

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Field Device Integration (FDI) –
Part 109-1: Profiles – HART® and WirelessHART®**

**Intégration des appareils de terrain (FDI) –
Partie 109-1: Profils – HART® et WirelessHART®**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2015 IEC, Geneva, Switzerland

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

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

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

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

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

IEC publications search - www.iec.ch/searchpub

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 60 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

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

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

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

Recherche de publications IEC - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 60 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

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



IEC 62769-109-1

Edition 1.0 2015-05

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Field Device Integration (FDI) –
Part 109-1: Profiles – HART® and WirelessHART®**

**Intégration des appareils de terrain (FDI) –
Partie 109-1: Profils – HART® et WirelessHART®**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100

ISBN 978-2-8322-2638-4

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

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms, definitions, abbreviated terms and acronyms	7
3.1 Terms and definitions.....	7
3.2 Abbreviated terms and acronyms	7
4 Conventions	8
4.1 EDDL syntax.....	8
4.2 XML syntax	8
4.3 Capitalizations	8
5 Profile for CP 9/1 (HART®) or CP 9/2 (WirelessHART®)	9
5.1 General.....	9
5.2 Catalog profile	9
5.2.1 Protocol support file.....	9
5.2.2 CommunicationProfile definition.....	9
5.2.3 Profile device.....	9
5.2.4 Protocol version information	9
5.3 Associating a Package with a CP 9/1 device	10
5.3.1 Device type identification mapping.....	10
5.3.2 Device type revision mapping	10
5.4 Information Model mapping.....	10
5.4.1 ProtocolType definition	10
5.4.2 DeviceType mapping	11
5.4.3 FunctionalGroup Identification definition	11
5.5 Topology elements.....	11
5.5.1 ConnectionPoint definition	11
5.5.2 Communication Device definition	17
5.5.3 Communication service provider definition	18
5.5.4 Network definition	19
5.6 Methods.....	19
5.6.1 Methods for FDI Communication Servers.....	19
5.6.2 Methods for Gateways	23
Annex A (normative) Topology scan schema.....	31
A.1 General.....	31
A.2 IdentificationType	31
A.3 AddressTypeTP	33
A.4 AddressTypeIP	34
A.5 AddressTypeTDMA.....	34
A.6 AddressType.....	35
A.7 ConnectionPointType.....	36
A.8 NetworkType.....	36
A.9 Network	37
Annex B (normative) Transfer service parameters.....	38
B.1 General.....	38

B.2	receiveData	38
B.3	sendData	38
B.4	TransferResultDataT	38
B.5	TransferSendDataT	39
Bibliography		40
Table 1 – CommunicationProfile definition		9
Table 2 – Catalog values for profile devices		9
Table 3 – Protocol Version Information		10
Table 4 – Device type catalog mapping		10
Table 5 – ProtocolType HART definition		10
Table 6 – Inherited DeviceType Property mapping		11
Table 7 – Identification parameters		11
Table 8 – ConnectionPointType HART_TP definition		12
Table 9 – ConnectionPointType HART_IP Definition		15
Table 10 – ConnectionPointType HART_TDMA Definition		16
Table 11 – Method Connect arguments		20
Table 12 – Method Disconnect arguments		20
Table 13 – Method Transfer arguments		21
Table 14 – Method GetPublishedData arguments		22
Table 15 – Method SetAddress arguments		22
Table 16 – Method Connect arguments		24
Table 17 – Method Disconnect arguments		24
Table 18 – Method Transfer arguments		25
Table 19 – Method GetPublishedData arguments		27
Table 20 – Method SetAddress arguments		28
Table A.1 – Attributes of IdentificationT		33
Table A.2 – Elements of AddressTypeTP		33
Table A.3 – Elements of AddressTypeIP		34
Table A.4 – Elements of AddressTypeTDMA		35
Table A.5 – Elements of AddressT		35
Table A.6 – Elements of ConnectionPointT		36
Table A.7 – Elements of NetworkT		36
Table B.1 – Attributes of TransferResultDataT		38
Table B.2 – Attributes of TransferSendDataT		39

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE INTEGRATION (FDI) –

**Part 109-1: Profiles –
HART® and WirelessHART®**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

International Standard IEC 62769-109-1 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

CDV	Report on voting
65E/356/CDV	65E/419/RVC

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 2.

A list of all parts in the IEC 62769 series, published under the general title *Field Device Integration (FDI)*, can be found on the IEC website.

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

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

INTRODUCTION

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents concerning

- a) method for the supplying and installation of device-specific functionalities, see Patent Family DE10357276;
- b) method and device for accessing a functional module of automation system, see Patent Family EP2182418;
- c) methods and apparatus to reduce memory requirements for process control system software applications, see Patent Family US2013232186;
- d) extensible device object model, see Patent Family US12/893,680.

IEC takes no position concerning the evidence, validity and scope of this patent right.

The holders of these patent rights have assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from:

- a) ABB Research Ltd
Claes Ryttoft
Affolterstrasse 4
Zurich, 8050
Switzerland
- b) Phoenix Contact GmbH & Co KG
Intellectual Property, Licenses & Standards
Flachsmarktstrasse 8, 32825 Blomberg
Germany
- c) Fisher Controls International LLC
John Dilger, Emerson Process Management LLLP
301 S. 1st Avenue, Marshalltown, Iowa 50158
USA
- d) Rockwell Automation Technologies, Inc.
1 Allen-Bradley Drive
Mayfield Heights, Ohio 44124
USA

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (<http://patents.iec.ch>) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

FIELD DEVICE INTEGRATION (FDI) –

Part 109-1: Profiles – HART® and WirelessHART®

1 Scope

This part of IEC 62769 specifies an FDI profile of IEC 62769 for IEC 61784-1_CP 9/1 (HART®)¹ and IEC 61784-1_CP 9/2 (WirelessHART®)¹.

2 Normative references

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

IEC 62541-100, *OPC unified architecture – Part 100: Device Interface*

IEC 62769-4:2015, *Field Device Integration (FDI) – Part 4: FDI Packages*

NOTE IEC 62769-4 is technically identical to FDI-2024.

IEC 62769-5, *Field Device Integration (FDI) – Part 5: FDI Information Model*

NOTE IEC 62769-5 is technically identical to FDI-2025.

IEC 62769-7, *Field Device Integration (FDI) – Part 7: Communication Devices*

NOTE IEC 62769-7 is technically identical to FDI-2027.

3 Terms, definitions, abbreviated terms and acronyms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62541-100, IEC 62769-4, IEC 62769-5 and IEC 62769-7 apply.

3.2 Abbreviated terms and acronyms

For the purposes of this document, the following abbreviations apply:

CP	Communication profile (see IEC 61784-1 or IEC 61784-2)
CPF	Communication profile family (see IEC 61784-1 or IEC 61784-2)
EDD	Electronic Device Description (see IEC 61804)
EDDL	Electronic Device Description Language (see IEC 61804)
FDI	Field Device Integration

¹ HART and WirelessHART are the trade names of the non-profit consortium HART Communication Foundation, Austin, Texas, USA. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade names. Use of the trade names requires permission of the trade name holder.

FSK	Frequency-Shift-Keying
HCF	HART Communication Foundation
ID	Identification
IM	Information Model
IP	Internet protocol
PDU	Protocol data unit
PSK	Phase-Shift-Keying
TCP	Transmission Control Protocol (see IETF RFC 793)
UDP	User Datagram Protocol (see IETF RFC 768)
XML	Extended markup language

4 Conventions

4.1 EDDL syntax

This document specifies content for the EDD component that is part of FDI Communication Packages. EDDL syntax uses the font `Courier New`. EDDL syntax is used for method signature, variable, data structure and component declarations.

4.2 XML syntax

XML syntax examples use font `Courier New`. The XML syntax is used to describe XML document schema.

Example: `<xs:simpleType name="ExampleT">`

4.3 Capitalizations

The IEC 62769 series uses capitalized terms to emphasize that these terms have an FDI specific meaning.

Some of these terms using an acronym as a prefix, for example

- FDI Client, or
- FDI Server.

Some of these terms are compound terms such as:

- FDI Communication Servers, or
- Profile Package.

Parameter names or attributes are concatenated to a single term, where the original terms start in this term with a capital letter such as:

- ProtocolSupportFile, or
- ProtocolType.

Parameter names or attributes can also be constructed by using an underscore character to concatenate two or more terms such as:

- PROFILE_ID, or
- HART_Network.

5 Profile for CP 9/1 (HART®) or CP 9/2 (WirelessHART®)

5.1 General

This profile document to the FDI specification in IEC 62769 specifies the protocol specifics needed for FDI Packages describing FDI Communication Servers, gateways and devices.

5.2 Catalog profile

5.2.1 Protocol support file

No additional file is required for CP 9/1 or CP 9/2 FDI Device Packages.

5.2.2 CommunicationProfile definition

IEC 62769-4 defines a CommunicationProfileT enumeration type for the Catalog XML schema. Table 1 defines the CP 9/1 specific values for this enumeration.

Table 1 – CommunicationProfile definition

CommunicationProfile	Description
hart_fsk	CP 9/1 device type that supports an FSK physical layer (Frequency-Shift-Keying on a pair of wires).
hart_psk	CP 9/1 device type that supports a PSK physical layer (Phase-Shift-Keying on a pair of wires). Devices supporting PSK are required to also inherently support FSK, and therefore PSK will always be used only in combination with at least FSK.
hart_wirelesshart	CP 9/2 device type that supports a wireless physical layer (communication between device and gateway).
hart_ip	CP 9/1 device type that supports Internet Protocol (these devices support both TCP and UDP).
hart_rs485	CP 9/1 device type that supports EIA-485 digital communication.
hart_ir	CP 9/1 device type that supports an Infrared physical layer (designed to be transparent to FSK masters – included only as information to indicate that the device supports IR connection).
NOTE It is possible for a single CP 9/1 device to support more than one CP.	

5.2.3 Profile device

A Profile Package shall provide the catalog values for profile devices, enabling the FDI Server to leverage a generic device description, if a specific one is not available. The definitions in Table 2 focus on catalog content that is vendor independent.

Table 2 – Catalog values for profile devices

Element	Attribute	Content
PackageType	—	Profile
DeviceModel	—	Empty
Manufacturer	—	Empty

5.2.4 Protocol version information

IEC 62769-4 defines an element type named InterfaceT for the Catalog XML Schema. The element type InterfaceT contains an element named Version which is supposed to provide version information about the applied communication protocol profile. The value has to follow the IEC 62769-4 defined version information schema defined in element type VersionT. Subclause 5.2.4 describes how to apply the currently known protocol versions for CP 9/1 or CP 9/2 entries in the device catalog. The general rule is to use the Universal Revision of the

protocol for the major version part of VersionT, and the value “0” for the minor version and build parts. Table 3 shows the Protocol Version Information.

Table 3 – Protocol Version Information

Protocol Version	InterfaceT Version value
HART Universal Revision 5	5.0.0
HART Universal Revision 6	6.0.0
HART Universal Revision 7	7.0.0
The Protocol Version defined in a package is provided for informational purposes only, and shall not be used to determine the compatibility or applicability of a package to a device.	

5.3 Associating a Package with a CP 9/1 device

5.3.1 Device type identification mapping

CP 9/1 device types are uniquely identified by parameters Manufacturer, Model and DeviceRevision. These parameters are used to associate a given device instance to an FDI Device Package. These parameters are mapped to the FDI Device Package Catalog according to Table 4.

Table 4 – Device type catalog mapping

Catalog element	CP mapping (see 5.4.2)
Manufacturer element of InterfaceT (IEC 62769-4:2015, Clause E.11)	Manufacturer String format “0xddd” where dddd is the Manufacturer number in hexadecimal format.
DeviceModel element of InterfaceT (IEC 62769-4:2015, Clause E.11)	Model String format “0xddd” where dddd is the Model number in hexadecimal format.
DeviceRevision element ListOfSupportedDeviceRevisionsT (IEC 62769-4:2015, Clause E.21)	DeviceRevision String format “x.0.0” where x is the DeviceRevision in decimal format (no leading zeros).

5.3.2 Device type revision mapping

Each device type is identified as per 5.3.1. If a package with matching DeviceRevision is not available, any CP 9/1 FDI package for a corresponding manufacturer and model shall always be compatible with a field device as long as the device revision of the field device is equal to or greater than the device revision specified in the FDI package.

5.4 Information Model mapping

5.4.1 ProtocolType definition

Table 5 defines the ProtocolType used to identify CP 9/1 network communications.

Table 5 – ProtocolType HART definition

Attribute	Value				
BrowseName	HART				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Inherits the properties of ProtocolType defined in IEC 62541-100.					

5.4.2 DeviceType mapping

Each device type inherits the properties of DeviceType. The mapping of the inherited properties from DeviceType is defined in Table 6.

Table 6 – Inherited DeviceType Property mapping

Property	Foundation mapping
SerialNumber	3-byte unique ID of a device, returned in bytes 9 to 11 of Command 0 or Command 11 or Command 21
RevisionCounter	2-byte configuration change counter, returned in bytes 14 and 15 of Command 0 or Command 11 or Command 21. -1 (not defined) for HART revision 5 devices
Manufacturer	For HART revision 7 or higher devices: 2-byte manufacturer code of a device, returned in bytes 17 and 18 of Command 0 or Command 11 or Command 21. For HART revision 6 or lower devices: Most significant byte shall be fixed to 0, and the least significant byte is returned in byte 1 of Command 0 or Command 11 or Command 21
Model	2-byte extended device type of a device, returned in bytes 1 and 2 of Command 0 or Command 11 or Command 21
DeviceManual	Entry text string (not supported) ^a
DeviceRevision	1-byte device revision level of a device, returned in byte 5 of Command 0 or Command 11 or Command 21
SoftwareRevision	1-byte software revision level of a device, returned in byte 6 of Command 0 or Command 11 or Command 21
HardwareRevision	1-byte hardware revision level of a device, returned in byte 7 (only 5 most significant bits) of Command 0 or Command 11 or Command 21
^a Device manuals are exposed as attachments of the FDI Device Package.	

5.4.3 FunctionalGroup Identification definition

As defined in IEC 62541-100, each device representation in the FDI Server hosted Information Model shall contain a protocol specific FunctionalGroup called Identification. This FunctionalGroup organizes variables found in the device type instance. The FunctionalGroup Identification for CP 9/1 is defined in Table 7.

Table 7 – Identification parameters

BrowseName	DataType	Optional/Mandatory
MANUFACTURER_ID	UInt16	Mandatory
DEVICE_TYPE	UInt16	Mandatory
DEVICE_REVISION	UInt8	Mandatory
UNIVERSAL_REVISION	UInt8	Optional
SERIAL_NUMBER	UInt24	Optional
HARDWARE_REVISION	UInt8	Optional
SOFTWARE_REVISION	UInt8	Optional
REVISION_COUNTER	UInt16	Optional

5.5 Topology elements

5.5.1 ConnectionPoint definition

5.5.1.1 General

CP 9/1 devices can support up to five different ConnectionPoint types that are used for network communications.

5.5.1.2 HART_TP5, HART_TP6, HART_TP7

The ConnectionPoint types HART_TP5, HART_TP6, and HART_TP7 shall be used to identify CP 9/1 token passing network communication and are defined in Table 8. HART_TP5, HART_TP6, and HART_TP7 all contain the same properties, but each provides different qualification information for some of the properties (described below). The Protocol Version (UNIVERSAL_REVISION) described in 5.2.3 can be used as an aid to determine which of the three token passing Connection Point types is the most appropriate. CP 9/1 token passing communications can be used on a variety of physical layers. FSK, PSK, RS485, and Infrared physical layer connections shall all use the HART_TP connection type. The ConnectionPoint types HART_TP5, HART_TP6, and HART_TP7 are subtypes of abstract type ConnectionPointType defined in IEC 62769-5.

