabilities are defined for an AREP, the AREP forwards the timeliness indicators provided by the Data Link Layer. These timeliness indicators make it possible for subscribers of published data to determine if the data they are receiving is up-to-date or “stale”. A brief description of these timeliness indicators is given in 4.2.3.3.2.3.

To support these types of timeliness, the publishing AREP establishes a publisher data link connection reflecting the type of timeliness configured for it by management. After connection establishment, the AREP receives user data and submits it to the DLL for transmission, where timeliness procedures are performed. When the Data Link Layer has the opportunity to transmit the data, it transmits the current timeliness status with the data.

At the subscriber AREP, a data link connection is opened to receive published data that reflects the type of timeliness configured for it by management. The Data Link Layer computes the timeliness of received data and then delivers it to the AREP. The data is then delivered to the user AP through the appropriate ASE.

4.2.7.11 Identification of ARs

FAL AEs are accessed through one or more data link addresses. ARs are identified by the data link addresses of the AREPs that participate in the AR.

To simplify AR identification for 1-to-many ARs, group DLSAP addresses are used instead of address lists for AREPs that receive unconfirmed service requests transferred using connectionless data link services. For AREPs that receive unconfirmed service requests transferred using connection-oriented data link services, the DLCEP address of the sending AREP is used to identify the AR; no other address is necessary. Therefore, the addresses used to identify a 1-to-many AR are reduced to either a pair of DLSAP addresses or a single DLCEP address, instead of an enumerated list of addresses.

For the dissemination of information between publishers and subscribers, these techniques permit publishers to be unaware of the identity of the subscribers.

4.2.7.12 Definition and creation of AREPs

AREP definitions specify instances of AREP classes. AREPs may be predefined or they may be defined using the create service if their AE supports this capability.

AREPs may be pre-defined and pre-established, or they may be pre-defined and dynamically established. Figure 11 depicts these two cases. AREPs also may require both dynamic definition (through System Management) and establishment or they may be dynamically defined such that they may be used without any establishment (they are defined in an established state).

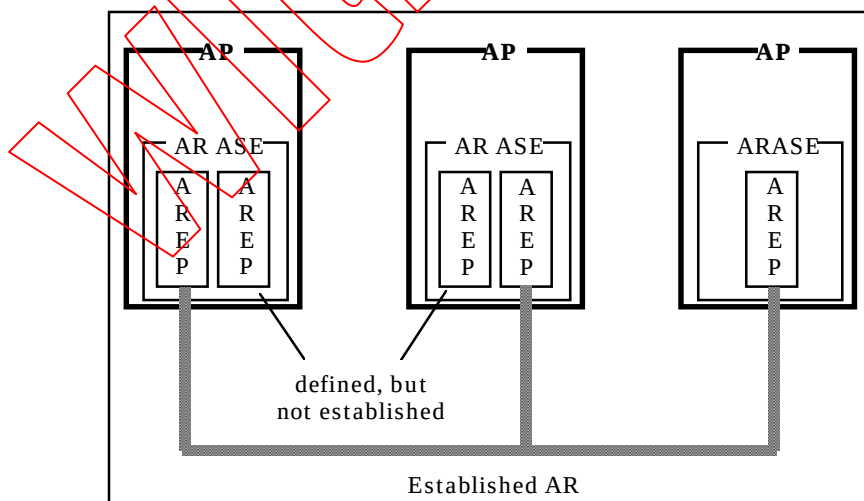


Figure 11 – Defined and Established AREPs

4.2.7.13 AR Establishment and Termination

ARs may be established either before the operational phase of the AP or during its operation. When established during the operation of an AP, the AR Establish service is used. The AR Establish service results in the exchange of AR Establishment APDUs.

Once an AR has been established, an AR may be terminated using the Abort service. The Abort service causes the AR endpoint to be closed immediately without the exchange of APDUs. Instead, the aborted endpoint issues a single abort APDU to the other endpoints participating in the AR that it is no longer capable of sending or receiving APDUs on the AR.

Additionally, an AR can be gracefully terminated, or de-established, using the AR DeEstablish service. The AR DeEstablish service results in the exchange of AR de-establishment APDUs. The definition of its operation is for future study.

4.2.7.14 Reserved ARs

The FAL defines a mechanism for specifying and reserving ARs for a specific purpose. One such AR is specified and reserved for use by system management. Reserved ARs are specified in IEC 61158-6, annex A.