The DevAddr property shall be the long address (5 bytes) for the device, and is the only parameter necessary to communicate with the field device.

The DevMfg property shall be the 2-byte Manufacturer ID, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevType property shall be the 2-byte extended device type, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevRev property shall be the device revision, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevTag property shall be the long tag for HART[®] protocol version 6 or 7 devices. The DevTag property shall be the tag for protocol version 5 devices. The DevTag property can be used to help automate the process of assigning live devices in the scan list to offline placeholders. HART_TP5 Connection Points shall limit the DevTag to 8 characters in length. HART_TP6 and HART_TP7 Connection Points shall limit the DevTag to 32 characters in length.

The DevPollAddr property shall be the poll address, and can be used to identify which device is located at a specific poll address. HART_TP5 Connection Points shall be limited to values between 0 and 15 for the DevPollAddr property. HART_TP6 Connection Points shall be limited to values between 0 and 31 for the DevPollAddr property. HART_TP7 Connection Points shall be limited to values between 0 and 63 for the DevPollAddr property.

For forward compatibility, a lower revision HART_TP Connection Point is compatible and can be used for a higher universal revision device connection. For example, if a future HART universal revision 8 device is encountered, and no HART_TP8 is available in the FDI server, HART_TP7 will be compatible and shall be used to connect to the device. If the Protocol Version (i.e. the Universal Revision) is unknown for any reason, the HART_TP5 Connection Point can be used, and will be forward compatible to later universal revisions.

Table 8 – ConnectionPointType HART_TP definition

Attribute	Value				
BrowseName	ConnectionPoint_HART_TP5 or ConnectionPoint_HART_TP6 or ConnectionPoint_HART_TP7				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Inherits the properties of ConnectionPointType defined in IEC 62769-5.					
HasProperty	Variable	DevAddr	UInt40	PropertyType	Mandatory
HasProperty	Variable	DevMfg	UInt16	PropertyType	Optional
HasProperty	Variable	DevType	UInt16	PropertyType	Optional
HasProperty	Variable	DevRev	UInt16	PropertyType	Optional
HasProperty	Variable	DevTag	String	PropertyType	Optional
HasProperty	Variable	DevPollAddr	UInt8	PropertyType	Optional

The ConnectionPoint type HART_TP5, HART_TP6, and HART_TP7 shall be described by an EDD element contained in a Communication Device related FDI Package that can drive a CP 9/1 network. Actual ConnectionPoint properties are declared by VARIABLE constructs grouped together in a COLLECTION named ConnectionPoint_HART_TP5, ConnectionPoint_HART_TP6, or ConnectionPoint_HART_TP7. The following EDDL source code is an example describing a TP5 Connection Point.

```
COMPONENT ConnectionPoint_HART_TP5
{
    LABEL "HART TP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_TP5;
}

VARIABLE DevAddr
{
    LABEL "Address";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(5);
    HANDLING READ & WRITE;
}

VARIABLE DevMfg
{
    LABEL "Manufacturer";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevType
{
    LABEL "Device Type";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevRev
{
    LABEL "Device Revision";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER;
    HANDLING READ & WRITE;
}

VARIABLE DevTag
{
    LABEL "Tag";
    CLASS DEVICE;
    TYPE ASCII(32);
    HANDLING READ & WRITE;
}
```

```

VARIABLE DevPollAddr
{
    LABEL "Poll Address";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER
    {
        MAX_VALUE 15; //Define appropriate max value for various revisions
    }
    HANDLING READ & WRITE;
}

COLLECTION ConnectionPoint_TP5
{
    LABEL "Connection Point";
    MEMBERS
    {
        ADDRESS,    DevAddr,    "Device Address";
        MFG,        DevMfg,     "Manufacturer";
        DEV_TYPE,   DevType,    "Device Type";
        DEV_REV,    DevRev,     "Device Revision";
        TAG,        DevTag,     "Device Tag";
        POLL_ADDR, DevPollAddr, "Poll Address";
    }
}

```

5.5.1.3 HART_IP

The ConnectionPoint type HART_IP shall be used to identify CP 9/1 IP network communication and is defined in Table 9. HART_IP communications can be used on a variety of physical layers. Ethernet connections shall all use the HART_IP connection type. Additional physical layers developed in the future may also use the HART_IP connection type. The ConnectionPoint type HART_IP is a subtype of abstract type ConnectionPointType defined in IEC 62769-5.

The IPAddress property shall indicate the IP Address (up to 16 bytes) used for the connection.

The IPVersion property shall indicate the version of IP used for the connection (either 4 or 6).

The IPPort property shall be the IP port number for the connection. The default port number used for HART IP is 5 094.

The DevAddr property shall be the long address (5 bytes) for the device.

The DevMfg property shall be the 2-byte Manufacturer ID, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevType property shall be the 2-byte extended device type, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevRev property shall be the device revision, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevTag property shall be the long tag (with maximum 32 characters), and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

Table 9 – ConnectionPointType HART_IP Definition

Attribute	Value				
BrowseName	ConnnectionPoint_HART_IP				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Inherits the properties of ConnectionPointType defined in IEC 62769-5					
HasProperty	Variable	IPAddress	ByteString	PropertyType	Mandatory
HasProperty	Variable	IPVersion	UInt8	PropertyType	Mandatory
HasProperty	Variable	IPPort	UInt16	PropertyType	Mandatory
HasProperty	Variable	DevAddr	UInt40	PropertyType	Mandatory
HasProperty	Variable	DevMfg	UInt16	PropertyType	Optional
HasProperty	Variable	DevType	UInt16	PropertyType	Optional
HasProperty	Variable	DevRev	UInt16	PropertyType	Optional
HasProperty	Variable	DevTag	String	PropertyType	Optional

The ConnectionPoint type HART_IP shall be described by an EDD element contained in a Communication Device related FDI Package that can drive a CP 9/1 network. Actual ConnectionPoint properties are declared by VARIABLE constructs grouped together in a COLLECTION named ConnectionPoint_HART_IP. The following EDDL source code is an example describing an IP Connection Point.

```

COMPONENT ConnectionPoint_HART_IP
{
    LABEL "HART IP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_IP;
}

ARRAY IPAddress
{
    LABEL "IP Address";
    CLASS DEVICE;
    TYPE OCTET(16);
    HANDLING READ & WRITE;
}

VARIABLE IPVersion
{
    LABEL "IP Version";
    CLASS DEVICE;
    TYPE ENUMERATED
    {
        { 4, "IPv4" },
        { 6, "IPv6" }
    }
    HANDLING READ & WRITE;
}

VARIABLE IPPort
{
    LABEL "IP Port";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER (2);
    DEFAULT_VALUE 5 094;
    HANDLING READ & WRITE;
}

```

```

COLLECTION ConnectionPoint_IP
{
    LABEL "Connection Point";
    MEMBERS
    {
        IPADDRESS, IPAddress, "IP Address";
        IPVERSION, IPVersion, "IP Version";
        IPPORT,      IPPort,      "IP Port";
        ADDRESS,     DevAddr,     "Device Address";
        MFG,          DevMfg,      "Manufacturer";
        DEV_TYPE,     DevType,     "Device Type";
        DEV_REV,      DevRev,      "Device Revision";
        TAG,          DevTag,      "Device Tag";
    }
}

```

5.5.1.4 HART_TDMA

The ConnectionPoint type HART_TDMA shall be used to identify CP 9/2 time division media access network communication and is defined in Table 10. HART_TDMA communications can be used on a variety of physical layers. The ConnectionPoint type HART_TDMA is a subtype of abstract type ConnectionPointType defined in IEC 62769-5. WirelessHART connections shall all use the HART_TDMA connection type. Additional physical layers developed in the future may also use the HART_TDMA connection type.

The Network property shall be the network ID for the network.

The DevAddr property shall be the long address (5 bytes) for the device.

The DevMfg property shall be the 2-byte Manufacturer ID, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevType property shall be the 2-byte extended device type, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevRev property shall be the device revision, and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

The DevTag property shall be the long tag (with maximum 32 characters), and can be used to help automate the process of assigning live devices in the scan list to offline placeholders.

Table 10 – ConnectionPointType HART_TDMA Definition

Attribute	Value				
BrowseName	ConnectionPoint_HART_TDMA				
IsAbstract	False				
References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
Inherits the properties of ConnectionPointType defined in IEC 62769-5.					
HasProperty	Variable	Network	UInt16	PropertyType	Mandatory
HasProperty	Variable	DevAddr	UInt40	PropertyType	Mandatory
HasProperty	Variable	DevMfg	UInt16	PropertyType	Optional
HasProperty	Variable	DevType	UInt16	PropertyType	Optional
HasProperty	Variable	DevRev	UInt16	PropertyType	Optional
HasProperty	Variable	DevTag	String	PropertyType	Optional

The ConnectionPoint type HART_TDMA shall be described by an EDD element contained in a Communication Device related FDI Package that can drive a CP 9/2 network. Actual ConnectionPoint properties are declared by VARIABLE constructs grouped together in a COLLECTION named ConnectionPoint_HART_TDMA. The following EDDL source code is an example describing a TDMA Connection Point.

```

COMPONENT ConnectionPoint_HART_TDMA
{
    LABEL "HART TDMA Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_TDMA;
}

VARIABLE Network
{
    LABEL "Network ID";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER (2);
    HANDLING READ & WRITE;
}

COLLECTION ConnectionPoint_TDMA
{
    LABEL "Connection Point";
    MEMBERS
    {
        NETWORK,    Network, "Network ID";
        ADDRESS,    DevAddr, "Device Address";
        MFG,         DevMfg,  "Manufacturer";
        DEV_TYPE,    DevType, "Device Type";
        DEV_REV,     DevRev,  "Device Revision";
        TAG,         DevTag,  "Device Tag";
    }
}

```

5.5.2 Communication Device definition

According to IEC 62769-7 each FDI Communication Package shall contain an EDD element describing the communication device. The following EDDL source code is an example describing an FDI Communication Server.

```

COMPONENT CommunicationServer_HART
{
    LABEL "HART Communication Server";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { Communication_Device_Setup_HART}
}

COMPONENT_RELATION Communication_Device_Setup_HART
{
    LABEL "Relation between Device and Communication Device";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        CommunicationDevice_HART{AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 4;
}

```

Semantics of the EDDL constructs shown above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type CommunicationServerType as described in IEC 62769-7.

According to IEC 62769-7 each FDI Communication Package shall contain at least one EDD element describing at least one communication device component. The following EDDL source code is an example for a CP 9/1 communication device.

```
COMPONENT CommunicationDevice_HART
{
    LABEL "HART Communication Device";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { ServiceProviderRelation_HART }
    BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION ServiceProviderRelation_HART
{
    LABEL "Relation to communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        CommunicationServiceProvider_HART {AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}
```

Semantics of the EDDL constructs shown above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type CommunicationServerChannelType as described in IEC 62769-7.

5.5.3 Communication service provider definition

According to IEC 62769-7 each FDI Communication Package shall contain at least one EDD element describing at least one communication service provider component. The following EDDL source code is an example for a CP 9/1 communication service provider component.

The component reference (ConnectionPoint_HART_IP) corresponds to one of the related Connection Point definitions in 5.5.1. The attribute BYTE_ORDER value is to be set according to the protocol.

```
COMPONENT CommunicationServiceProvider_HART
{
    LABEL "HART Communication Service Provider";
    CAN_DELETE FALSE;
    CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
    COMPONENT_RELATIONS { ServiceProviderConnectionPointRelation_HART }
    BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION ServiceProviderConnectionPointRelation_HART
{
    LABEL "Relation between communication service and connection point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING
    {
        IPAddress
    }
    COMPONENTS
    {
        ConnectionPoint_HART_IP { AUTO_CREATE 1; }
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}
```

Semantics of the EDDL constructs shown above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type *CommunicationServiceType* as described in IEC 62769-7.

5.5.4 Network definition

According to IEC 62769-7 each FDI Communication Package shall contain at least one EDDL element describing network configuration constraints using the component construct. The following EDDL source code is an example describing a network.

```
COMPONENT Network_HART
{
    LABEL "HART Network";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK;
    COMPONENT_RELATIONS { NetworkConnectionPointRelation_HART }
}

COMPONENT_RELATION NetworkConnectionPointRelation_HART
{
    LABEL "Relation between network and connection point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {DevPollAddr}
    COMPONENTS
    {
        ConnectionPoint_HART_TP5
        {
            MAXIMUM_NUMBER 16;
        },
        ConnectionPoint_HART_TP6
        {
            MAXIMUM_NUMBER 32;
        },
        ConnectionPoint_HART_TP7
        {
            MAXIMUM_NUMBER 64;
        }
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 64;    //Defined to limitations of the comm. device
}
```

Semantics of the EDDL constructs shown above are described in IEC 62769-7. The EDDL COMPONENT will be utilized by the FDI server and FDI Communication Server to create an instance of type *NetworkType* as described in IEC 62769-5. Some communication devices may organize multiple IO cards as individual networks, while others may present a single network that comprises all Connection Points across all IO cards.

5.6 Methods

5.6.1 Methods for FDI Communication Servers

5.6.1.1 General

The FDI Communication Server shall implement services according to the method signatures described in 5.6.1 and according to the Information Model .

5.6.1.2 Connect

Table 11 shows the Method Connect arguments.

Signature:

```
Connect (
    [in]  ByteString CommunicationRelationId,
    [in]  ByteString longAddress,
    [out] Int32      ServiceError);
```

Table 11 – Method Connect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the Device ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
LongAddress	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 5.5.1. The argument holds the device's long (5-byte) address. The value can be obtained by the method Scan.
ServiceError	0: OK / execution finished, connection established successfully -1: Connect Failed / canceled by caller -2: Call Failed / unknown service ID -3: Connect Failed / device not found -4: Connect Failed / invalid device node address -5: Connect Failed / invalid device identification
NOTE 1 IEC 62769-7 defines the argument AddressData of the Connect Method as an array of Variant. The LongAddress arguments defined in the table are represented as entries of the Variant array in the order they are specified above.	
NOTE 2 IEC 62769-7 defines the argument DeviceInformation as a protocol specific argument list in which the Connect method stores the resulting data. The DeviceInformation argument is defined as an array of Variant. The DeviceInformation argument is not used for CP 9/1 devices.	

5.6.1.3 Disconnect

Table 12 shows the Method Disconnect arguments.

Signature:

```
Disconnect (
    [in]  ByteString CommunicationRelationId,
    [out] Int32      ServiceError);
```

Table 12 – Method Disconnect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the Device ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
ServiceError	0: OK / disconnect finished successfully -1: Disconnect Failed / no existing communication relation -2: Disconnect Failed / invalid communication relation identifier

5.6.1.4 Transfer

Table 13 shows the Method Transfer arguments.