4.3 Fieldbus Application Layer Naming and Addressing

This subclause refines the principles defined in ISO 7498-3 that involve the identification (naming) and location (addressing) of APOs referenced through the Fieldbus Application Layer.

This subclause defines how names and numeric identifiers are used to identify APOs accessible through the FAL. This subclause also specifies how Data Link Layer addresses are used to locate APs in the Fieldbus environment.

4.3.1 Identifying Objects accessed through the FAL

APOs accessed through the FAL are identified independent of their location. That is, if the location of the AP that contains the APO changes, the APO may still be referenced using the same set of identifiers.

Identifiers for APs and APOs within the FAL are defined as key attributes in the class definitions for APOs. Within these APO class definitions, four types of key attributes are used: names, numeric identifiers, symbolic addresses, and numeric addresses.

Identifiers are assigned for the purpose of discriminating among APs and APOs. Therefore, at a given time,

- (1) within the scope of a Fieldbus system, no two APs may have the same name, and
- (2) within the scope of an AP, no two APOs may have the same identifier, and
- (3) within the scope of an AP, an APO may not have the same identifier as its AP.

4.3.1.1 Names

Names are string-oriented identifiers. They are defined to permit APs and APOs to be named within the domain of the system where they are used. Therefore, although the scope of the name of an APO is specific to the AP in which it resides, the assignment of the name is administered within the system in which it is configured.

Names may be descriptive, although they do not have to be. Descriptive names make it possible to provide meaningful information, such as its use, about the object they name.

Names may also be coded. Coded names make it possible to identify an object using a short, compressed form of a name. They are typically simpler to transfer and process, but not as easy to understand as descriptive names.

The names used within the FAL may be either descriptive or coded. Their form and their values are assigned by the FAL user.

4.3.1.2 Numeric Identifiers

Numeric identifiers are identifiers whose values are integers. They are designed for efficient use within the Fieldbus system, and are assigned for efficient access to APOs by their AP. The scope of numeric identifiers is local to the AP.

4.3.1.3 Numeric Addresses

Numeric addresses represent logical memory addresses for APOs of a limited set of classes. This type of identifier is defined to simplify naming of APOs that are normally associated with a specific location in memory.

4.3.1.4 Symbolic Addresses

Symbolic addresses represent logical identifiers for APOs independent of their use or the system where they are configured. They are defined to permit manufacturers of Fieldbus APs to assign identifiers to APOs independent from names assigned within the domain of the system where they are used.

4.3.2 Addressing APs accessed through the FAL

Fieldbus addresses represent the network locations of APs. Addresses relevant to the FAL are DLSAP addresses used to locate the AREPs of an AP.

4.3.2.1 Use of Data Link Layer Addresses

DL-addresses are used by the FAL to locate AR endpoints. These addresses may be DLSAP-addresses or DLCEP-addresses, depending on how the AREP is mapped to the Data Link Layer. An AREP's Data Link Layer mapping defines which capabilities of the Data Link Layer are used to support conveyance of APDUs by the AREP.

DL addresses are not used to identify objects accessed or referenced through the AR endpoint, and they are only used to locate the endpoint, not identify it.

4.3.2.2 Binding of AR Endpoints to Data Link Layer Service Access Points

Data Link Layer addresses locate the Data Link Layer Service Access points (DLSAPs) and connection endpoints (DLCEPs) to which an AR endpoint may be bound. Therefore, an AR endpoint is located by its binding to a single DLSAP or DLCEP that may be identified by one or more DLL-addresses. Only one AR Endpoint may be bound to a specific DLCEP, or source and destination DLSAP pair in the case of connectionless data link operation, at a given time.

4.4 Architecture Summary

The summary of the FAL architecture is presented in this clause. Figure 12 illustrates the major components of the FAL architecture and how they relate to each other.