Signature:

```

Transfer (
    [in]  ByteString CommunicationRelationId,
    [in]  UInt16      Command,
    [in]  ByteString Request,
    [out] ByteString Reply,
    [out] Int32       ServiceError);

```

Table 13 – Method Transfer arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
Command	Command number of the CP 9/1 command to be sent. The FDI Communication Server is required to generate and send the PDU in the proper format according to CP 9/1 protocol specifications (e.g. taking care of command expansion).
Request	An array of bytes representing the data bytes in the CP 9/1 command request (e.g. starting after the byte count, and ending before the checksum).
Reply	An array of bytes representing the data bytes in the CP 9/1 command response (e.g. starting after the byte count, and ending before the checksum).
ServiceError	0: OK / execution finished, Reply contains the result -1: Transfer Failed / cancelled by caller -2: Call Failed / unknown service ID -3: Transfer Failed / no existing communication relation. -4: Transfer Failed / invalid communication relation identifier -5: Transfer Failed / invalid Request content -6: Transfer Failed / invalid Reply format
NOTE 1 IEC 62769-7 defines the argument SendData of the Transfer Method as an array of Variant. The arguments Command and Request defined in the table are represented as entries of the Variant array in the order they are specified above.	
NOTE 2 IEC 62769-7 defines the argument ReceiveData of the Transfer Method as an array of Variant. The argument Reply defined in the table is represented as the entry of the Variant array.	

5.6.1.5 GetPublishedData

CP 9/1 burst and event notification messages represent unsolicited messages as defined in IEC 62769-7. Table 14 shows the Method GetPublishedData arguments.

Signature:

```

GetPublishedData (
    [in]  ByteString CommunicationRelationId,
    [out] UInt16      Command,
    [out] ByteString Reply,
    [out] DateTime     TimeStamp
    [out] Int32       ServiceError);

```

Table 14 – Method GetPublishedData arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network within the Information Model.
Command	Command number of the CP 9/1 command published. The FDI Communication Server is required to parse the PDU in various formats according to CP 9/1 Protocol standards (e.g. taking care of command expansion).
Reply	An array of bytes representing the data bytes in the published CP 9/1 command response (e.g. starting after the byte count, and ending before the checksum).
TimeStamp	Time at which the published data was captured.
ServiceError	0: OK / execution finished, Reply contains the result -1: GetPublishedData Failed / cancelled by caller -2: Call Failed / unknown service ID -3: GetPublishedData Failed / not supported -4: GetPublishedData Failed / no existing communication relation -5: GetPublishedData Failed / invalid communication relation identifier -8: GetPublishedData Failed / no burst / event data published
NOTE IEC 62769-7 defines the argument ReceiveData of the GetPublishedData Method as an array of Variant. The arguments Command and Reply defined in the table are represented as entries of the Variant array in the order they are specified above.	

5.6.1.6 SetAddress

Table 15 shows the Method SetAddress arguments.

Signature

```
SetAddress (
    [in] UInt8 OldPollAddress,
    [in] UInt8 NewPollAddress,
    [out] Int32 ServiceError);
```

Table 15 – Method SetAddress arguments

Argument	Description
OldPollAddress	The argument value holds the current address of a device. Allowed values are 0..63.
NewPollAddress	The argument value holds the new address for a device. Allowed values are 0..63
ServiceError	0: OK / execution finished successfully -1: SetAddress Failed / cancelled by caller -2: Call Failed / unknown service ID -3: SetAddress Failed / not initialized -4: SetAddress Failed / not connected to a network -5: SetAddress Failed / no device found responding to oldAddress -6: SetAddress Failed / duplicate address error -7: SetAddress Failed / device did not accept new address -8: SetAddress Failed / invalid oldAddress (in terms of syntax, data type, data format, and so on)

Argument	Description
	-9: SetAddress Failed / invalid newAddress (in terms of syntax, data type, data format, and so on)
	-10: SetAddress Failed / not possible in status connected
NOTE 1 IEC 62769-7 defines the argument OldAddress of the SetAddress Method as an array of Variant. The argument OldPollAddress defined in the table is represented as the entry of the Variant array.	
NOTE 2 IEC 62769-7 defines the argument NewAddress of the SetAddress Method as an array of Variant. The argument NewPollAddress defined in the table is represented as the entry of the Variant array.	

5.6.1.7 Scan

The method signature specified in IEC 62769-7 applies. The corresponding topologyScanResult schema is specified in Annex A.

5.6.1.8 ResetScan

The method signature specified in IEC 62769-7 applies. The corresponding topologyScanResult schema is specified in Annex A.

5.6.2 Methods for Gateways

5.6.2.1 General

The method signatures described in 5.6.2 shall be implemented in the EDD element IEC 62769-4 contained in a Gateway related FDI Package containing the communication device definitions.

5.6.2.2 Connect

Table 16 shows the Method Connect arguments.

Signature:

```
METHOD BeginConnect(
    DD_String      CommunicationRelationId,
    DD_String      longAddress,
    unsigned long   ServiceID,
    unsigned long   &DelayForNextCall,
    long            &ServiceError)
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD EndConnect(
    DD_String      CommunicationRelationId,
    unsigned long   ServiceID,
    unsigned long   &DelayForNextCall,
    long            &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD CancelConnect(
    DD_String      CommunicationRelationId,
    unsigned long   ServiceID,
    long            &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

Table 16 – Method Connect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
LongAddress	The argument name shall match with the corresponding attribute name defined for the ConnectionPoint which is described by a corresponding EDD element specified in 5.5.1. The argument holds the device's long (5-byte) address. The value can be obtained by the method Scan.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndConnect invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndConnect 0: OK / execution finished, connection established successfully -1: Connect Failed / cancelled by caller -2: Call Failed / unknown service ID -3: Connect Failed / device not found -4: Connect Failed / invalid device address -5: Connect Failed / invalid device identification

5.6.2.3 Disconnect

Table 17 shows the Method Disconnect arguments.

Signature:

```

METHOD Disconnect(
    DD_String  communicationRelationId,
    Long       &serviceError)
{
    DEFINITION {<Gateway specific implementation>}
}

```

Table 17 – Method Disconnect arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
ServiceError	0: OK / disconnect finished successfully -1: Disconnect Failed / no existing communication relation -2: Disconnect Failed / invalid communication relation identifier

5.6.2.4 Transfer

Table 18 shows the Method Transfer arguments.

Signature:

```

METHOD BeginTransfer(
    DD_String      CommunicationRelationId,
    unsigned short Command,
    DD_String      Request,
    DD_String      &Reply,
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD EndTransfer(
    DD_String      CommunicationRelationId,
    DD_String      &Reply,
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD CancelTransfer(
    DD_String      CommunicationRelationId,
    unsigned long   ServiceId,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

```

Table 18 – Method Transfer arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network which is directly connected to the FDI Communication Server hardware. The nodeId allows finding the direct parent-child relation.
Command	Command number of the CP 9/1 command to be sent. The FDI Communication Server is required to generate and send the PDU in the proper format according to CP 9/1 protocol specifications (e.g. taking care of command expansion).
Request	An array of bytes representing the data bytes in the CP 9/1 command request (e.g. starting after the byte count, and ending before the checksum).
Reply	An array of bytes representing the data bytes in the CP 9/1 command response (e.g. starting after the byte count, and ending before the checksum).
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndTransfer invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndTransfer 0: OK / execution finished, REPLY contains the result -1: Transfer Failed / cancelled by caller -2: Call Failed / unknown service ID -3: Transfer Failed / no existing communication relation. -4: Transfer Failed / invalid communication relation identifier -5: Transfer Failed / invalid REQUEST content

Argument	Description
	-6: Transfer Failed / invalid REPLY format

5.6.2.5 GetPublishedData

CP 9/1 burst and event notification messages represent unsolicited messages as defined in IEC 62769-7. Table 19 shows the Method GetPublishedData arguments.

Signature:

```
METHOD BeginGetPublishedData(
    DD_String      CommunicationRelationId,
    unsigned short &Command,
    DD_String      &Reply,
    TIME           &TimeStamp
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD EndGetPublishedData(
    DD_String      CommunicationRelationId,
    unsigned short &Command,
    DD_STRING      &Reply,
    TIME           &TimeStamp
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD CancelGetPublishedData(
    DD_String      CommunicationRelationId,
    unsigned long   ServiceId,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

Table 19 – Method GetPublishedData arguments

Argument	Description
CommunicationRelationId	The argument value contains the nodeId of the ConnectionPoint representing the connection between a device and a physical network within the Information Model.
Command	Command number of the CP 9/1 command published. The FDI Communication Server is required to parse the PDU in various formats according to CP 9/1 Protocol standards (e.g. taking care of command expansion).
Reply	An array of bytes representing the data bytes in the published CP 9/1 command response (e.g. starting after the byte count, and ending before the checksum).
TimeStamp	Time at which the published data was captured.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndGetPublishedData invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndGetPublishedData 0: OK / execution finished, ReceivedData and TimeStamp contains the result -1: GetPublishedData Failed / cancelled by caller -2: Call Failed / unknown service ID -3: GetPublishedData Failed / not supported -4: GetPublishedData Failed / no existing communication relation -5: GetPublishedData Failed / invalid communication relation identifier -8: GetPublishedData Failed / no burst information published

5.6.2.6 SetAddress

Table 20 shows the Method SetAddress arguments.

Signature:

```
METHOD BeginSetAddress(
    unsigned char OldPollAddress,
    unsigned char NewPollAddress,
    unsigned long ServiceId,
    unsigned long &DelayForNextCall,
    long          &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD EndSetAddress(
    unsigned long ServiceId,
    unsigned long &DelayForNextCall,
    long          &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD BeginSetAddress(
    unsigned long ServiceId,
    long          &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

Table 20 – Method SetAddress arguments

Argument	Description
OldPollAddress	The argument value holds the current address of a device. The allowed values are 0..63.
NewPollAddress	The argument value holds the new address for a device. The allowed values are 0..63.
ServiceId	The service transaction code establishes the relation between the service request and the corresponding response.
DelayForNextCall	The value specifies a delay time in ms to limit the EndTransfer invocation cycle that shall not be faster than specified in the argument value.
ServiceError	1: OK / function started asynchronously, result has to be polled with EndSetAddress 0: OK / execution finished successfully -1: SetAddress Failed / cancelled by caller -2: Call Failed / unknown service ID -3: SetAddress Failed / not initialized -4: SetAddress Failed / not connected to a network -5: SetAddress Failed / no device found responding to oldAddress -6: SetAddress Failed / duplicate address error -7: SetAddress Failed / device did not accept new address -8: SetAddress Failed / invalid oldAddress (in terms of syntax, data type, data format, and so on) -9: SetAddress Failed / invalid newAddress (in terms of syntax, data type, data format, and so on) -10: SetAddress Failed /not possible in status connected

5.6.2.7 Scan

The method signature specified in IEC 62769-7 applies. The HART Gateway business logic shall create the scan result using the following EDD definitions, which shall be included in the communication device COMPONENT definition. The EDD LIST also held in the COMPONENT holds the list of devices corresponding to the data structure defined in the XML schema. This LIST shall be referred to by the SCAN_LIST attribute of the communication device component. The corresponding topologyScanResult schema is specified in Annex A.

```
VARIABLE DevAddr
{
  LABEL "Address";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(5);
  HANDLING READ & WRITE;
}
```

```
VARIABLE DevMfg
{
  LABEL "Manufacturer";
  CLASS DEVICE;
  TYPE UNSIGNED_INTEGER(2);
  HANDLING READ & WRITE;
}
```

```
VARIABLE DevType
{
    LABEL "Device Type";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevRev
{
    LABEL "Device Revision";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER;
    HANDLING READ & WRITE;
}

VARIABLE DevTag
{
    LABEL "Tag";
    CLASS DEVICE;
    TYPE ASCII(32);
    HANDLING READ & WRITE;
}

COLLECTION ScanItemType_TP
{
    MEMBERS
    {
        ADDRESS, DevAddr;
    }
}

COLLECTION DeviceIdentificationType
{
    MEMBERS
    {
        MFG,          DevMfg;
        DEV_TYPE,    DevType;
        DEV_REV,     DevRev;
        TAG,         DevTag;
    }
}

COLLECTION ScanResultType
{
    MEMBERS
    {
        ScanItem,          ScanItemType_TP;
        DeviceIdentification, DeviceIdentificationType;
    }
}

LIST TopologyScanResult
{
    TYPE ScanResultType;
    CAPACITY 64;    //Defined to limitation of the comm. device
}
```

5.6.2.8 ScanNext

The method signature specified in IEC 62769-7 applies. The HART Gateway business logic shall create the scan result using the same EDD definitions defined in 5.6.2.7. The corresponding topologyScanResult schema is specified in Annex A.

Annex A (normative)

Topology scan schema

A.1 General

The Topology scan result schema specified in Annex A describes the HART specific format method scan argument `topologyScanResult`. The XML document content and structure shall correspond to the Information Model designed concept to describe a topology in order to enable generic matching between physical devices connected to the network and the FDI Server hosted Information Model.

A.2 IdentificationType

The element content corresponds to the "FunctionalGroup Identification".

The XML schema for IdentificationType is:

```
<xs:complexType name="IdentificationT">
  <xs:attribute name="MANUFACTURER_ID" type="xs:unsignedShort"
use="required">
  </xs:attribute>
  <xs:attribute name="DEVICE_TYPE" type="xs:unsignedShort"
use="required">
  </xs:attribute>
  <xs:attribute name="UNIVERSAL_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="DEVICE_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="SERIAL_NUMBER" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="16777216"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="HARDWARE_REVISION" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedByte">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="32"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SOFTWARE_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="REV_COUNTER" type="xs:unsignedShort"
default="0">
  </xs:attribute>
  <xs:attribute name="TAG" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

The attributes of IdentificationType are described in Table A.1.

Table A.1 – Attributes of IdentificationT

Attribute	Description
MANUFACTURER_ID	Manufacturer identification number.
DEVICE_TYPE	Manufacturer's model number associated with the resource.
UNIVERSAL_REVISION	Universal HART revision associated with the device.
DEVICE_REVISION	Manufacturer's device revision number associated with the device.
SERIAL_NUMBER	Unique serial number of the device.
HARDWARE_REVISION	Manufacturer hardware revision associated with the device.
SOFTWARE_REVISION	Manufacturer firmware revision associated with the device.
REV_COUNTER	The configuration change counter associated with the device.
TAG	The configured Identification tag associated with the device.

A.3 AddressTypeTP

The XML schema for AddressTypeTP is:

```
<xs:complexType name="AddressTypeTP">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:element name="DevPollAddr" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="63"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The elements of AddressTypeTP are described in Table A.2.

Table A.2 – Elements of AddressTypeTP

Element	Description
DevAddr	Holds the 5-byte HART address of the associated device.
DevPollAddr	Holds the HART short address used for polling the associated device.

A.4 AddressTypeIP

The XML schema for AddressTypeIP is:

```
<xs:simpleType name="IPv4T">
  <xs:restriction base="xs:string">
    <xs:pattern value="((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="IPv6T">
  <xs:restriction base="xs:string">
    <xs:pattern value="([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4}"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="AddressTypeIP">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:choice>
          <xs:element name="IPv4Address" type="hart:IPv4T">
          </xs:element>
          <xs:element name="IPv6Address" type="hart:IPv6T">
          </xs:element>
        </xs:choice>
        <xs:element name="IPPort" type="xs:unsignedShort">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The elements of AddressTypeIP are described in Table A.3.

Table A.3 – Elements of AddressTypeIP

Element	Description
DevAddr	Holds the 5-byte HART address of the associated device.
IPAddress	Holds the network IP address of the associated device.
IPPort	Holds the IP Port number used for the associated device.

A.5 AddressTypeTDMA

The XML schema for AddressTypeTDMA is:

```
<xs:complexType name="AddressTypeTDMA">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:element name="NetworkID" type="xs:unsignedShort">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The elements of AddressTypeTDMA are described in Table A.4.

Table A.4 – Elements of AddressTypeTDMA

Element	Description
DevAddr	Holds the 5-byte HART address of the associated device.
IPAddress	Holds the network IP address of the associated device.
IPVersion	Holds the IP version number used for the associated device (either version 4 or 6).
IPPort	Holds the IP Port number used for the associated device.

A.6 AddressType

The XML schema for AddressType is:

```
<xs:complexType name="AddressT">
  <xs:choice>
    <xs:element name="AddressTP" type="hart:AddressTypeTP">
    </xs:element>
    <xs:element name="AddressIP" type="hart:AddressTypeIP">
    </xs:element>
    <xs:element name="AddressTDMA" type="hart:AddressTypeTDMA">
    </xs:element>
  </xs:choice>
</xs:complexType>

<xs:complexType name="LongAddressT">
  <xs:sequence>
    <xs:element name="DevAddr">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="([A-Fa-f0-9]{10})"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The elements of AddressType are described in Table A.5.

Table A.5 – Elements of AddressT

Element	Description
AddressTP	Token passing addresses hold the address information used for ConnectionPointType_HART_TP5, 6, or 7 described in 5.5.1.2.
AddressIP	IP addresses hold the address information described in ConnectionPointType_HART_IP described in 5.5.1.2.
AddressTDMA	TDMA addresses hold the address information described in ConnectionPointType_HART_TDMA described in 5.5.1.2.

A.7 ConnectionPointType

The XML schema for ConnectionPointType is:

```
<xs:complexType name="ConnectionPointT">
  <xs:sequence>
    <xs:element name="Identification" type="hart:IdentificationT">
    </xs:element>
    <xs:element name="Address" type="hart:AddressT">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The elements of a ConnectionPointType are described in Table A.6.

Table A.6 – Elements of ConnectionPointT

Element	Description
Identification	The element data holds the device type identification data. Compared to the Information Model (IEC 62769-5) the ConnectionPoint does not contain or refer to the device type identification data. But in order to support the FDI host system in finding the package that matches the connected device this schema associates the device type identification with the ConnectionPoint.
Address	The address element holds the specific information used to uniquely address the device in the network. The address will be one of three possible types described by AddressType.

A.8 NetworkType

The element type describes the complete scan result for a single network because of the scan method that is provided per instance of a "Communication Device" which exists in a 1:1 relation to a network instance.

The XML schema for a NetworkType is:

```
<xs:complexType name="NetworkT">
  <xs:sequence>
    <xs:element name="ConnectionPoint" type="hart:ConnectionPointT"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

The elements of a NetworkType are described in Table A.7.

Table A.7 – Elements of NetworkT

Element	Description
ConnectionPoint	The ConnectionPoint element holds the address and identification of the network connected device that has been found during bus scan operations.

A.9 Network

The subsequent element hierarchy is used to return the scan result according to the Information Model (IEC 62769-5) representation of a topology.

The XML schema for a Network element is:

```
<xs:element name="Network" type="hart:NetworkT">  
</xs:element>
```

Annex B (normative)

Transfer service parameters

B.1 General

Direct Access Services specified in IEC 62769-2 enable the User Interface Plug-in (UIP) to directly exchange data with the device. Direct data exchange means that data exchanged between a device and a UIP may not be reflected in the Information Model. The IEC 62769-6 defined interface IDirectAccess corresponds to the IEC 62769-2 specified Direct Access Services. Interface IDirectAccess defined functions BeginTransfer and EndTransfer need to convey protocol specific information. The following XML schema documents the protocol specifics.

B.2 receiveData

The element described in the following contains data that is returned through the IDirectAccess function Endtransfer and GetPublishedData defined return value.

```
<xs:element name="receiveData" type="hart:TransferResultDataT">
</xs:element>
```

B.3 sendData

The element described in the following contains data to be submitted through IDirectAccess function Begintransfer defined argument sendData.

```
<xs:element name="sendData" type="hart:TransferSendDataT">
</xs:element>
```

B.4 TransferResultDataT

The XML element type defines the service parameter data format that shall be applied to the EndTransfer and GetPublishedData defined return value. The attribute names defined in TransferResultDataType correspond to the same name of Transfer method arguments specified in 5.6.1.4 and 5.6.2.4.

```
<xs:complexType name="TransferResultDataT">
  <xs:attribute name="COMMAND" type="xs:unsignedInt"
    use="required">
  </xs:attribute>
  <xs:attribute name="REPLY" type="xs:hexBinary" use="required">
  </xs:attribute>
</xs:complexType>
```

Table B.1 – Attributes of TransferResultDataT

Attribute	Description
COMMAND	The attribute corresponds to the CP9/1 command number.
REPLY	The attribute corresponds to the REPLY data bytes in the CP9/1 command data field.

B.5 TransferSendDataT

The XML element type defines the service parameter data format that shall be applied to BeginTransfer defined argument sendData. The attribute names defined in TransferSendDataType correspond to the same name of Transfer method arguments specified in 5.6.1.4 and 5.6.2.4.

```
<xs:complexType name="TransferSendDataT">
  <xs:attribute name="COMMAND" type="xs:unsignedInt"
    use="required">
  </xs:attribute>
  <xs:attribute name="REQUEST" type="xs:hexBinary" use="required">
  </xs:attribute>
</xs:complexType>
```

Table B.2 – Attributes of TransferSendDataT

Attribute	Description
COMMAND	The attribute corresponds to the CP9/1 command number.
REQUEST	The attribute corresponds to the REQUEST data bytes in the CP9/1 command data field.

Bibliography

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

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

IEC 61804 (all parts), *Function blocks (FB) for process control and Electronic Device Description Language (EDDL)*

IEC 62769-1, *Field Device Integration (FDI) – Part 1: Overview*

NOTE 1 IEC 62769-1 is technically identical to FDI-2021.

IEC 62769-2, *Field Device Integration (FDI) – Part 2: FDI Client*

NOTE 2 IEC 62769-2 is technically identical to FDI-2022.

IEC 62769-3, *Field Device Integration (FDI) – Part 3: FDI Server*

NOTE 3 IEC 62769-3 is technically identical to FDI-2023.

IEC 62769-6, *Field Device Integration (FDI) – Part 6: FDI Technology Mapping*

NOTE 4 IEC 62769-6 is technically identical to FDI-2026.

HCF_SPEC-13, *HART® Field Communication Protocol Specification*, available at <www.hartcomm.org>

HCF_SPEC-99, *Command Summary Specification*, available at <www.hartcomm.org>

HCF_SPEC-127, *Universal Command Specification*, available at <www.hartcomm.org>

HCF_SPEC-151, *Common Practice Command Specification*, available at <www.hartcomm.org>

IETF RFC 768, *User Datagram Protocol*, available at <<http://www.ietf.org>>

IETF RFC 793, *Transmission Control Protocol*, available at <<http://www.ietf.org>>

SOMMAIRE

AVANT-PROPOS	44
INTRODUCTION	46
1 Domaine d'application	47
2 Références normatives	47
3 Termes, définitions, abréviations et acronymes	47
3.1 Termes et définitions	47
3.2 Abréviations et acronymes	47
4 Conventions	48
4.1 Syntaxe EDDL	48
4.2 Syntaxe XML	48
4.3 Majuscules	48
5 Profil pour CP 9/1 (HART®) ou CP 9/2 (WirelessHART®)	49
5.1 Généralités	49
5.2 Profil de catalogue	49
5.2.1 Fichier de prise en charge de protocole	49
5.2.2 Définition de CommunicationProfile	49
5.2.3 Appareil de profil	49
5.2.4 Informations relatives à la version de protocole	50
5.3 Association d'un paquetage avec un appareil CP 9/1	50
5.3.1 Mapping d'identification du type d'appareil	50
5.3.2 Mapping de révisions de type d'appareil	51
5.4 Mapping du Modèle d'Information	51
5.4.1 Définition de ProtocolType	51
5.4.2 Mapping de DeviceType	51
5.4.3 Définition du FunctionalGroup "Identification"	52
5.5 Eléments de topologie	52
5.5.1 Définition de ConnectionPoint	52
5.5.2 Définition de l'Appareil de communication	58
5.5.3 Définition de fournisseur de service de communication	59
5.5.4 Définition de réseau	59
5.6 Méthodes	60
5.6.1 Méthodes pour les Serveurs de communication FDI	60
5.6.2 Méthodes pour les passerelles	64
Annexe A (normative) Schéma de balayage topologique	72
A.1 Généralités	72
A.2 IdentificationType	72
A.3 AddressTypeTP	74
A.4 AddressTypeIP	75
A.5 AddressTypeTDMA	75
A.6 AddressType	76
A.7 ConnectionPointType	77
A.8 NetworkType	77
A.9 Network	78
Annexe B (normative) Paramètres du service Transfer	79
B.1 Généralités	79

B.2	receiveData	79
B.3	sendData	79
B.4	TransferResultDataT	79
B.5	TransferSendDataT	80
Bibliographie		81
Tableau 1 – Définition de CommunicationProfile		49
Tableau 2 – Valeurs de catalogue pour appareils de profils		50
Tableau 3 – Informations relatives à la version de protocole		50
Tableau 4 – Mapping des types d'appareils au catalogue		50
Tableau 5 – Définition du ProtocolType HART		51
Tableau 6 – Mapping des propriétés héritées de DeviceType		51
Tableau 7 – Paramètres pour "Identification"		52
Tableau 8 – Définition du ConnectionPointType HART_TP		53
Tableau 9 – Définition du ConnectionPointType HART_IP		55
Tableau 10 – Définition du ConnectionPointType HART_TDMA		57
Tableau 11 – Arguments de la méthode Connect		61
Tableau 12 – Arguments de la méthode Disconnect		61
Tableau 13 – Arguments de la méthode Transfer		62
Tableau 14 – Arguments de la méthode GetPublishedData		63
Tableau 15 – Arguments de la méthode SetAddress		63
Tableau 16 – Arguments de la méthode Connect		65
Tableau 17 – Arguments de la méthode Disconnect		65
Tableau 18 – Arguments de la méthode Transfer		66
Tableau 19 – Arguments de la méthode GetPublishedData		68
Tableau 20 – Arguments de la méthode SetAddress		69
Tableau A.1 – Attributs d'IdentificationT		74
Tableau A.2 – Eléments d'AddressTypeTP		74
Tableau A.3 – Eléments d'AddressTypeIP		75
Tableau A.4 – Eléments d'AddressTypeTDMA		76
Tableau A.5 – Eléments d'AddressT		76
Tableau A.6 – Eléments de ConnectionPointT		77
Tableau A.7 – Eléments de NetworkT		77
Tableau B.1 – Attributs de TransferResultDataT		79
Tableau B.2 – Attributs de TransferSendDataT		80

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 109-1: Profils – HART® et WirelessHART®

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.

La Norme internationale IEC 62769-109-1 a été établie par le sous-comité 65E: Les appareils et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Le texte de cette norme est issu des documents suivants:

CDV	Rapport de vote
65E/356/CDV	65E/419/RVC

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

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

Une liste de toutes les parties de la série IEC 62769, publiées sous le titre général *Intégration des appareils de terrain (FDI)*, peut être consultée sur le site web de l'IEC.

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

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

INTRODUCTION

La Commission Electrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation de brevets intéressant:

- a) la Méthode de Fourniture et d'Installation des Fonctionnalités Spécifiques aux Appareils (cf. famille de brevets DE10357276);
- b) la Méthode et l'appareil utilisés pour l'accès à un module fonctionnel du système d'automation (cf. famille de Brevets EP2182418);
- c) les méthodes et les appareils utilisés pour diminuer les exigences mémoire relatives aux applications logicielles du système de commande de processus (cf. famille de Brevets US2013232186);
- d) modèle d'objet d'appareil extensible (cf. famille de brevets US12/893,680).

L'IEC ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété.

Le détenteur de ces droits de propriété a donné l'assurance à l'IEC qu'il consent à négocier des licences avec des demandeurs du monde entier, soit sans frais soit à des termes et conditions raisonnables et non discriminatoires. A ce propos, la déclaration du détenteur des droits de propriété est enregistrée à l'IEC. Des informations peuvent être demandées à:

- a) ABB Research Ltd
Claes Ryttoft
Affolterstrasse 4
Zurich, 8050
Suisse
- b) Phoenix Contact GmbH & Co KG
Intellectual Property, Licenses & Standards
Flachsmarktstrasse 8, 32825 Blomberg
Allemagne
- c) Fisher Controls International LLC
John Dilger, Emerson Process Management LLLP
301 S. 1st Avenue, Marshalltown, Iowa 50158
Etats-Unis d'Amérique
- d) Rockwell Automation Technologies, Inc.
1 Allen-Bradley Drive
Mayfield Heights, Ohio 44124
Etats-Unis d'Amérique

L'attention est d'autre part attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux qui ont été mentionnés ci-dessus. L'IEC ne saurait être tenue pour responsable de l'identification de ces droits de propriété en tout ou partie.

L'ISO (www.iso.org/patents) et l'IEC (<http://patents.iec.ch>) tiennent à jour des bases de données en ligne sur les brevets relatifs à leurs normes. Les utilisateurs sont encouragés à consulter ces bases de données pour obtenir l'information la plus récente concernant les brevets.

INTÉGRATION DES APPAREILS DE TERRAIN (FDI) –

Partie 109-1: Profils – HART® et WirelessHART®

1 Domaine d'application

La présente partie de l'IEC 62769 spécifie un profil FDI de l'IEC 62769 pour l'IEC 61784-1_CP 9/1 (HART®)¹ et l'IEC 61784-1_CP 9/2 (WirelessHART®)¹.

2 Références normatives

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

IEC 62541-100, *Architecture unifiée OPC – Partie 100: Interface d'appareils*

IEC 62769-4:2015, *Intégration des appareils de terrain (FDI) – Partie 4: Paquetages FDI*

NOTE L'IEC 62769-4 est techniquement identique à la FDI-2024.

IEC 62769-5, *Intégration des appareils de terrain (FDI) – Partie 5: Modèle d'information FDI*

NOTE L'IEC 62769-5 est techniquement identique à la FDI-2025.

IEC 62769-7, *Intégration des appareils de terrain (FDI) – Partie 7: Appareils de communication FDI*

NOTE L'IEC 62769-7 est techniquement identique à la FDI-2027.

3 Termes, définitions, abréviations et acronymes

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 62541-100, l'IEC 62769-4, l'IEC 62769-5 et l'IEC 62769-7 s'appliquent.

3.2 Abréviations et acronymes

Pour les besoins du présent document, les abréviations suivantes s'appliquent:

CP	Communication profile (profil de communication, voir l'IEC 61784-1 ou l'IEC 61784-2)
CPF	Communication profile family (famille de profils de communication, voir l'IEC 61784-1 ou l'IEC 61784-2)

¹ HART et WirelessHART sont les appellations commerciales du consortium HART Communication Foundation, une organisation à but non lucratif d'Austin, Texas, USA. Cette information est donnée à l'intention des utilisateurs du présent document et ne signifie nullement que l'IEC approuve ou recommande l'emploi exclusif du produit ainsi désigné. La conformité n'exige pas l'utilisation de l'appellation commerciale. L'utilisation de l'appellation commerciale exige l'autorisation du détenteur de l'appellation commerciale.