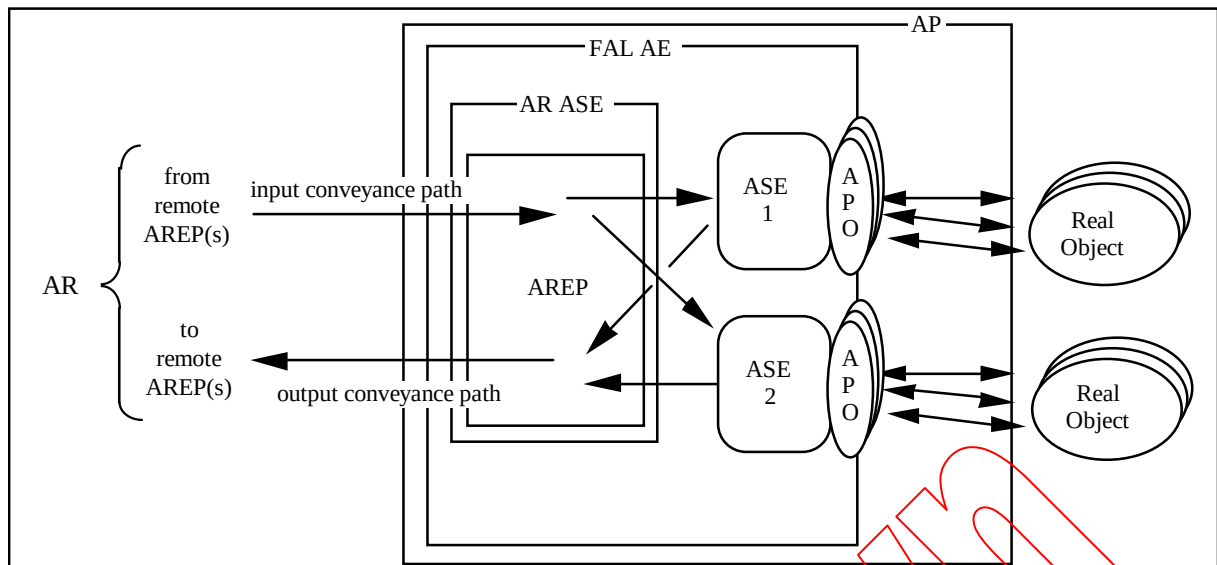


Figure 12 – FAL Architectural Components

Figure 12 depicts an AP that communicates through the FAL AE. The AP represents its internal real objects as APOs for remote access to them. Two ASEs are shown that provide the remote access services to their related APOs. The AR ASE contains a single AREP that conveys service requests and responses for the ASEs to one or more remote AREPs located in remote APs.

4.5 FAL Service Procedures and Time Sequence Diagrams

This subclause contains service procedures for FAL confirmed and unconfirmed services.

4.5.1 FAL Confirmed Service Procedures

FAL confirmed services are capable of operating through queues or buffers.

The requesting user submits a confirmed service request primitive to its FAL AE and starts an associated timer to monitor the request. The appropriate FAL ASE builds the related confirmed service request APDU Body and conveys it on the specified AR using an AR Confirmed Send Service request primitive. It also creates a transaction state machine.

Upon receipt of the confirmed service request APDU Body in an AR Confirmed Send service indication primitive, the responding ASE decodes it. If a protocol error was encountered while decoding the received APDU Body, the ASE uses the AR Abort Service to abort the AR or it internally requests the AP ASE to generate a Reject-PDU (which option is taken is dependent on the AR type). If a protocol error did not occur, the ASE delivers a confirmed service indication primitive to its user.

If the responding user is able to successfully process the request, the user returns a confirmed service response (+) primitive.

If the responding user is unable to successfully process the request, the service fails and the user issues a confirmed service response (-) primitive indicating the reason.

The responding ASE builds a confirmed service response APDU Body for a confirmed service response (+) primitive or a confirmed service error APDU Body for a confirmed service response (-) primitive and conveys it on the specified AR using an AR Confirmed Send service response primitive.

Upon receipt of the returned APDU Body in an AR Confirmed Send service confirmation (+ or -) primitive the initiating ASE delivers a confirmed service confirmation primitive to its user which specifies success or failure, and the reason for failure if a failure occurred. It also cancels the corresponding transaction state machine.

However, if the AP's transaction timer expires before the corresponding response or error APDU is received, the AP may take corrective actions, which may include instructing its local ASE to cancel the transaction state machine. Such instruction to the ASE would occur through a locally defined interface.

4.5.2 Confirmed Service Time Sequence Diagram

Figure 13 illustrates the sequence of confirmed service primitives.