EDD	Electronic Device Description (description d'appareil électronique, voir l'IEC 61804)
EDDL	Electronic Device Description Language (langage de description d'appareil électronique, voir l'IEC 61804)
FDI	Field Device Integration (intégration des appareils de terrain)
FSK	Frequency-Shift-Keying (modulation par déplacement de fréquence)
HCF	HART Communication Foundation
ID	Identification
IM	Information Model (modèle d'Information)
IP	Internet protocol (protocole Internet)
PDU	Protocol data unit (unité de données de protocole)
PSK	Phase-Shift-Keying (modulation par déplacement de phase)
TCP	Transmission Control Protocol (protocole de commande de transmission, voir l'IETF RFC 793)
UDP	User Datagram Protocol (protocole datagramme d'utilisateur, voir l'IETF RFC 768)
XML	Extended Markup Language (langage de balisage extensible)

4 Conventions

4.1 Syntaxe EDDL

Le présent document spécifie le contenu du composant EDD qui fait partie des Paquetages de Communication FDI. La syntaxe EDDL utilise la police *Courier New*. La Syntaxe EDDL est utilisée pour les déclarations des signatures de méthodes, des variables, des structures de données et des composants.

4.2 Syntaxe XML

Les exemples de syntaxe XML utilisent la police *Courier New*. La syntaxe XML est utilisée pour décrire le schéma des documents XML.

Exemple: `<xs:simpleType name="ExampleT">`

4.3 Majuscules

La série IEC 62769 utilise des termes en majuscules pour souligner que ces termes ont une signification spécifique à la FDI.

Certains de ces termes utilisent un acronyme comme un préfixe, par exemple:

- FDI Client (Client FDI) ou
- FDI Server (Serveur FDI).

Certains de ces termes sont des termes composés comme:

- FDI Communication Servers (Serveurs de communication FDI) ou
- Profile Package (Paquetage de profil).

Les noms de paramètres ou attributs sont concaténés en un seul terme, où les termes originaux commencent dans ce terme par une lettre majuscule comme:

- ProtocolSupportFile, ou

- ProtocolType.

Les noms de paramètres ou attributs peuvent aussi être construits en utilisant le caractère de soulignement pour concaténer deux ou plus de plus termes comme:

- PROFILE_ID, ou
- HART_Network

5 Profil pour CP 9/1 (HART®) ou CP 9/2 (WirelessHART®)

5.1 Généralités

Ce document de profil de la spécification FDI dans la norme IEC 6276 décrit les spécificités du protocole dont ont besoin les Paquetages FDI décrivant des Serveurs de Communication FDI, des passerelles et des appareils FDI.

5.2 Profil de catalogue

5.2.1 Fichier de prise en charge de protocole

Aucun fichier complémentaire n'est exigé pour les Paquetages d'Appareils FDI CP 9/1 ou CP 9/2.

5.2.2 Définition de CommunicationProfile

L'IEC 62769-4 définit un type d'énumération CommunicationProfileT pour le schéma XML Catalog. Le Tableau 1 définit les valeurs spécifiques au CP 9/1 pour cette énumération.

Tableau 1 – Définition de CommunicationProfile

CommunicationProfile	Description
hart_fsk	Type d'appareil CP 9/1 qui prend en charge une couche physique FSK (Frequency-Shift-Keying, modulation par déplacement de fréquence sur une paire de fils).
hart_psk	Type d'appareil CP 9/1 qui prend en charge une couche physique PSK (Phase-Shift-Keying, modulation par déplacement de phase) sur une paire de fils. Les appareils prenant en charge la modulation PSK doivent également prendre intrinsèquement en charge la modulation FSK. Par conséquent, la PSK ne sera toujours utilisée qu'en combinaison avec au moins la FSK.
hart_wirelesshart	Type d'appareil CP 9/2 qui prend en charge une couche physique sans fil (communication entre appareil et passerelle).
hart_ip	Type d'appareil CP 9/1 qui prend en charge le Protocole Internet (ces appareils prennent en charge les protocoles TCP et UDP).
hart_rs485	Type d'appareil CP 9/1 qui prend en charge la communication numérique EIA-485.
hart_ir	Type d'appareil CP 9/1 qui prend en charge une Couche physique infrarouge (conçue pour être transparente aux maîtres FSK – incluse uniquement à titre d'information pour indiquer que l'appareil prend en charge la connexion IR).
NOTE Un même appareil CP 9/1 peut prendre en charge plus d'un CP.	

5.2.3 Appareil de profil

Un Paquetage de Profils doit fournir les valeurs de catalogue pour les appareils des profils, permettant au Serveur FDI d'exercer un effet de levier sur une description d'appareil générique, si une description spécifique n'est pas disponible. Les définitions dans le Tableau 2 sont axées sur le contenu de catalogue qui est indépendant du vendeur.

Tableau 2 – Valeurs de catalogue pour appareils de profils

Elément	Attribut	Contenu
PackageType	—	Profile
DeviceModel	—	Vide
Manufacturer	—	Vide

5.2.4 Informations relatives à la version de protocole

L'IEC 62769-4 définit un type d'élément nommé InterfaceT pour le schéma XML Catalog. Le type d'élément InterfaceT contient un élément nommé Version qui est censé fournir des informations de version relatives au profil de protocole de communication appliqué. La valeur doit suivre le schéma d'informations de version définie par l'IEC 62769-4, lequel schéma est défini dans le type d'élément VersionT. Le paragraphe 5.2.4 décrit comment appliquer les versions de protocole actuellement connues pour les entrées CP 9/1 ou CP 9/2 du catalogue d'appareils. La règle générale est d'utiliser la Révision Universelle du protocole pour la partie version majeure de VersionT et la valeur "0" pour les parties version mineure et build ("mouture"). Le Tableau 3 donne les informations relatives à la version de protocole.

Tableau 3 – Informations relatives à la version de protocole

Version de protocole	Valeur de version InterfaceT
Révision universelle HART 5	5.0.0
Révision universelle HART 6	6.0.0
Révision universelle HART 7	7.0.0
La version de protocole définie dans un paquetage est fournie à des fins d'information seulement et ne doit pas être utilisée pour déterminer la compatibilité ou l'applicabilité d'un paquetage vis-à-vis d'un appareil.	

5.3 Association d'un paquetage avec un appareil CP 9/1

5.3.1 Mapping d'identification du type d'appareil

Les types d'appareils CP 9/1 sont identifiés de façon univoque par les paramètres Manufacturer, Model et DeviceRevision. Ces paramètres sont utilisés pour associer une instance donnée d'appareil avec un Paquetage d'Appareil FDI. Ces paramètres sont mappés au Catalogue de Paquetage d'Appareil FDI conformément au Tableau 4.

Tableau 4 – Mapping des types d'appareils au catalogue

Elément de Catalog	Mapping CP (voir 5.4.2)
Elément Manufacturer d'InterfaceT (IEC 62769-4:2015, Article E.11)	Manufacturer Format de chaîne "0xdddd" où dddd est le numéro Manufacturer au format hexadécimal.
Elément DeviceModel d'InterfaceT (IEC 62769-4:2015, Article E.11)	Model Format de chaîne "0xdddd" où dddd est le numéro Model au format hexadécimal.
Elément DeviceRevision ListOfSupportedDeviceRevisionsT (IEC 62769-4:2015, Article E.21)	DeviceRevision Chaîne au format "x.0.0" où x est la DeviceRevision au format décimal (pas de zéro de tête).

5.3.2 Mapping de révisions de type d'appareil

Chaque type d'appareil est identifié selon 5.3.1. Si un paquetage avec une DeviceRevision correspondante n'est pas disponible, tout paquetage FDI CP 9/1 pour un manufacturer (fabricant) et model (modèle) qui correspondent doit toujours être compatible avec un appareil de terrain tant que la révision d'appareil relative à l'appareil de terrain est supérieure ou égale à la révision d'appareil spécifiée dans le paquetage FDI.

5.4 Mapping du Modèle d'Information

5.4.1 Définition de ProtocolType

Le Tableau 5 définit le ProtocolType utilisé pour identifier des communications réseau CP 9/1.

Tableau 5 – Définition du ProtocolType HART

Attribut	Valeur				
BrowseName	HART				
IsAbstract	False				
Références	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
Hérite des propriétés de ProtocolType défini dans l'IEC 62541-100.					

5.4.2 Mapping de DeviceType

Chaque type d'appareil hérite des propriétés du DeviceType. Le mapping des propriétés héritées du DeviceType est défini dans le Tableau 6.

Tableau 6 – Mapping des propriétés héritées de DeviceType

Propriété	Mapping à la Foundation
SerialNumber	ID unique de trois octets d'un appareil, retourné dans les octets 9 à 11 de la Commande 0 ou de la Commande 11 ou de la Commande 21.
RevisionCounter	Compteur de changement de configuration de deux octets, retourné dans les octets 14 et 15 de la Commande 0 ou de la Commande 11 ou de la Commande 21. -1 (non défini) pour les appareils de la révision HART 5.
Manufacturer	Pour les appareils de la révision HART 7 ou supérieure: Il s'agit du code de fabricant de deux octets d'un appareil, retourné dans les octets 17 et 18 de la Commande 0 ou de la Commande 11 ou de la Commande 21. Pour les appareils de la révision HART 6 ou inférieure: L'octet de poids fort doit être fixé à 0 et l'octet de poids faible est retourné dans l'octet 1 de la Commande 0 ou de la Commande 11 ou de la Commande 21.
Model	Type étendu d'appareils de deux octets d'un appareil, retourné dans les octets 1 et 2 de la Commande 0 ou de la Commande 11 ou de la Commande 21.
DeviceManual	Entrée au format de chaîne textuelle (non prise en charge) ^a .
DeviceRevision	Niveau de révision d'appareils d'un octet d'un appareil, retourné dans l'octet 5 de la Commande 0 ou de la Commande 11 ou de la Commande 21.
SoftwareRevision	Niveau de révision de logiciel d'un octet d'un appareil, retourné dans l'octet 6 de la Commande 0 ou de la Commande 11 ou de la Commande 21.
HardwareRevision	Niveau de révision de matériel d'un octet d'un appareil, retourné dans l'octet 7 (seulement les cinq bits de poids fort) de la Commande 0 ou de la Commande 11 ou de la Commande 21.
^a Les manuels d'appareils sont exposés comme pièces jointes du Paquetage d'Appareil FDI.	

5.4.3 Définition du FunctionalGroup "Identification"

Comme défini dans l'IEC 62541-100, chaque représentation d'appareil dans le Modèle d'Information hébergé par le Serveur FDI doit contenir un FunctionalGroup (c'est-à-dire: groupe fonctionnel) spécifique à un protocole appelé "Identification". Ce FunctionalGroup organise les variables détectées dans l'instance de type d'appareil. Le FunctionalGroup "Identification" pour CP 9/1 est défini dans le Tableau 7.

Tableau 7 – Paramètres pour "Identification"

BrowseName	DataType	Facultatif/Obligatoire
MANUFACTURER_ID	UInt16	Obligatoire
DEVICE_TYPE	UInt16	Obligatoire
DEVICE_REVISION	UInt8	Obligatoire
UNIVERSAL_REVISION	UInt8	Facultatif
SERIAL_NUMBER	UInt24	Facultatif
HARDWARE_REVISION	UInt8	Facultatif
SOFTWARE_REVISION	UInt8	Facultatif
REVISION_COUNTER	UInt16	Facultatif

5.5 Éléments de topologie

5.5.1 Définition de ConnectionPoint

5.5.1.1 Généralités

Les appareils CP 9/1 peuvent prendre en charge jusqu'à cinq types différents de ConnectionPoint qui sont utilisés pour les communications réseau.

5.5.1.2 HART_TP5, HART_TP6, HART_TP7

Les types ConnectionPoint HART_TP5, HART_TP6 et HART_TP7 doivent être utilisés pour identifier la communication de réseau avec passage de jeton CP 9/1 et sont définis dans le Tableau 8. Les types HART_TP5, HART_TP6 et HART_TP7 contiennent tous les mêmes propriétés, mais chacun d'eux fournit des informations de quantification différentes pour certaines des propriétés (décrites ci-dessous). La Version de Protocole (UNIVERSAL_REVISION) décrite en 5.2.3 peut être utilisée comme aide pour déterminer lequel de trois types de Points de Connexion avec passage de jeton est le plus approprié. Les communications avec passage de jeton CP 9/1 peuvent être utilisées sur une diversité de couches physiques. Les connexions de couche physique FSK, PSK, RS485 et infrarouges doivent toutes utiliser le type de connexion HART_TP. Les types ConnectionPoint HART_TP5, HART_TP6 et HART_TP7 sont des sous-types du type abstrait ConnectionPointType défini dans l'IEC 62769-5.

La propriété DevAddr doit être l'adresse longue (cinq octets) pour l'appareil. En outre, elle est le seul paramètre nécessaire pour communiquer avec l'appareil de terrain.

La propriété DevMfg doit être le Manufacturer ID (Identificateur de fabricant) de deux octets et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété DevType doit être le type étendu d'appareils de deux octets et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété DevRev doit être la révision d'appareil et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété DevTag doit être le marqueur long pour les appareils de la version de protocole HART[®] 6 ou 7. La propriété DevTag doit être le marqueur pour les appareils de la version de protocole 5. La propriété DevTag peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne. Les Points de Connexion HART_TP5 doivent limiter à 8 caractères la longueur de DevTag. Les Points de Connexion HART_TP6 et HART_TP7 doivent limiter à 32 caractères la longueur de DevTag.

La propriété DevPollAddr doit être l'adresse d'interrogation. Par ailleurs, elle peut être utilisée pour identifier quel appareil est localisé à une adresse d'interrogation spécifique. Les Points de Connexion HART_TP5 doivent être limités à des valeurs comprises entre 0 et 15 pour la propriété DevPollAddr. Les Points de Connexion HART_TP6 doivent être limités à des valeurs comprises entre 0 et 31 pour la propriété DevPollAddr. Les Points de Connexion HART_TP7 doivent être limités à des valeurs comprises entre 0 et 63 pour la propriété DevPollAddr.

Pour la post-compatibilité, un Point de Connexion HART_TP de révision inférieure est compatible et peut être utilisé pour une connexion d'appareil d'une révision universelle supérieure. Par exemple, si un appareil d'une future révision universelle HART 8 est rencontré et aucun point de connexion HART_TP8 n'est disponible dans le Serveur FDI, le point de connexion HART_TP7 sera compatible et doit être utilisé pour se connecter à l'appareil. Si la Version de Protocole (c'est-à-dire la Révision universelle) est inconnue pour une raison quelconque, le Point de Connexion HART_TP5 peut être utilisé et sera post-compatible avec les révisions universelles ultérieures.