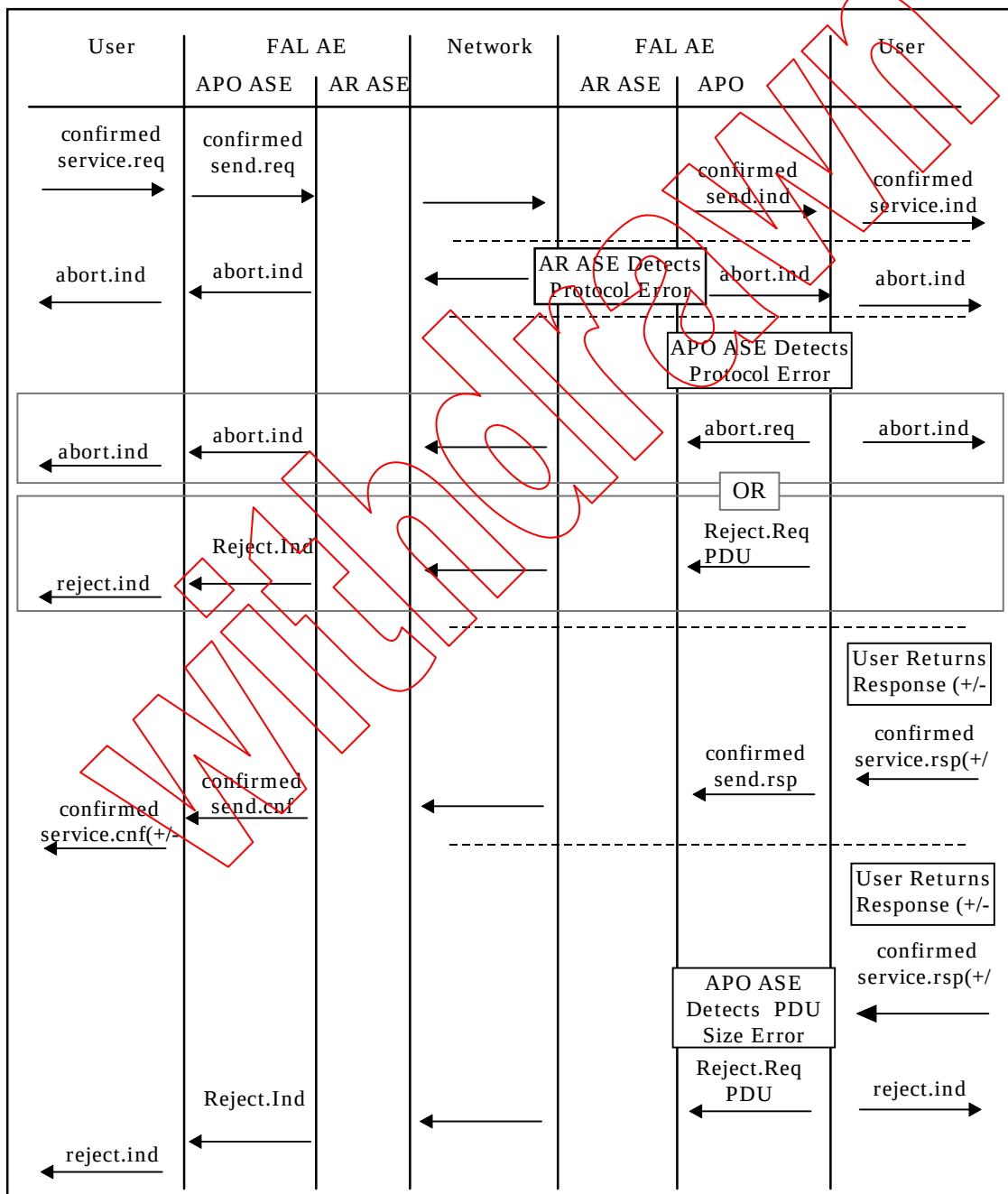


Figure 13 – Time Sequence Diagram For The Confirmed Services

4.5.3 FAL Unconfirmed Service Procedures

FAL unconfirmed services are capable of operating through queues or buffers.

The requesting user submits an unconfirmed service request primitive to its FAL AE. The appropriate FAL ASE builds the related unconfirmed service request APDU Body and conveys it on the specified AR using an AR Unconfirmed Send Service request primitive.

Upon receipt of the unconfirmed request APDU Body in an AR Unconfirmed Send Service indication primitive, the receiving ASE(s) participating in the AR delivers the appropriate unconfirmed service indication primitive to its user. Timeliness parameters are included in the indication primitive if the AR that conveyed the APDU Body supports timeliness.

4.5.4 Unconfirmed Service Time Sequence Diagram

Figure 14 illustrates the sequence of unconfirmed service primitives.

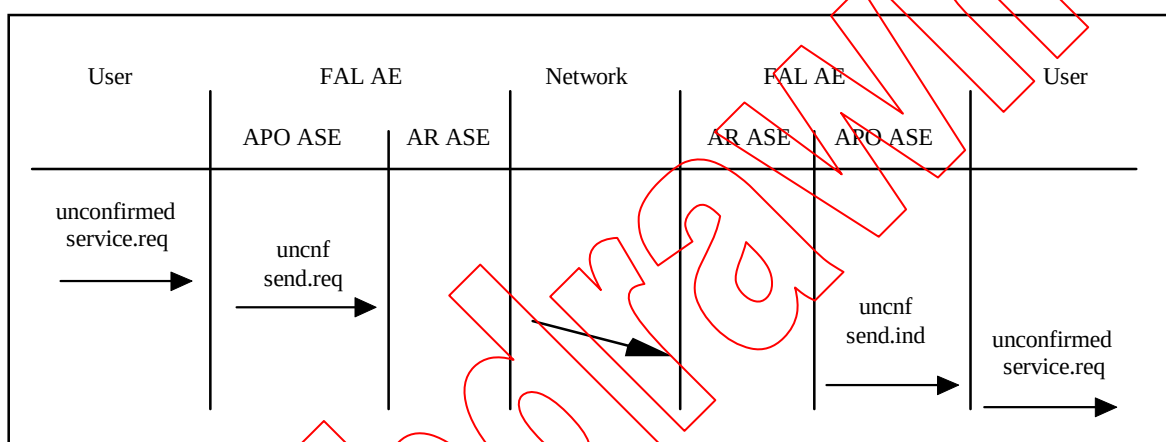


Figure 14 – Time Sequence Diagram For Unconfirmed Services

4.6 Common FAL attributes

The following attributes are defined for all FAL Class specifications of the FAL Classes that follow, except for the Data Type class. Therefore, they are defined here instead of with the other attributes for each of the services.

ATTRIBUTES:

- 1 (o) Key Attribute: Numeric Identifier
- 2 (o) Key Attribute: Name
- 3 (o) Attribute: User Description
- 4 (o) Attribute: Object Revision

Numeric Identifier

This optional key attribute specifies the numeric id of the object. It is used as a shorthand reference by the FAL protocol to identify the object. There are three possibilities for identification purposes: numeric identifier or name or both. This attribute is required for the Data Type model.

Name

This optional key attribute specifies the name of the object. There are three possibilities for identification purposes: numeric identifier or name or both.

User description

This optional attribute contains user defined descriptive information about the object.

Object Revision

This optional attribute specifies the revision level of the object. It is a structured attribute composed of major and minor revision numbers. If Object Revision is supported, it contains both a Major Revision and a Minor Revision with a value range 0 to 15 for each. The use of major/minor fields is intended to provide the following features:

Major Revision

The Major Revision field contains the major revision value for the object. A change to the major revision indicates that interoperability is affected by the change.

Minor Revision

The Minor Revision field contains the minor revision value for the object. A change to the minor revision indicates that interoperability was not affected by the change – that is users of the object will continue to be capable of interoperating with the object when its minor revision is changed, provided that the major revision remains the same.

4.7 Common FAL Service Parameters

In the specifications of the FAL services that follow, the following parameters are used frequently and consistently. Therefore, they are defined here instead of with the other parameters for each of the services.

AREP

This parameter contains sufficient information to locally identify the AREP to be used to convey the service. This parameter may use a key attribute of the AREP to identify the application relationship. When an AREP supports multiple contexts (established using the Initiate service) at the same time, the AREP parameter is extended to identify the context as well as the AREP.

Invoke ID

This parameter identifies this invocation of the service. It is used to associate service requests with responses. Therefore, no two outstanding service invocations may be identified by the same Invoke ID value.

FAL ASE/FAL Class

This parameter specifies the FAL ASE (AP, AR, Variable, Data Type, Event, Function Invocation, Load Region) and the FAL Class within the ASE (e.g. AREP, Variable List, Notifier, Action).

Numeric ID

This parameter is the numeric identifier of the object.

Error Info

This parameter provides error information for service errors. It is returned in confirmed service response(-) primitives. It is composed of the following elements.

Error Class

This parameter indicates the general class of error. Valid values are specified in the definition of Error Code parameter, below.