Tableau 8 – Définition du ConnectionPointType HART_TP

Attribut	Valeur				
BrowseName	ConnexionPoint_HART_TP5 ou ConnexionPoint_HART_TP6 ou ConnexionPoint_HART_TP7				
IsAbstract	False				
Références	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Hérite des propriétés de ConnectionPointType défini dans l'IEC 62769-5.					
HasProperty	Variable	DevAddr	UInt40	PropertyType	Obligatoire
HasProperty	Variable	DevMfg	UInt16	PropertyType	Facultatif
HasProperty	Variable	DevType	UInt16	PropertyType	Facultatif
HasProperty	Variable	DevRev	UInt16	PropertyType	Facultatif
HasProperty	Variable	DevTag	String	PropertyType	Facultatif
HasProperty	Variable	DevPollAddr	UInt8	PropertyType	Facultatif

Les types ConnectionPoint HART_TP5, HART_TP6 et HART_TP7 doivent être décrits par un élément EDD contenu dans un Paquetage FDI relatif aux appareils de communication qui peut piloter un réseau CP 9/1. Les réelles propriétés de ConnectionPoint sont déclarées par des constructions VARIABLE regroupées dans une COLLECTION appelée ConnectionPoint_HART_TP5, ConnectionPoint_HART_TP6 ou ConnectionPoint_HART_TP7. Le code source EDDL ci-après est un exemple décrivant un Point de Connexion TP5.

```

COMPONENT ConnectionPoint_HART_TP5
{
    LABEL "HART TP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_TP5;
}

VARIABLE DevAddr
{
    LABEL "Address";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(5);
    HANDLING READ & WRITE;

```

```

}

VARIABLE DevMfg
{
    LABEL "Manufacturer";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevType
{
    LABEL "Device Type";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevRev
{
    LABEL "Device Revision";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER;
    HANDLING READ & WRITE;
}

VARIABLE DevTag
{
    LABEL "Tag";
    CLASS DEVICE;
    TYPE ASCII(32);
    HANDLING READ & WRITE;
}

VARIABLE DevPollAddr
{
    LABEL "Poll Address";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER
    {
        MAX_VALUE 15; //Define appropriate max value for various revisions
    }
    HANDLING READ & WRITE;
}

COLLECTION ConnectionPoint_TP5
{
    LABEL "Connection Point";
    MEMBERS
    {
        ADDRESS,    DevAddr,    "Device Address";
        MFG,        DevMfg,    "Manufacturer";
        DEV_TYPE,   DevType,    "Device Type";
        DEV_REV,    DevRev,     "Device Revision";
        TAG,        DevTag,     "Device Tag";
        POLL_ADDR, DevPollAddr, "Poll Address";
    }
}

```


5.5.1.3 HART_IP

Le type `ConnectionPoint` `HART_IP` doit être utilisé pour identifier la communication de réseau IP CP 9/1 et est défini dans le Tableau 9. Les communications `HART_IP` peuvent être utilisées sur une diversité de couches physiques. Toutes les connexions Ethernet doivent utiliser le type de connexion `HART_IP`. Les couches physiques supplémentaires qui seront développées dans le futur peuvent aussi utiliser le type de connexion `HART_IP`. Le type `ConnectionPoint` `HART_IP` est un sous-type du type abstrait `ConnectionPointType` défini dans l'IEC 62769-5.

La propriété `IPAddress` doit indiquer l'adresse IP (jusqu'à 16 octets) utilisée pour la connexion.

La propriété `IPVersion` doit indiquer la version IP utilisée pour la connexion (version 4 ou 6).

La propriété `IPPort` doit être le numéro de port IP pour la connexion. Le numéro de port par défaut utilisé pour `HART_IP` est 5 094.

La propriété `DevAddr` doit être l'adresse longue (cinq octets) pour l'appareil.

La propriété `DevMfg` doit être le Manufacturer ID (Identificateur de fabricant) de deux octets et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété `DevType` doit être le type étendu d'appareils de deux octets et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété `DevRev` doit être la révision d'appareil et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété `DevType` doit être le marqueur long (avec 32 caractères au maximum) et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

Tableau 9 – Définition du `ConnectionPointType` `HART_IP`

Attribut	Valeur				
<code>BrowseName</code>	<code>ConnectionPoint_HART_IP</code>				
<code>IsAbstract</code>	False				
Références	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Hérite des propriétés de <code>ConnectionPointType</code> défini dans l'IEC 62769-5.					
<code>HasProperty</code>	Variable	<code>IPAddress</code>	<code>ByteString</code>	<code>PropertyType</code>	Obligatoire
<code>HasProperty</code>	Variable	<code>IPVersion</code>	<code>UInt8</code>	<code>PropertyType</code>	Obligatoire
<code>HasProperty</code>	Variable	<code>IPPort</code>	<code>UInt16</code>	<code>PropertyType</code>	Obligatoire
<code>HasProperty</code>	Variable	<code>DevAddr</code>	<code>UInt40</code>	<code>PropertyType</code>	Obligatoire
<code>HasProperty</code>	Variable	<code>DevMfg</code>	<code>UInt16</code>	<code>PropertyType</code>	Facultatif
<code>HasProperty</code>	Variable	<code>DevType</code>	<code>UInt16</code>	<code>PropertyType</code>	Facultatif
<code>HasProperty</code>	Variable	<code>DevRev</code>	<code>UInt16</code>	<code>PropertyType</code>	Facultatif
<code>HasProperty</code>	Variable	<code>DevTag</code>	<code>String</code>	<code>PropertyType</code>	Facultatif

Le type `ConnectionPoint` `HART_IP` doit être décrit par un élément EDD contenu dans un Paquetage FDI relatif aux appareils de communication qui peut piloter un réseau CP 9/1. Les propriétés réelles `ConnectionPoint` sont déclarées par des constructions de `VARIABLE` regroupées dans une `COLLECTION` appelée `ConnectionPoint_HART_IP`. Le code source EDDL ci-après est un exemple décrivant un Point de Connexion IP.

```

COMPONENT ConnectionPoint_HART_IP
{
    LABEL "HART IP Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_IP;
}

ARRAY IPAddress
{
    LABEL "IP Address";
    CLASS DEVICE;
    TYPE OCTET(16);
    HANDLING READ & WRITE;
}

VARIABLE IPVersion
{
    LABEL "IP Version";
    CLASS DEVICE;
    TYPE ENUMERATED
    {
        { 4, "IPv4" },
        { 6, "IPv6" }
    }
    HANDLING READ & WRITE;
}

VARIABLE IPPort
{
    LABEL "IP Port";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER (2);
    DEFAULT_VALUE 5 094;
    HANDLING READ & WRITE;
}

COLLECTION ConnectionPoint_IP
{
    LABEL "Connection Point";
    MEMBERS
    {
        IPADDRESS, IPAddress, "IP Address";
        IPVERSION, IPVersion, "IP Version";
        IPPORT, IPPort, "IP Port";
        ADDRESS, DevAddr, "Device Address";
        MFG, DevMfg, "Manufacturer";
        DEV_TYPE, DevType, "Device Type";
        DEV_REV, DevRev, "Device Revision";
        TAG, DevTag, "Device Tag";
    }
}

```

5.5.1.4 HART_TDMA

Le type ConnectionPoint HART_TDMA doit être utilisé pour identifier une communication réseau à accès multiple par répartition dans le temps CP 9/2 et il est défini dans le Tableau 10. Les communications HART_TDMA peuvent être utilisées sur une diversité de couches physiques. Le type ConnectionPoint HART_TDMA est un sous-type du type abstrait ConnectionPointType défini dans l'IEC 62769-5. Toutes les connexions WirelessHART doivent utiliser le type de connexion HART_TDMA. Les couches physiques supplémentaires

qui seront développées dans le futur peuvent aussi utiliser le type de connexion HART_TDMA.

La propriété Network doit être l'ID de réseau pour le réseau.

La propriété DevAddr doit être l'adresse longue (cinq octets) pour l'appareil.

La propriété DevMfg doit être le Manufacturer ID (Identificateur de fabricant) de deux octets et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété DevType doit être le type étendu d'appareils de deux octets et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété DevRev doit être la révision d'appareil et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

La propriété DevType doit être le marqueur long (avec 32 caractères au maximum) et elle peut être utilisée pour aider à automatiser le processus consistant à assigner des appareils actifs de la liste de balayage à des réceptacles hors ligne.

Tableau 10 – Définition du ConnectionPointType HART_TDMA

Attribut	Valeur				
BrowseName	ConnectionPoint_HART_TDMA				
IsAbstract	False				
Références	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Hérite des propriétés de ConnectionPointType défini dans l'IEC 62769-5.					
HasProperty	Variable	Network	UInt16	PropertyType	Obligatoire
HasProperty	Variable	DevAddr	UInt40	PropertyType	Obligatoire
HasProperty	Variable	DevMfg	UInt16	PropertyType	Facultatif
HasProperty	Variable	DevType	UInt16	PropertyType	Facultatif
HasProperty	Variable	DevRev	UInt16	PropertyType	Facultatif
HasProperty	Variable	DevTag	String	PropertyType	Facultatif

Le type ConnectionPoint HART_TDMA doit être décrit par un élément EDD contenu dans un Paquetage FDI relatif aux appareils de communication qui peut piloter un réseau CP 9/2. Les propriétés réelles ConnectionPoint sont déclarées par des constructions de VARIABLE regroupées dans une COLLECTION appelée ConnectionPoint_HART_TDMA. Le code source EDDL ci-après est un exemple décrivant un Point de Connexion TDMA.

```

COMPONENT ConnectionPoint_HART_TDMA
{
    LABEL "HART TDMA Connection Point";
    CLASSIFICATION NETWORK_CONNECTION_POINT;
    CAN_DELETE FALSE;
    PROTOCOL HART;
    CONNECTION_POINT ConnectionPoint_TDMA;
}

VARIABLE Network
{
    LABEL "Network ID";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER (2);
    HANDLING READ & WRITE;

```

```

}

COLLECTION ConnectionPoint_TDMA
{
    LABEL "Connection Point";
    MEMBERS
    {
        NETWORK,    Network,    "Network ID";
        ADDRESS,    DevAddr,    "Device Address";
        MFG,         DevMfg,     "Manufacturer";
        DEV_TYPE,    DevType,    "Device Type";
        DEV_REV,     DevRev,     "Device Revision";
        TAG,         DevTag,     "Device Tag";
    }
}

```

5.5.2 Définition de l'Appareil de communication

Conformément à l'IEC 62769-7, chaque Paquetage de communication FDI doit contenir un élément EDD décrivant l'appareil de communication. Le code source EDDL ci-après est un exemple décrivant un Serveur de Communication FDI.

```

COMPONENT CommunicationServer_HART
{
    LABEL "HART Communication Server";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { Communication_Device_Setup_HART}
}

COMPONENT_RELATION Communication_Device_Setup_HART
{
    LABEL "Relation between Device and Communication Device";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        CommunicationDevice_HART{AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 4;
}

```

La sémantique des constructions EDDL affichées ci-dessus est décrite dans l'IEC 62769-7. EDDL COMPONENT sera utilisé par le serveur FDI et le Serveur de Communication FDI pour créer une instance de type CommunicationServerType comme décrit dans l'IEC 62769-7.

Conformément à l'IEC 62769-7, chaque Paquetage de Communication FDI doit contenir au moins un élément EDD décrivant au moins un composant appareil de communication. Le code source EDDL ci-après est un exemple décrivant un appareil de communication CP 9/1.

```

COMPONENT CommunicationDevice_HART
{
    LABEL "HART Communication Device";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS { ServiceProviderRelation_HART}
    BYTE_ORDER BIG_ENDIAN;
}

```

```

COMPONENT_RELATION ServiceProviderRelation_HART
{
    LABEL "Relation to communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
    COMPONENTS
    {
        CommunicationServiceProvider_HART {AUTO_CREATE 1;}
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

La sémantique des constructions EDDL affichées ci-dessus est décrite dans l'IEC 62769-7. EDDL COMPONENT sera utilisé par le serveur FDI et le Serveur de Communication FDI pour créer une instance de type *CommunicationServerChannelType* comme décrit dans l'IEC 62769-7.

5.5.3 Définition de fournisseur de service de communication

Conformément à l'IEC 62769-7, chaque Paquetage de Communication FDI doit contenir au moins un élément EDD qui décrit au moins un composant fournisseur de service de communication. Le code source EDDL ci-après est un exemple décrivant un composant fournisseur de service communication CP 9/1.

La référence du composant (*ConnectionPoint_HART_IP*) correspond à l'une des définitions de Point de Connexion correspondantes données en 5.5.1. La valeur de l'attribut *BYTE_ORDER* doit être positionnée en fonction du protocole.

```

COMPONENT CommunicationServiceProvider_HART
{
    LABEL "HART Communication Service Provider";
    CAN_DELETE FALSE;
    CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
    COMPONENT_RELATIONS { ServiceProviderConnectionPointRelation_HART }
    BYTE_ORDER BIG_ENDIAN;
}

COMPONENT_RELATION ServiceProviderConnectionPointRelation_HART
{
    LABEL "Relation between communication service and connection point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING
    {
        IPAddress
    }
    COMPONENTS
    {
        ConnectionPoint_HART_IP { AUTO_CREATE 1; }
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 1;
}

```

La sémantique des constructions EDDL affichées ci-dessus est décrite dans l'IEC 62769-7. EDDL COMPONENT sera utilisé par le serveur FDI et le Serveur de Communication FDI pour créer une instance de type *CommunicationServiceType* comme décrit dans l'IEC 62769-7.

5.5.4 Définition de réseau

Conformément à l'IEC 62769-7, chaque Paquetage de Communication FDI doit contenir au moins un élément EDD décrivant des contraintes de configuration de réseau utilisant la

construction "composant". Le code source EDDL ci-après est un exemple décrivant un réseau.

```
COMPONENT Network_HART
{
    LABEL "HART Network";
    CAN_DELETE TRUE;
    CLASSIFICATION NETWORK;
    COMPONENT_RELATIONS { NetworkConnectionPointRelation_HART }
}

COMPONENT_RELATION NetworkConnectionPointRelation_HART
{
    LABEL "Relation between network and connection point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {DevPollAddr}
    COMPONENTS
    {
        ConnectionPoint_HART_TP5
        {
            MAXIMUM_NUMBER 16;
        },
        ConnectionPoint_HART_TP6
        {
            MAXIMUM_NUMBER 32;
        },
        ConnectionPoint_HART_TP7
        {
            MAXIMUM_NUMBER 64;
        }
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER 64;    //Defined to limitations of the comm. device
}
```

La sémantique des constructions EDDL affichées ci-dessus est décrite dans l'IEC 62769-7. EDDL COMPONENT sera utilisé par le serveur FDI et le Serveur de Communication FDI pour créer une instance de type *NetworkType* comme décrit dans l'IEC 62769-5. Certains appareils de communication peuvent organiser plusieurs cartes E/S (entrées/sorties) comme des réseaux individuels, alors que d'autres peuvent présenter un seul réseau qui comprend tous les Points de Connexion à travers toutes les cartes E/S.

5.6 Méthodes

5.6.1 Méthodes pour les Serveurs de communication FDI

5.6.1.1 Généralités

Le Serveur de Communication FDI doit mettre en œuvre des services selon les signatures de méthode décrites en 5.6.1 et conformément au Modèle d'Information.

5.6.1.2 Connect

Le Tableau 11 répertorie les arguments de la Méthode Connect.

Signature:

```
Connect (
    [in]  ByteString CommunicationRelationId,
    [in]  ByteString longAddress,
    [out] Int32      ServiceError);
```

Tableau 11 – Arguments de la méthode Connect

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint de l'appareil qui représente la connexion entre un appareil et un réseau physique directement connecté au matériel du Serveur de Communication FDI. Le nodeId permet de trouver la relation directe parent-enfant.
LongAddress	Le nom d'argument doit correspondre au nom d'attribut correspondant défini pour le ConnectionPoint qui est décrit par un élément EDD correspondant spécifié en 5.5.1. L'argument contient l'adresse longue (cinq octets) de l'appareil. La valeur peut être obtenue par la méthode Scan.
ServiceError	0: OK/exécution terminée, connexion établie avec succès -1: Echec de Connect/annulée par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de Connect/appareil introuvable -4: Echec de Connect/adresse de nœud appareil non valide -5: Echec de Connect/identification d'appareil non valide
NOTE 1 L'IEC 62769-7 définit l'argument AddressData de la Méthode Connect comme une matrice de Variantes. Les arguments LongAddress définis dans le tableau sont représentés comme des entrées de la matrice de Variantes dans l'ordre dans lequel elles sont spécifiées ci-dessus. NOTE 2 L'IEC 62769-7 définit l'argument DeviceInformation comme une liste d'arguments spécifiques au protocole dans laquelle la méthode Connect stocke les données obtenues. L'argument DeviceInformation est défini comme une matrice de Variantes. L'argument DeviceInformation n'est pas utilisé pour les appareils CP 9/1.	

5.6.1.3 Disconnect

Le Tableau 12 répertorie les arguments de la Méthode Disconnect.

Signature:

```

Disconnect (
    [in]  ByteString CommunicationRelationId,
    [out] Int32      ServiceError);

```

Tableau 12 – Arguments de la méthode Disconnect

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint de l'appareil qui représente la connexion entre un appareil et un réseau physique directement connecté au matériel du Serveur de Communication FDI. Le nodeId permet de trouver la relation directe parent-enfant.
ServiceError	0: OK/déconnexion effectuée avec succès -1: Echec de Disconnect/aucune relation de communication existante -2: Echec de Disconnect/identificateur de relation de communication non valide

5.6.1.4 Transfer

Le Tableau 13 répertorie les arguments de la Méthode Transfer.

Signature:

```
Transfer (
    [in]  ByteString CommunicationRelationId,
    [in]  UInt16      Command,
    [in]  ByteString Request,
    [out] ByteString Reply,
    [out] Int32       ServiceError);
```

Tableau 13 – Arguments de la méthode Transfer

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint représentant la connexion entre un appareil et un réseau physique qui est directement connecté au matériel du Serveur de Communication FDI. Le nodeId permet de trouver la relation directe parent-enfant.
Command	Numéro de commande relatif à la commande CP 9/1 qui doit être envoyée. Le Serveur de Communication FDI doit générer et envoyer la PDU dans le format correct conformément aux spécifications de protocole CP 9/1 (ex.: prise en compte de l'expansion de la commande).
Request	Une matrice d'octets représentant les octets de données dans la demande de commande CP 9/1 (ex.: commençant après le nombre total d'octets et finissant avant la somme de contrôle).
Reply	Une matrice d'octets représentant les octets de données dans la réponse de commande CP 9/1 (ex.: commençant après le nombre total d'octets et finissant avant la somme de contrôle).
ServiceError	0: OK/exécution terminée, Reply contient le résultat -1: Echec de Transfer/annulé par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de Transfer/aucune relation de communication existante -4: Echec de Transfer/identificateur de relation de communication non valide -5: Echec de Transfer/contenu de Request non valide -6: Echec de Transfer/format de Reply non valide
NOTE 1 L'IEC 62769-7 définit l'argument SendData de la méthode Transfer comme une matrice de Variantes. Les arguments Command et Request définis dans le tableau sont représentés comme des entrées de la matrice de Variantes dans l'ordre dans lequel elles sont spécifiées ci-dessus.	
NOTE 2 L'IEC 62769-7 définit l'argument ReceiveData de la méthode Transfer comme une matrice de variantes. L'argument Reply défini dans le tableau est représenté comme l'entrée de la matrice de Variantes.	

5.6.1.5 GetPublishedData

Les messages de notification d'événements et de salves CP 9/1 représentent des messages non sollicités tels que définis dans l'IEC 62769-7. Le Tableau 14 répertorie les arguments de la Méthode GetPublishedData.

Signature:

```
GetPublishedData (
    [in]  ByteString CommunicationRelationId,
    [out] UInt16      Command,
    [out] ByteString Reply,
    [out] DateTime     Timestamp
    [out] Int32       ServiceError);
```


Tableau 14 – Arguments de la méthode GetPublishedData

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint représentant la connexion entre un appareil et un réseau physique au sein du Modèle d'Information.
Command	Numéro de commande relatif à la commande CP 9/1 publiée. Le Serveur de Communication FDI doit analyser la PDU dans divers formats conformément aux normes de Protocole CP 9/1 (ex.: prise en compte de l'expansion de la commande).
Reply	Une matrice d'octets représentant les octets de données dans la réponse de commande CP 9/1 publiée (ex.: commençant après le nombre total d'octets et finissant avant la somme de contrôle).
TimeStamp	Heure à laquelle les données publiées ont été saisies.
ServiceError	0: OK/exécution terminée, Reply contient le résultat -1: Echec de GetPublishedData/annulé par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de GetPublishedData/non pris en charge -4: Echec de GetPublishedData/aucune relation de communication existante -5: Echec de GetPublishedData/identificateur de relation de communication non valide -8: Echec de GetPublishedData/aucune donnée de save/d'événement publiée
NOTE L'IEC 62769-7 définit l'argument ReceiveData de la méthode GetPublishedData comme une matrice de variantes. Les arguments Command et Reply définis dans le tableau sont représentés comme des entrées de la matrice de Variantes dans l'ordre dans lequel elles sont spécifiées ci-dessus.	

5.6.1.6 SetAddress

Le Tableau 15 répertorie les arguments de la Méthode SetAddress.

Signature

```

SetAddress (
    [in]   UInt8 OldPollAddress,
    [in]   UInt8 NewPollAddress,
    [out]  Int32 ServiceError);

```

Tableau 15 – Arguments de la méthode SetAddress

Argument	Description
OldPollAddress	La valeur d'argument contient l'adresse actuelle d'un appareil. Les valeurs permises sont 0..63.
NewPollAddress	La valeur d'argument contient la nouvelle adresse d'un appareil. Les valeurs permises sont 0..63.
ServiceError	0: OK/exécution terminée avec succès -1: Echec de SetAddress/annulé par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de SetAddress/non initialisé -4: Echec de SetAddress/non connecté à un réseau -5: Echec de SetAddress/aucun appareil correspondant à oldAddress n'a été trouvé -6: Echec de SetAddress/erreur d'adresse en double -7: Echec de SetAddress/l'appareil n'a pas accepté la nouvelle adresse -8: Echec de SetAddress/oldAddress non valide

Argument	Description
	(en termes de syntaxe, de type de données, de format des données, etc.) -9: Echec de SetAddress/newAddress non valide (en termes de syntaxe, de type de données, de format des données, etc.) -10: Echec de SetAddress/impossible à l'état connecté
NOTE 1 L'IEC 62769-7 définit l'argument OldAddress de la méthode SetAddress comme une matrice de Variantes. L'argument OldPollAddress défini dans le tableau est représenté comme l'entrée de la matrice de Variantes.	
NOTE 2 L'IEC 62769-7 définit l'argument NewAddress de la méthode SetAddress comme une matrice de Variantes. L'argument NewPollAddress défini dans le tableau est représenté comme l'entrée de la matrice de Variantes.	

5.6.1.7 Scan

La signature de méthode spécifiée dans l'IEC 62769-7 s'applique. Le schéma topologyScanResult correspondant est spécifié à l'Annexe A.

5.6.1.8 ResetScan

La signature de méthode spécifiée dans l'IEC 62769-7 s'applique. Le schéma topologyScanResult correspondant est spécifié à l'Annexe A.

5.6.2 Méthodes pour les passerelles

5.6.2.1 Généralités

Les signatures de méthode décrites en 5.6.2 doivent être mises en œuvre dans l'élément EDD IEC 62769-4 contenu dans un Paquetage FDI relatif à une Passerelle contenant les définitions d'appareils de communication.

5.6.2.2 Connect

Le Tableau 16 répertorie les arguments de la Méthode Connect.

Signature:

```

METHOD BeginConnect(
    DD_String      CommunicationRelationId,
    DD_String      longAddress,
    unsigned long   ServiceID,
    unsigned long   &DelayForNextCall,
    long           &ServiceError)
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD EndConnect(
    DD_String      CommunicationRelationId,
    unsigned long   ServiceID,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD CancelConnect(
    DD_String      CommunicationRelationId,
    unsigned long   ServiceID,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

```

Tableau 16 – Arguments de la méthode Connect

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint représentant la connexion entre un appareil et un réseau physique qui est directement connecté au matériel du Serveur de Communication FDI. Le nodeId permet de trouver la relation directe parent-enfant.
LongAddress	Le nom d'argument doit correspondre au nom d'attribut correspondant défini pour le ConnectionPoint qui est décrit par un élément EDD correspondant spécifié en 5.5.1. L'argument contient l'adresse longue (cinq octets) de l'appareil. La valeur peut être obtenue par la méthode Scan.
ServiceId	Le code de transaction de service établit la relation entre la demande de service et la réponse correspondante.
DelayForNextCall	La valeur spécifie un délai, en ms, pour limiter le cycle d'invocation EndConnect qui ne doit pas être plus rapide que la valeur spécifiée dans la valeur d'argument.
ServiceError	1: OK/fonction démarrée de façon asynchrone, le résultat doit être interrogé avec EndConnect 0: OK/exécution terminée, connexion établie avec succès -1: Echec de Connect/annulée par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de Connect/appareil introuvable -4: Echec de Connect/adresse d'appareil non valide -5: Echec de Connect/identification d'appareil non valide

5.6.2.3 Disconnect

Le Tableau 17 répertorie les arguments de la Méthode Disconnect.

Signature:

```

METHOD Disconnect(
    DD_String  communicationRelationId,
    Long       &serviceError)
{
    DEFINITION {<Gateway specific implementation>}
}

```

Tableau 17 – Arguments de la méthode Disconnect

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint représentant la connexion entre un appareil et un réseau physique qui est directement connecté au matériel du Serveur de Communication FDI. Le nodeId permet de trouver la relation directe parent-enfant.
ServiceError	0: OK/déconnexion effectuée avec succès -1: Echec de Disconnect/aucune relation de communication existante -2: Echec de Disconnect/identificateur de relation de communication non valide

5.6.2.4 Transfer

Le Tableau 18 répertorie les arguments de la Méthode Transfer.

Signature:

```

METHOD BeginTransfer(
    DD_String      CommunicationRelationId,
    unsigned short Command,
    DD_String      Request,
    DD_String      &Reply,
    unsigned long  ServiceId,
    unsigned long  &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD EndTransfer(
    DD_String      CommunicationRelationId,
    DD_String      &Reply,
    unsigned long  ServiceId,
    unsigned long  &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD CancelTransfer(
    DD_String      CommunicationRelationId,
    unsigned long  ServiceId,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

```

Tableau 18 – Arguments de la méthode Transfer

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint représentant la connexion entre un appareil et un réseau physique qui est directement connecté au matériel du Serveur de Communication FDI. Le nodeId permet de trouver la relation directe parent-enfant.
Command	Numéro de commande relatif à la commande CP 9/1 qui doit être envoyée. Le Serveur de Communication FDI doit générer et envoyer la PDU dans le format correct conformément aux spécifications de protocole CP 9/1 (ex.: prise en compte de l'expansion de la commande).
Request	Une matrice d'octets représentant les octets de données dans la demande de commande CP 9/1 (ex.: commençant après le nombre total d'octets et finissant avant la somme de contrôle).
Reply	Une matrice d'octets représentant les octets de données dans la réponse de commande CP 9/1 (ex.: commençant après le nombre total d'octets et finissant avant la somme de contrôle).
ServiceId	Le code de transaction de service établit la relation entre la demande de service et la réponse correspondante.
DelayForNextCall	La valeur spécifie un délai, en ms, pour limiter le cycle d'invocation EndTransfer qui ne doit pas être plus rapide que la valeur spécifiée dans la valeur d'argument.
ServiceError	1: OK/fonction démarrée de façon asynchrone, le résultat doit être interrogé avec EndTransfer 0: OK/exécution terminée, REPLY contient le résultat -1: Echec de Transfer/annulé par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de Transfer/aucune relation de communication existante

Argument	Description
	-4: Echec de Transfer/identificateur de relation de communication non valide
	-5: Echec de Transfer/contenu de Request non valide
	-6: Echec de Transfer/format de REPLY non valide

5.6.2.5 GetPublishedData

Les messages de notification d'événements et de salves CP 9/1 représentent des messages non sollicités tels que définis dans l'IEC 62769-7. Le Tableau 19 répertorie les arguments de la Méthode GetPublishedData.

Signature:

```
METHOD BeginGetPublishedData (
    DD_String      CommunicationRelationId,
    unsigned short &Command,
    DD_String      &Reply,
    TIME           &TimeStamp
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD EndGetPublishedData (
    DD_String      CommunicationRelationId,
    unsigned short &Command,
    DD_STRING      &Reply,
    TIME           &TimeStamp
    unsigned long   ServiceId,
    unsigned long   &DelayForNextCall,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

```
METHOD CancelGetPublishedData (
    DD_String      CommunicationRelationId,
    unsigned long   ServiceId,
    long           &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}
```

Tableau 19 – Arguments de la méthode GetPublishedData

Argument	Description
CommunicationRelationId	La valeur d'argument contient le nodeId du ConnectionPoint représentant la connexion entre un appareil et un réseau physique au sein du Modèle d'Information.
Command	Numéro de commande relatif à la commande CP 9/1 publiée. Le Serveur de Communication FDI doit analyser la PDU dans divers formats conformément aux normes de protocole CP 9/1 (ex.: prise en compte de l'expansion de la commande).
Reply	Une matrice d'octets représentant les octets de données dans la réponse de commande CP 9/1 publiée (ex.: commençant après le nombre total d'octets et finissant avant la somme de contrôle).
TimeStamp	Heure à laquelle les données publiées ont été saisies.
ServiceId	Le code de transaction de service établit la relation entre la demande de service et la réponse correspondante.
DelayForNextCall	La valeur spécifie un délai, en ms, pour limiter le cycle d'invocation EndGetPublishedData qui ne doit pas être plus rapide que la valeur spécifiée dans la valeur d'argument.
ServiceError	1: OK/fonction démarrée de façon asynchrone, le résultat doit être interrogé avec EndGetPublishedData 0: OK/exécution terminée, ReceivedData et TimeStamp contiennent le résultat -1: Echec de GetPublishedData/annulé par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de GetPublishedData/non pris en charge -4: Echec de GetPublishedData/aucune relation de communication existante -5: Echec de GetPublishedData/identificateur de relation de communication non valide -8: Echec de GetPublishedData/aucune information de salve publiée

5.6.2.6 SetAddress

Le Tableau 20 répertorie les arguments de la Méthode SetAddress.

Signature:

```

METHOD BeginSetAddress (
    unsigned char  OldPollAddress,
    unsigned char  NewPollAddress,
    unsigned long  ServiceId,
    unsigned long  &DelayForNextCall,
    long          &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD EndSetAddress (
    unsigned long  ServiceId,
    unsigned long  &DelayForNextCall,
    long          &ServiceError);
{
    DEFINITION {<Gateway specific implementation>}
}

METHOD BeginSetAddress (
    unsigned long  ServiceId,
    long          &ServiceError);
{

```

```

    DEFINITION {<Gateway specific implementation>}
}

```

Tableau 20 – Arguments de la méthode SetAddress

Argument	Description
OldPollAddress	La valeur d'argument contient l'adresse actuelle d'un appareil. Les valeurs permises sont 0..63.
NewPollAddress	La valeur d'argument contient la nouvelle adresse d'un appareil. Les valeurs permises sont 0..63.
ServiceId	Le code de transaction de service établit la relation entre la demande de service et la réponse correspondante.
DelayForNextCall	La valeur spécifie un délai, en ms, pour limiter le cycle d'invocation EndTransfer qui ne doit pas être plus rapide que la valeur spécifiée dans la valeur d'argument.
ServiceError	1: OK/fonction démarrée de façon asynchrone, le résultat doit être interrogé avec EndSetAddress 0: OK/exécution terminée avec succès -1: Echec de SetAddress/annulé par l'appelant -2: Echec de l'appel/ID de service inconnu -3: Echec de SetAddress/non initialisé -4: Echec de SetAddress/non connecté à un réseau -5: Echec de SetAddress/aucun appareil correspondant à oldAddress n'a été trouvé -6: Echec de SetAddress/erreur d'adresse en double -7: Echec de SetAddress/l'appareil n'a pas accepté la nouvelle adresse -8: Echec de SetAddress/oldAddress non valide (en termes de syntaxe, de type de données, de format des données, etc.) -9: Echec de SetAddress/newAddress non valide (en termes de syntaxe, de type de données, de format des données, etc.) -10: Echec de SetAddress/impossible à l'état connecté

5.6.2.7 Scan

La signature de méthode spécifiée dans l'IEC 62769-7 s'applique. La logique applicative de la Passerelle HART doit créer le résultat de balayage en utilisant les définitions EDD suivantes qui doivent être incluses dans la définition de COMPONENT des appareils de communication. L'EDD LIST également contenue dans COMPONENT contient la liste des appareils correspondant à la structure de données définie dans le schéma XML. Cette LIST doit être référencée par l'attribut SCAN_LIST du composant d'appareil de communication. Le schéma topologyScanResult correspondant est spécifié à l'Annexe A.

```

VARIABLE DevAddr
{
    LABEL "Address";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(5);
    HANDLING READ & WRITE;
}

```

```

VARIABLE DevMfg
{
    LABEL "Manufacturer";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevType
{
    LABEL "Device Type";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER(2);
    HANDLING READ & WRITE;
}

VARIABLE DevRev
{
    LABEL "Device Revision";
    CLASS DEVICE;
    TYPE UNSIGNED_INTEGER;
    HANDLING READ & WRITE;
}

VARIABLE DevTag
{
    LABEL "Tag";
    CLASS DEVICE;
    TYPE ASCII(32);
    HANDLING READ & WRITE;
}

COLLECTION ScanItemType_TP
{
    MEMBERS
    {
        ADDRESS, DevAddr;
    }
}

COLLECTION DeviceIdentificationType
{
    MEMBERS
    {
        MFG,          DevMfg;
        DEV_TYPE,    DevType;
        DEV_REV,     DevRev;
        TAG,         DevTag;
    }
}

COLLECTION ScanResultType
{
    MEMBERS
    {
        ScanItem,          ScanItemType_TP;
        DeviceIdentification, DeviceIdentificationType;
    }
}

LIST TopologyScanResult
{
    TYPE ScanResultType;
}

```



```
CAPACITY 64;    //Defined to limitation of the comm. device  
}
```

5.6.2.8 ScanNext

La signature de méthode spécifiée dans l'IEC 62769-7 s'applique. La logique applicative de la Passerelle HART doit créer le résultat de balayage en utilisant les mêmes définitions EDD définies en 5.6.2.7. Le schéma topologyScanResult correspondant est spécifié à l'Annexe A.

Annexe A (normative)

Schéma de balayage topologique

A.1 Généralités

Le Schéma du résultat de balayage topologique spécifié à l'Annexe A décrit l'argument de la méthode scan au format spécifique à HART `topologyScanResult`. Le contenu et la structure du document XML doivent correspondre au concept conçu de Modèle d'Information pour décrire une topologie afin de permettre une correspondance générique entre des appareils physiques connectés au réseau et le Modèle d'Information hébergé par le Serveur FDI.

A.2 IdentificationType

Le contenu de l'élément correspond au FunctionalGroup "Identification".

Le schéma XML pour IdentificationType est le suivant:

```
<xs:complexType name="IdentificationT">
  <xs:attribute name="MANUFACTURER_ID" type="xs:unsignedShort"
use="required">
  </xs:attribute>
  <xs:attribute name="DEVICE_TYPE" type="xs:unsignedShort"
use="required">
  </xs:attribute>
  <xs:attribute name="UNIVERSAL_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="DEVICE_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="SERIAL_NUMBER" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedInt">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="16777216"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="HARDWARE_REVISION" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedByte">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="32"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SOFTWARE_REVISION" type="xs:unsignedByte"
use="required">
  </xs:attribute>
  <xs:attribute name="REV_COUNTER" type="xs:unsignedShort"
default="0">
  </xs:attribute>
  <xs:attribute name="TAG" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
```

Les attributs d'IdentificationType sont décrits dans le Tableau A.1.

Tableau A.1 – Attributs d'IdentificationT

Attribut	Description
MANUFACTURER_ID	Numéro d'identification du fabricant.
DEVICE_TYPE	Numéro de modèle associé à la ressource fourni par le fabricant.
UNIVERSAL_REVISION	Révision universelle HART associée à l'appareil.
DEVICE_REVISION	Numéro de révision d'appareil associé à l'appareil fourni par le fabricant.
SERIAL_NUMBER	Numéro de série unique de l'appareil.
HARDWARE_REVISION	Révision de matériel du fabricant associée à l'appareil.
SOFTWARE_REVISION	Révision de micrologiciel ("firmware") du fabricant associée à l'appareil.
REV_COUNTER	Compteur de changements de configuration associé à l'appareil.
TAG	Marqueur d'Identification configuré associé à l'appareil.

A.3 AddressTypeTP

Le schéma XML pour AddressTypeTP est le suivant:

```
<xs:complexType name="AddressTypeTP">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:element name="DevPollAddr" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="63"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Les éléments d'AddressTypeTP sont décrits dans le Tableau A.2.

Tableau A.2 – Eléments d'AddressTypeTP

Elément	Description
DevAddr	Contient l'adresse HART de cinq octets de l'appareil associé.
DevPollAddr	Contient l'adresse courte HART pour l'interrogation de l'appareil associé.

A.4 AddressTypeIP

Le schéma XML pour AddressTypeIP est le suivant:

```
<xs:simpleType name="IPv4T">
  <xs:restriction base="xs:string">
    <xs:pattern value="((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)")"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="IPv6T">
  <xs:restriction base="xs:string">
    <xs:pattern value="([A-Fa-f0-9]{1,4}:){7}[A-Fa-f0-9]{1,4})"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="AddressTypeIP">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:choice>
          <xs:element name="IPv4Address" type="hart:IPv4T">
          </xs:element>
          <xs:element name="IPv6Address" type="hart:IPv6T">
          </xs:element>
        </xs:choice>
        <xs:element name="IPPort" type="xs:unsignedShort">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Les éléments d'AddressTypeIP sont décrits dans le Tableau A.3.

Tableau A.3 – Eléments d'AddressTypeIP

Elément	Description
DevAddr	Contient l'adresse HART de cinq octets de l'appareil associé.
IPAddress	Contient l'adresse IP de réseau de l'appareil associé.
IPPort	Contient le numéro de port IP utilisé pour l'appareil associé.

A.5 AddressTypeTDMA

Le schéma XML pour AddressTypeTDMA est le suivant:

```
<xs:complexType name="AddressTypeTDMA">
  <xs:complexContent>
    <xs:extension base="hart:LongAddressT">
      <xs:sequence>
        <xs:element name="NetworkID" type="xs:unsignedShort">
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Les éléments d'AddressTypeTDMA sont décrits dans le Tableau A.4.

Tableau A.4 – Eléments d'AddressTypeTDMA

Elément	Description
DevAddr	Contient l'adresse HART de cinq octets de l'appareil associé.
IPAddress	Contient l'adresse IP de réseau de l'appareil associé.
IPVersion	Contient le numéro de version IP utilisé pour l'appareil associé (version 4 ou 6).
IPPort	Contient le numéro de port IP utilisé pour l'appareil associé.

A.6 AddressType

Le schéma XML pour AddressType est le suivant:

```
<xs:complexType name="AddressT">
  <xs:choice>
    <xs:element name="AddressTP" type="hart:AddressTypeTP">
    </xs:element>
    <xs:element name="AddressIP" type="hart:AddressTypeIP">
    </xs:element>
    <xs:element name="AddressTDMA" type="hart:AddressTypeTDMA">
    </xs:element>
  </xs:choice>
</xs:complexType>

<xs:complexType name="LongAddressT">
  <xs:sequence>
    <xs:element name="DevAddr">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:pattern value="([A-Fa-f0-9]{10})"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Les éléments d'AddressType sont décrits dans le Tableau A.5.

Tableau A.5 – Eléments d'AddressT

Elément	Description
AddressTP	Les adresses de passage de jeton contiennent les informations d'adresse utilisées pour les ConnectionPointType_HART_TP5, 6 ou 7 décrits en 5.5.1.2.
AddressIP	Les adresses IP contiennent les informations d'adresse décrites dans le ConnectionPointType_HART_IP décrit en 5.5.1.2.
AddressTDMA	Les adresses TDMA contiennent les informations d'adresse décrites dans le ConnectionPointType_HART_TDMA décrit en 5.5.1.2.

A.7 ConnectionPointType

Le schéma XML pour ConnectionPointType est le suivant:

```
<xs:complexType name="ConnectionPointT">
  <xs:sequence>
    <xs:element name="Identification" type="hart:IdentificationT">
    </xs:element>
    <xs:element name="Address" type="hart:AddressT">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Les éléments d'un ConnectionPointType sont décrits dans le Tableau A.6

Tableau A.6 – Eléments de ConnectionPointT

Elément	Description
Identification	Les données de l'élément contiennent les données d'identification du type d'appareil. En comparaison au Modèle d'Information (IEC 62769-5), le ConnectionPoint ne contient pas les données d'identification du type d'appareil et ne s'y réfère pas non plus. Mais pour aider le système hôte FDI à trouver le paquetage qui correspond à l'appareil connecté, ce schéma associe l'identification du type d'appareil avec le ConnectionPoint.
Address	L'élément d'adresse contient les informations spécifiques utilisées pour adresser de façon unique l'appareil dans le réseau. L'adresse sera l'un des trois types possibles décrits par AddressType.

A.8 NetworkType

Le type d'élément décrit le résultat de balayage complet pour un seul réseau, à cause de la méthode scan qui est fournie par instance d'un "Appareil de Communication" qui existe en relation biunivoque (1:1) avec une instance de réseau.

Le schéma XML pour un NetworkType est le suivant:

```
<xs:complexType name="NetworkT">
  <xs:sequence>
    <xs:element name="ConnectionPoint" type="hart:ConnectionPointT"
      maxOccurs="unbounded">
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Les éléments d'un NetworkType sont décrits dans le Tableau A.7

Tableau A.7 – Eléments de NetworkT

Elément	Description
ConnectionPoint	L'élément ConnectionPoint contient l'adresse et l'identification de l'appareil connecté au réseau qui a été détecté au cours des opérations de balayage de bus.

A.9 Network

La hiérarchie d'éléments suivante est utilisée pour retourner le résultat de balayage en fonction de la représentation de Modèle d'Information (IEC 62769-5) d'une topologie.

Le schéma XML pour un élément Network est le suivant:

```
<xs:element name="Network" type="hart:NetworkT">  
</xs:element>
```


Annexe B (normative)

Paramètres du service Transfer

B.1 Généralités

Les Services d'Accès Direct spécifiés dans l'IEC 62769-2 permettent au Plugiciel d'Interface Utilisateur (User Interface Plug-in, UIP) d'échanger directement des données avec l'appareil. L'échange direct des données signifie que les données échangées entre un appareil et un UIP peuvent ne pas être reflétées dans le Modèle d'Information. L'interface IDirectAccess définie dans l'IEC 62769-6 correspond aux Services d'Accès Direct définis dans l'IEC 62769-2. Les fonctions BeginTransfer et EndTransfer définies par l'interface IDirectAccess ont besoin d'acheminer des informations spécifiques à un protocole. Le schéma XML ci-après documente les spécificités de protocole.

B.2 receiveData

L'élément décrit ci-après contient les données qui sont retournées par l'intermédiaire de la valeur de retour définie par les fonctions EndTransfer et GetPublishedData de l'interface IDirectAccess.

```
<xs:element name="receiveData" type="hart:TransferResultDataT">
</xs:element>
```

B.3 sendData

L'élément décrit ci-après contient les données qui doivent être présentées par l'intermédiaire de l'argument sendData défini par la fonction BeginTransfer de l'interface IDirectAccess.

```
<xs:element name="sendData" type="hart:TransferSendDataT">
</xs:element>
```

B.4 TransferResultDataT

Le type d'élément XML définit le format des données de paramètres de service qui doit être appliqué à la valeur de retour définie par EndTransfer et GetPublishedData. Les noms d'attributs définis dans TransferResultDataType correspondent aux mêmes noms d'arguments de la méthode Transfer spécifiés en 5.6.1.4 et 5.6.2.4.

```
<xs:complexType name="TransferResultDataT">
  <xs:attribute name="COMMAND" type="xs:unsignedInt"
    use="required">
  </xs:attribute>
  <xs:attribute name="REPLY" type="xs:hexBinary" use="required">
  </xs:attribute>
</xs:complexType>
```

Tableau B.1 – Attributs de TransferResultDataT

Attribut	Description
COMMAND	L'attribut correspond au numéro de commande CP9/1.
REPLY	L'attribut correspond aux octets de données REPLY dans le champ de données de commande CP9/1.

B.5 TransferSendDataT

Le type d'élément XML définit le format des données de paramètres de service qui doit être appliqué à l'argument sendData défini par BeginTransfer. Les noms d'attributs définis dans TransferSendDataType correspondent aux mêmes noms d'arguments de la méthode Transfer spécifiés en 5.6.1.4 et 5.6.2.4.

```
<xs:complexType name="TransferSendDataT">
  <xs:attribute name="COMMAND" type="xs:unsignedInt"
    use="required">
  </xs:attribute>
  <xs:attribute name="REQUEST" type="xs:hexBinary" use="required">
  </xs:attribute>
</xs:complexType>
```

Tableau B.2 – Attributs de TransferSendDataT

Attribut	Description
COMMAND	L'attribut correspond au numéro de commande CP9/1.
REQUEST	L'attribut correspond aux octets de données REQUEST dans le champ de données de commande CP9/1.

Bibliographie

IEC 61784-1, *Réseaux de communication industriels – Profils – Partie 1: Profils de bus de terrains*

IEC 61784-2, *Réseaux de communication industriels – Profils – Partie 2: Profils de bus de terrain supplémentaires pour les réseaux en temps réel basés sur l'ISO/IEC 8802-3*

IEC 61804 (toutes les parties),

IEC 61804 (toutes les parties), *Blocs fonctionnels (FB) pour les procédés industriels et le langage de description d'appareil électronique (EDDL)*

IEC 62769-1, *Intégration des appareils de terrain (FDI) – Partie 1: Vue d'ensemble*

NOTE 1 L'IEC 62769-1 est techniquement identique à la FDI-2021.

IEC 62769-2, *Intégration des appareils de terrain (FDI) – Partie 2: Client FDI*

NOTE 2 L'IEC 62769-2 est techniquement identique à la FDI-2022.

IEC 62769-3, *Intégration des appareils de terrain (FDI) – Partie 3: Serveur FDI*

NOTE 3 L'IEC 62769-3 est techniquement identique à la FDI-2023.

IEC 62769-6, *Intégration des appareils de terrain (FDI) – Partie 6: Mapping de technologies FDI*

NOTE 4 L'IEC 62769-6 est techniquement identique à la FDI-2026.

HCF_SPEC-13, *HART® Field Communication Protocol Specification*, disponible en anglais seulement à l'adresse <www.hartcomm.org>

HCF_SPEC-99, *Command Summary Specification*, disponible en anglais seulement à l'adresse <www.hartcomm.org>

HCF_SPEC-127, *Universal Command Specification*, disponible en anglais seulement à l'adresse <www.hartcomm.org>

HCF_SPEC-151, *Common Practice Command Specification*, disponible en anglais seulement à l'adresse <www.hartcomm.org>

IETF RFC 768, *User Datagram Protocol*, disponible en anglais seulement à l'adresse <<http://www.ietf.org>>

IETF RFC 793, *Transmission Control Protocol*, disponible en anglais seulement à l'adresse <<http://www.ietf.org>>

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch