IEC 62453-2

Edition 1.0 2009-06

INTERNATIONAL STANDARD

Field device tool (FDT) interface specification – Part 2: Concepts and detailed description

IEC 62453-2:2009(E)

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2009 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.

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

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.

- Catalogue of IEC publications: www.iec.ch/searchpub

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

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

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 62453-2

Edition 1.0 2009-06

INTERNATIONAL STANDARD

Field device tool (FDT) interface specification – Part 2: Concepts and detailed description

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE XG

ICS 25.040.40; 35.100.05; 35.110

ISBN 2-8318-1050-4

® Registered trademark of the International Electrotechnical Commission

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

CONTENTS

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 2: Concepts and detailed description

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62453-2 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This part, in conjunction with the other parts of the first edition of the IEC 62453 series cancels and replaces IEC/PAS 62453-1, IEC/PAS 62453-2, IEC/PAS 62453-3, IEC/PAS 62453-4 and IEC/PAS 62453-5 published in 2006, and constitutes a technical revision.

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

FDIS	Report on voting
65E/124/FDIS	65E/137/RVD

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

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

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

A list of all parts of the IEC 62453 series, under the general title *Field Device Tool (FDT) interface specification,* can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result 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.

A bilingual version of this publication may be issued at a later date.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

INTRODUCTION

This part of IEC 62453 is an interface specification for developers of FDT (Field Device Tool) components for function control and data access within a client/server architecture. The specification is a result of an analysis and design process to develop standard interfaces to facilitate the development of servers and clients by multiple vendors that need to interoperate seamlessly.

With the integration of fieldbusses into control systems, there are a few other tasks which need to be performed. In addition to fieldbus- and device-specific tools, there is a need to integrate these tools into higher-level system-wide planning- or engineering tools. In particular, for use in extensive and heterogeneous control systems, typically in the area of the process industry, the unambiguous definition of engineering interfaces that are easy to use for all those involved is of great importance.

A device-specific software component created according to this standard is called Device Type Manager (DTM). It integrates all device–specific data, functions and business rules into the system via the FDT services defined herein.

The FDT/DTM approach is open for all kind of fieldbusses and enables integration variety of devices into heterogeneous systems.

Figure 1 shows how IEC 62453-2 is aligned in the structure of the IEC 62453 series.

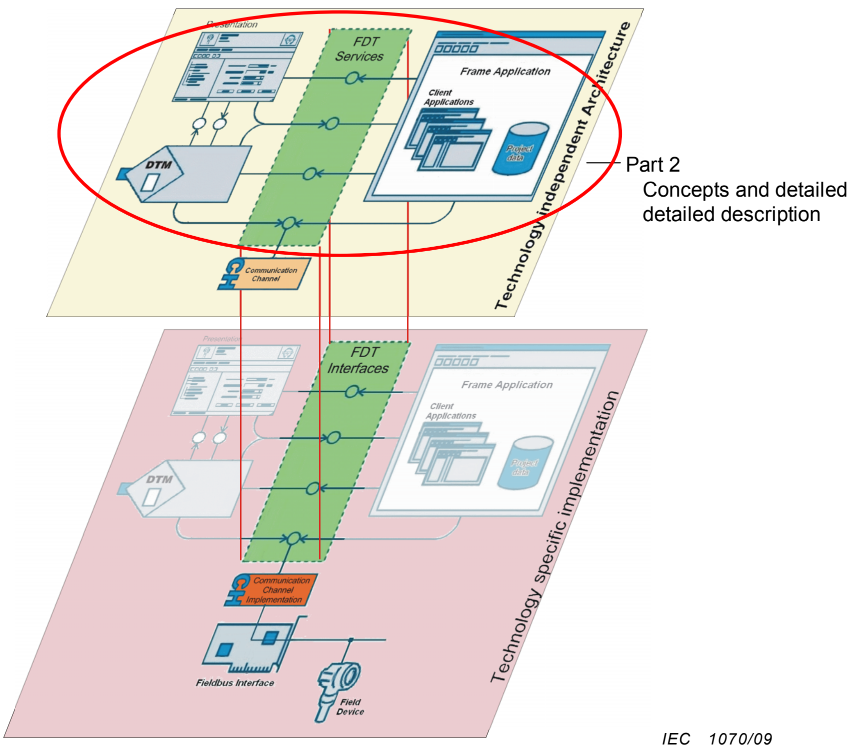


Figure 1 – Part 2 of the IEC 62453 series

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 2: Concepts and detailed description

1 Scope

This part of IEC 62453 explains the common principles of the field device tool concept. These principles can be used in various industrial applications such as engineering systems, configuration programs and monitoring and diagnostic applications.

This standard specifies the general objects, general object behavior and general object interactions that provide the base of FDT.

2 Normative references

The following referenced documents are indispensable for the application of this specification. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131 (all parts), *Programmable controllers*

IEC/TR 62390, *Common automation device – Profile guideline*

IEC 62453-1:2009, *Field Device Tool (FDT) interface specification – Part 1: Overview and guidance*

IEC 62453-3xy (all parts):2009, *Field Device Tool (FDT) interface specification – Part 3xy: Communication profile integration*

IEC/TR 62453-41:2009, *Field Device Tool (FDT) interface specification* **–** *Part 41: Object model integration profile – Common object model*

ISO/IEC 19501:2005, *Information technology – Open Distributed Processing – Unified Modeling Language (UML)* Version 1.4.2

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62453-1 and the following apply.

3.1.1
FDT version
implementation version defined by the related technology specific organization

NOTE The FDT version is specified in IEC/TR 62453-41.

3.1.2
monolithic DTM
is one single DTM that represents the complete device with all its modules

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

NOTE There are also other concepts representing modules of a device in this specification, for example Module DTM and BTM

3.2 Symbols and abbreviated terms

For the purposes of this document, the symbols and abbreviations given in IEC 62453-1 and the following apply.

DD	Device description
OODMS	Object Oriented Database Management System
RDBMS	Relational Database Management System

3.3 Conventions

3.3.1 State availability statement

The state availability statement uses [_] and [X] for stating the availability of a service in a specific state. The expressions have the following meaning:

[_] '<name of state>'	service is not available in this state;
[X] '<name of state>'	service is available in this state.

3.3.2 Data type names and references to data types

The conventions for naming and referencing data types are explained in Clause A.1

4 Fundamentals

4.1 General

This clause introduces the FDT model and covers topics which are specific to the requirements of field device integration.

4.2 Abstract FDT model

4.2.1 FDT model overview

Figure 2 provides an overview of the objects defined in FDT and their relationships to each other

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

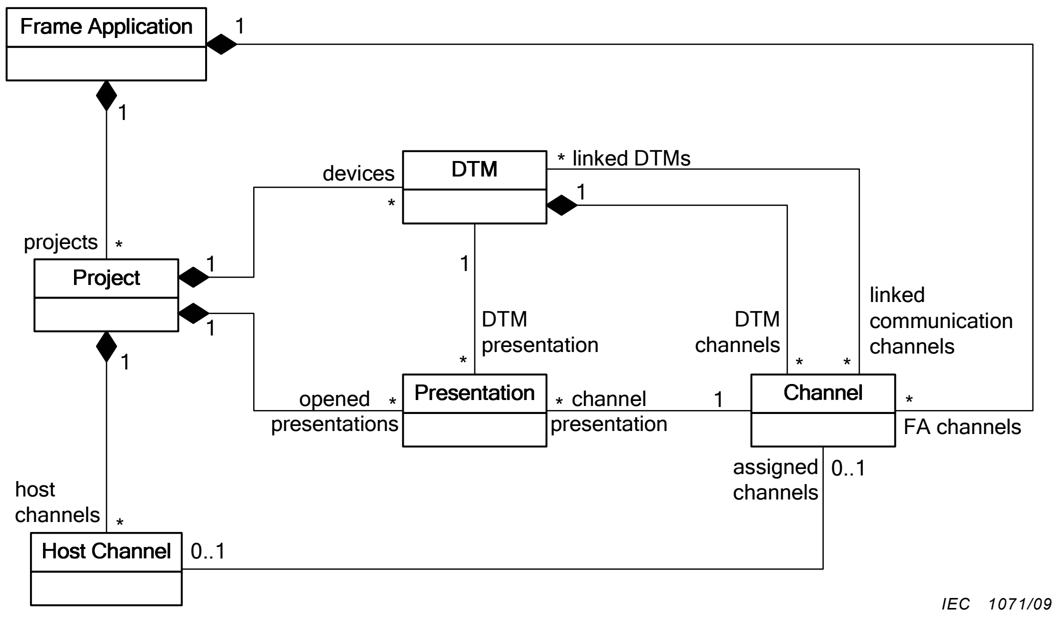


Figure 2 – Abstract FDT model

Table 1 contains brief descriptions of all objects. More detail description of objects and related services are provided in subsequent chapters.

The FDT objects may be executed all on one platform or in a distributed system.

For data exchange and for the interaction between objects, the data types as defined in Annex A are used. The concrete implementation of these data types is defined in the IEC 62453-4z documents describing the mapping to specific implementation technologies.

Table 1 – Description of FDT objects

Object	Description
Frame Application	The Frame Application is a logical object to represent an environment like an engineering system or a standalone tool (see 4.2.2). It controls the lifetime of DTM-instances within the project. A Frame Application may handle several projects
Project	The Project belongs to the Frame Application. The Project is a logical object describing the management of device-instances. It controls the lifetime and dataset of device-instances within a Frame Application. FDT does not define any services for the Project, since it is a pure Frame Application internal object

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Object	Description
HostChannel	A HostChannel belongs to the Frame Application. The HostChannel is a logical object representing a part of a function of a host system, such as DCS or PLC. It is required when additional information for the processing of device I/O data is needed, e.g. if DTM or BTM data refers to more than one I/O value. For example one Octet often is used to group the state values of eight digital I/O points. In these applications, the HostChannel object provides an association between contents of the channel data and individual process I/O points. To make the cyclic I/O data available within a Frame Application, there shall be an association between the I/O function blocks and the process values of a device. The inputs and outputs of I/O function blocks are represented by HostChannels, the process value is represented by a Channel. The association between HostChannel and Channel is called channel assignment. FDT does not define any services for the Host Channel, since it is a pure Frame Application internal object
DTM	A DTM is a software component containing the device specific application software. It encapsulates all device–specific data, functions and business rules. A DTM is generally not stand-alone executable. It needs a Frame Application (see 4.2.2) that acts as the runtime environment. A DTM is typically developed by the device manufacturer and shipped together with the devices. It depends on the software design of the DTM, whether it can be used for one specific device type or a group of different device types or a device type family. Each device installed on a plant is represented by a DTM object. Therefore one or several DTM objects are needed to handle the different devices. The DTMs are installed in the system, so that the system can be dynamically extended by installing new DTMs for new devices. A DTM may represent different types of devices, e.g. field devices, communication devices or gateway devices. (see 4.2.3)
BTM	A BTM (see 4.2.3.6) is used to represent flexible software objects within a device, like function blocks. These software blocks are more flexible than software modules, for instance they may be moved between devices. Also a protocol may require modeling of device structures as blocks.
Presentation	Presentation objects (see 4.2.4) represent a (visual) user interface. The Presentation object belongs to the DTM, to the BTM or to the Channel.
Channel	Channel-Object could behave in three ways: As a ‘Communication-Channel’, a ‘Process-Channel’ and a combination of both (see 4.2.5).

All the associations shown in Figure 2 above are explained in Table 2 below.

Table 2 – Description of associations between FDT objects

Association	Description
assigned channels	To make the cyclic I/O data available within a Frame Application there has to be an association between the I/O function blocks and the process values of a device. The I/O function blocks are represented by a host channel, the process value by a channel. The Frame Application (project) is responsible for handling this association. <u>Use cases:</u> • channel assignment <u>Services:</u> Informational services • DTM service: GetChannels • channel service: ReadChannelInformation Management services • channel service: WriteChannelInformation Depending services • proprietary DTM services for channel management

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Association	Description
channel presentation	The instances of presentation objects are associated to the instance of their DTM business object by this relationship. <u>Use cases:</u> • all channel use cases with DTM specific GUI <u>Services:</u> Informational services (see also 7.3) • channel service: GetChannelFuntions • channel service: GetGuiInformation Depending services • frame service: OpenPresentationRequest • frame service: ClosePresentationRequest
devices	A project holds the list of DTM instances by the association devices. Releasing the project results in a release of all associated DTM instances. A DTM instance cannot be part of more than one project. Use cases: • system generation • system planning <u>Services:</u> Informational services • frame service: GetChildNodes Management services • services related to sub-topology management see 7.6.2 • ValidateAddChild • ChildAdded • ValidateRemoveChild • ChildRemoved • FA services related to topology management
DTM channels	A DTM can have a list of channels, which are associated by DTM channels. A channel is part of a DTM. <u>Services:</u> Informational services • frame service: GetChannels Management services • There are no services to manipulate this association. The lifetime of channel instances is controlled by a DTM
DTM presentation	The presentation object instances are associated with their DTM instance by this relationship. <u>Use cases:</u> • All DTM use cases with DTM specific GUI <u>Services:</u> • open presentation • close presentation
FA channels	The Frame Application (project) may have a list of Communication Channels. There are no services to manipulate this association. See 4.2.2
host channels	The FA maps the Process Channels of DTMs to internal IO management of the projects
linked communication channels	The topology of DTMs is shown with this association. Within the topology tree a DTM object is the child node of a channel. The association shows the connection from a device to a channel. The Frame Application (project) is responsible for handling this association. <u>Services:</u> • linked channels can be explored by GetParentNodes • SetLinkedCommunicationChannel • ReleaseLinkedCommunicationChannel

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Association	Description
linked DTMs	The topology of field devices is shown with this association. Within the topology tree a channel object is the parent node for a device. The association shows the connection from a channel to a device. The Frame Application (project) is responsible for handling this association. <u>Services:</u> • linked DTMs can be explored by GetChildNodes • SetLinkedCommunicationChannel • ReleaseLinkedCommunicationChannel
opened presentations	Open presentation objects are linked by this association to projects. The Frame Application (project) is responsible for handling this association
projects	A Frame Application can create and manage multiple projects, which are connected by the “projects” association. The Frame Application (project) is responsible for handling this association

4.2.2 Frame Application (FA)

The Frame Application is the runtime environment for the DTMs. It provides the services described in 7.7 which may be used by the hosted DTMs.

If the Frame Application has built-in communication capabilities, then it is able to provide Communication Channels (see 4.2.5.3) which provide the services to access the fieldbus. In this case it has full control over communication and is able perform low-level actions such as monitoring or filtering. Frame Applications without build-in communication capabilities are using Communication DTMs (see 4.2.3.2). Figure 3 shows the relation between the Frame Application and Communication Channel object.

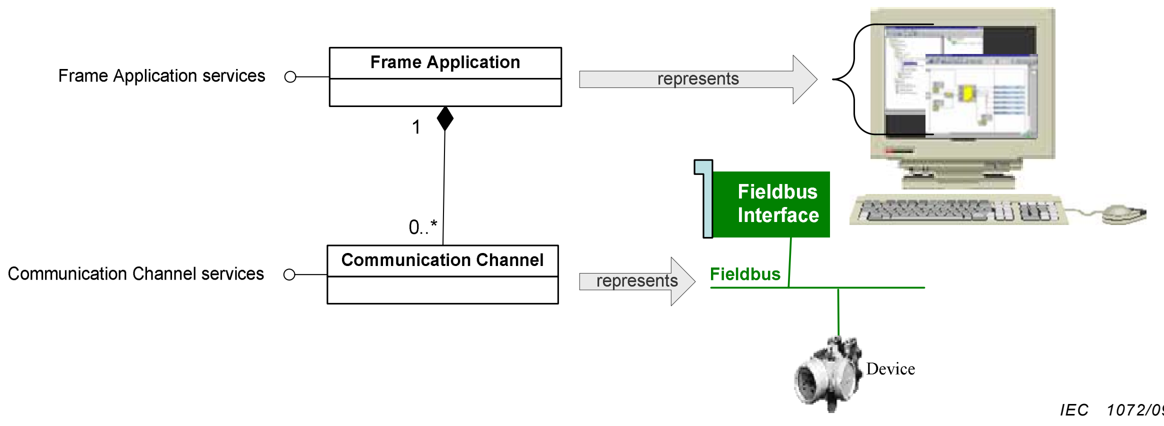


Figure 3 – Frame Application with integrated Communication Channel

The Frame Application’s channel objects represent the gateway from the FDT-specific to the Frame Application-specific communication. On the Frame Application’s side the Communication Channel may be a PC I/O board or engineering topology with processing units and proprietary bus systems.

4.2.3 Device Type Manager (DTM)

4.2.3.1 DTM Overview

The DTM provides the services described in 7.2 (see Figure 4). From a Frame Application’s point of view, all management and task-related interactions with the device functionality are available via these services.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

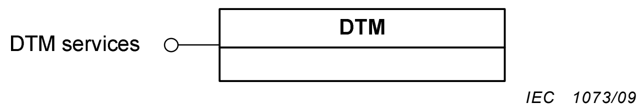


Figure 4 – Device Type Manager (DTM)

A DTM may represent the device structure with data type net:DtmDeviceInstanceTopology. This information may be retrieved by service GetActiveTypeInfo (see 7.2.3.4). The structure of the device may be described by a list of net:Module. Each module may have a number of properties, including configuration data, Process Channels and Communication Channels.

A DTM exposes a list of supported device types with the data type fdt:DtmDeviceTypes (see 4.7.1).

Different types of devices, for example, communication devices, field devices, modules, gateways or blocks are represented by different types of DTMs.

4.2.3.2 Communication DTM (COMM-DTM)

A Communication DTM is a special type of DTM containing the application software for specific communication hardware. For example, it may support configuration settings like baud-rate, start and stop bits, etc.

A Communication DTM shall provide Communication Channels (see 4.2.5.3) which provide the services to access the fieldbus (see Figure 5).

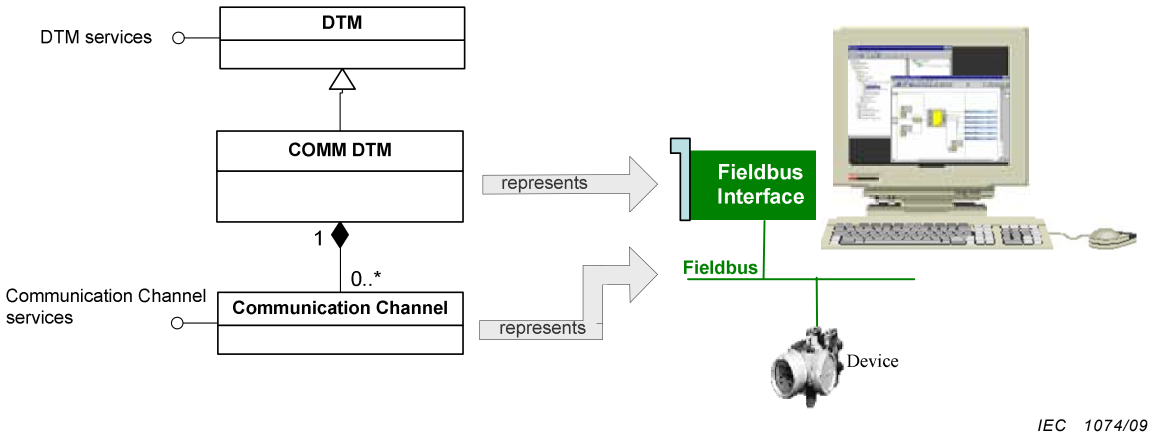


Figure 5 – Communication DTM

The advantage of Communication Channels provided by a DTM compared to the channels provided by a Frame Application (see 4.2.2) is, that they may be integrated into a Frame Application and are thus extending the number of protocols the system is able to access. Both ways of providing Communication Channels may coexist in a Frame Application.

4.2.3.3 Device DTM

A Device DTM is a special type of DTM containing the application software for a 'normal' field device, for example, a pressure transmitter or a valve. Such a DTM for example supports functions to configure and parameterize the device (see Figure 6).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

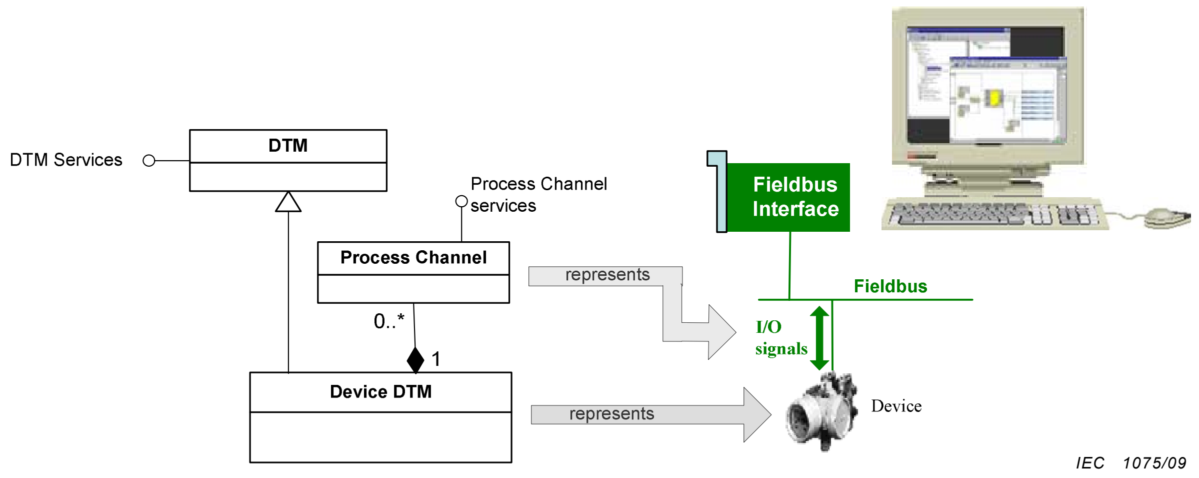


Figure 6 – Device DTM

If it is required to make the device's I/O signals or process values accessible to the host system, then the Device DTM shall provide Process Channel objects (see 4.2.5.2).

4.2.3.4 Gateway DTM

A Gateway DTM is a special type of DTM containing the application software for a device linking fieldbusses (see Figure 7).

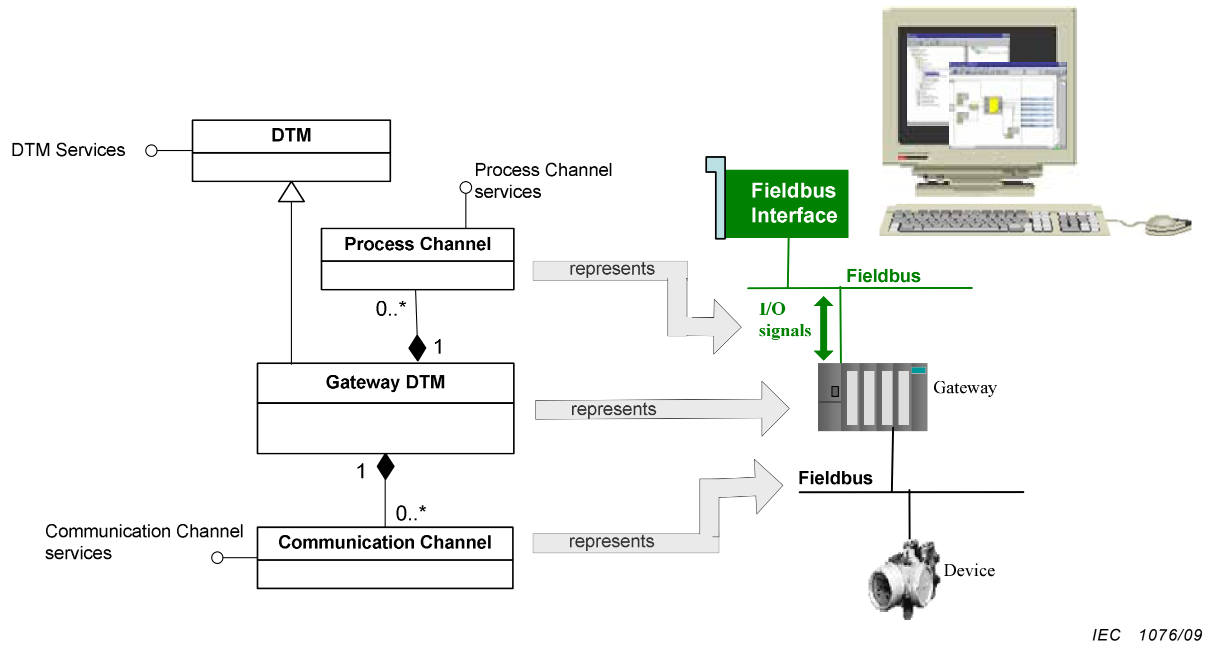


Figure 7 – Gateway DTM

Such a DTM can be seen as a combination of a Communication DTM and a Device DTM. It shall provide Communication Channels (see 4.2.5.3) to provide access to the linked fieldbus and Process Channels (see 4.2.5.2) for the fieldbus where it is connected to, if it is required to make the gateway I/O signals or process values accessible to the host system.

4.2.3.5 Module DTM

A Module DTM is a special type of DTM containing the application software for a device hardware or software module (see Figure 8).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Some devices are subdivided into modules and sub-modules. A module is either a hardware module or a software module. Hardware modules are plugged into physical slots of the device. For example, an I/O module can be plugged into a Remote I/O device. Such a modular device may be represented by several DTMs as shown in the following Figure 8.

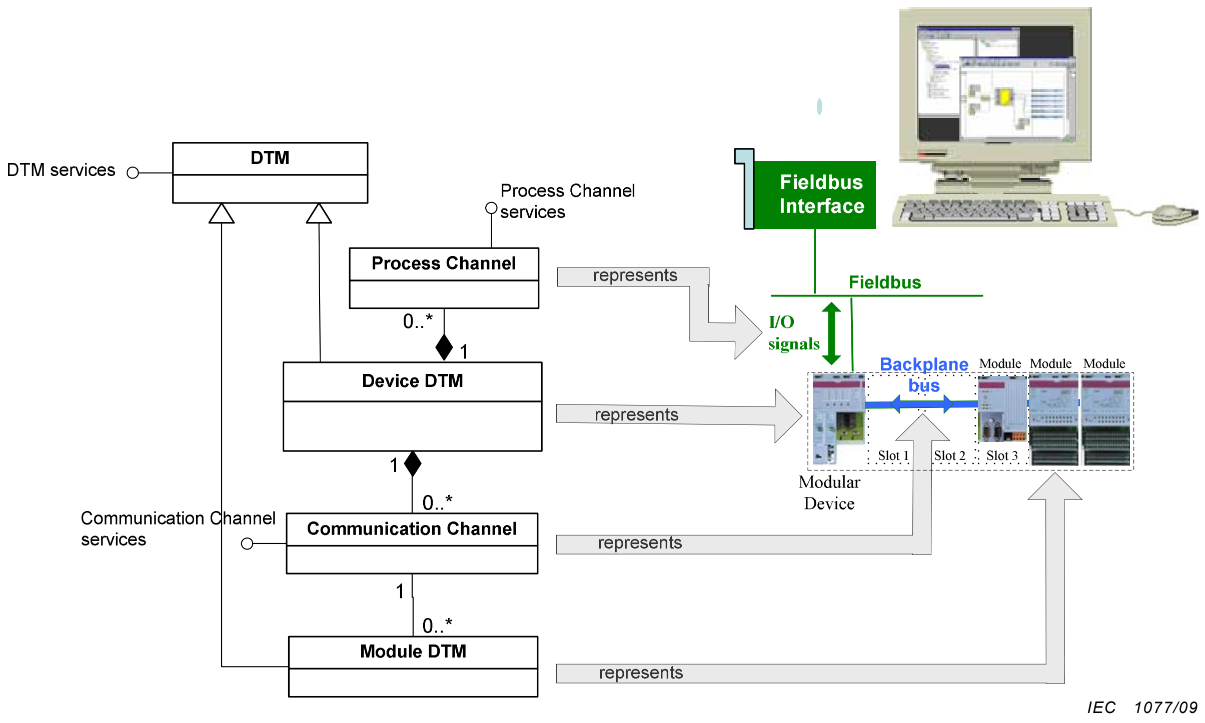


Figure 8 – Module DTM

A Device or Gateway DTM is the root element for complete device representation. This DTM shall provide Communication Channels (see 4.2.5.3) which provide services to access the backplane bus that connects the different modules and sub-modules. Dependent on the software design of the DTM, one channel may be provided for representation of the whole backplane bus or multiple channels for representation of the slots where the modules are plugged in.

The modules are represented by Module DTMs. The Device (or Gateway) DTM and Module DTMs are generally shipped together and supplied by the modular device vendor.

A Module DTM is connected to the Device (or Gateway) DTM via its Communication Channel. The Module DTM is able to communicate with its module via the services provided by the Communication Channel. This concept is called ‘Nested Communication” and is further described in 4.6.

The Module DTM itself shall provide own Communication Channels, if it is representing a module that is the access point to another (linked) fieldbus.

4.2.3.6 Block Type Manager (BTM)

A BTM is a special type of DTM containing the application software for accessible objects inside a device such as function blocks and resource blocks (see Figure 9).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

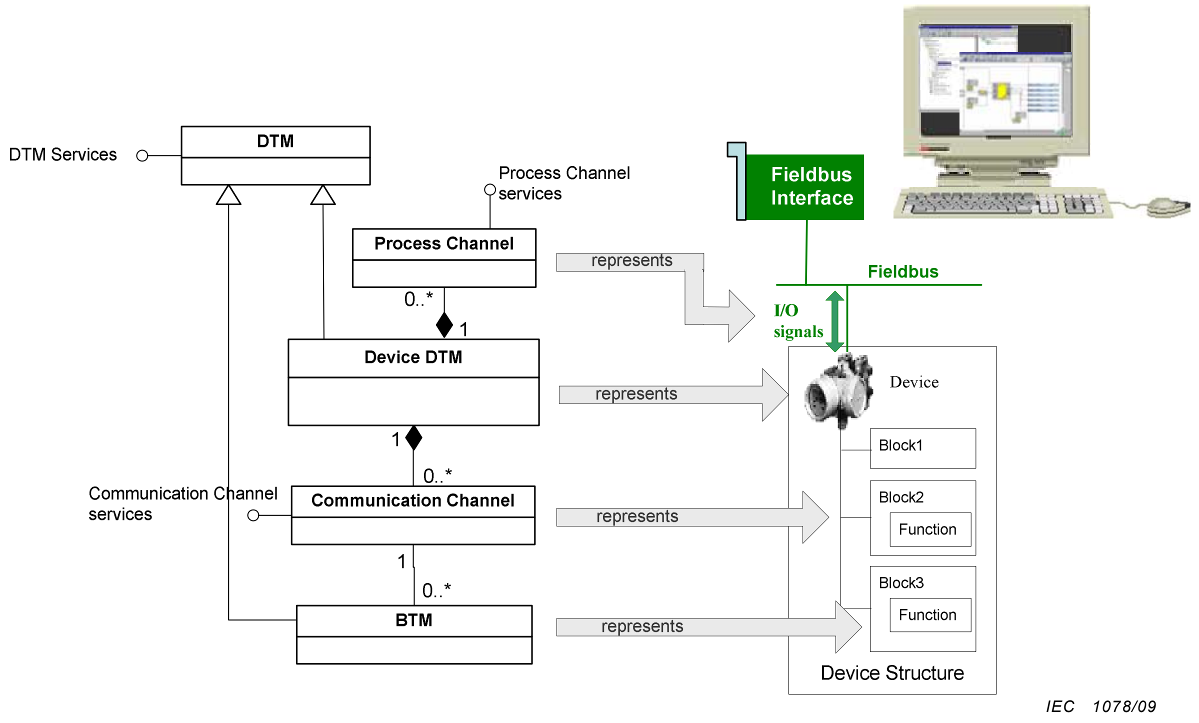


Figure 9 – Block Type Manager (BTM)

A Device or Gateway DTM is the root element for complete device representation. This DTM shall provide Communication Channels (see 4.2.5.3) which provide services to access the block information within the device.

The blocks are represented by BTMs. The software blocks are more flexible than device software modules (see 4.2.3.5), for instance standardized blocks may be moved between devices. The Device (or Gateway) DTM and the BTMs for device specific blocks are generally shipped together and supplied by the device vendor. BTMs for standardized blocks may be supplied by a 3rd party vendor.

A BTM is connected to the Device (or Gateway) DTM via its Communication Channel. The BTM is able to interact with its block via the services provided by the Communication Channel. This concept is called ‘Nested Communication” and is further described in 4.6.

The BTM concept is independent from fieldbus protocol. However, a protocol may require modeling of device structures as blocks. Thus, whether and how the BTM concept is used to model device structures is defined by the IEC 62453-3xy documents describing the protocol profile integration in FDT.

4.2.4 Presentation object

Each of the device specific objects (DTM, BTM and Communication Channel) may be packaged together with Presentation objects which support the services defined in 7.3 (see Figure 10)

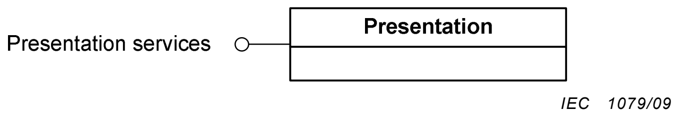


Figure 10 – Presentation object

These presentation objects are implementation specific objects that provide a graphical user interface for the business object.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The presentation object may be a type of object, that allows integration into the GUI of the FA or it may be an object that runs outside of the GUI of the Frame Application (e.g. a standalone executable).

4.2.5 Channel object

4.2.5.1 Channel overview

There are three different types of channels, the Communication Channels the Process Channels and combinations of them as shown in the Figure 11.

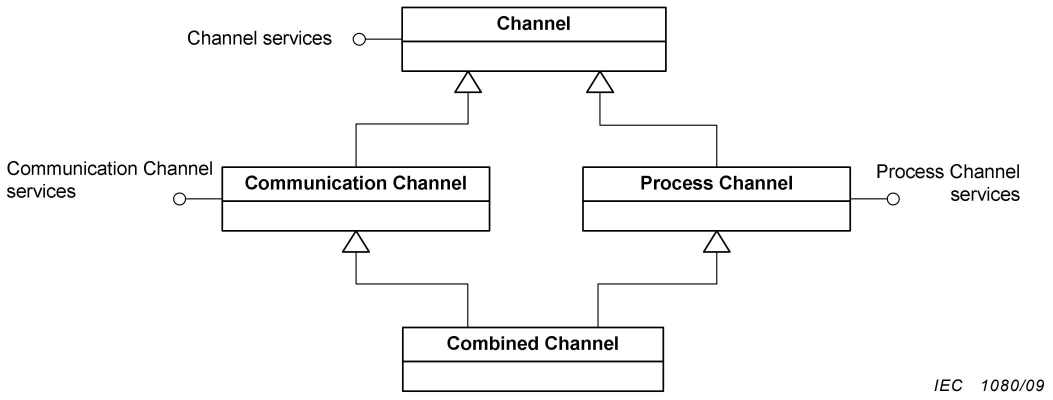


Figure 11 – Channel object

4.2.5.2 Process Channel

Input and output signals provided by devices are created and integrated into the function planning of the control system. If the device provides I/O signals the associated DTM shall offer information about I/O signals of its device. This information includes addressing, signal and data type and is conveyed by Process Channels. The information does not include the current I/O signal values.

A Process Channel supports the services defined in 7.5. These services for example provide access to I/O signal information like addressing, signal and data type, but not to the current I/O signal value itself.

The I/O signal information provided by the Process Channel is protocol specific. Which information shall be provided by Process Channels is defined by the IEC 62453-3xy document describing the protocol profile integration in FDT.

The number and type of I/O signals may depend on the configuration of the device. Thus the number of Process Channels and provided information may also dependent on the device configuration made by the user. A Frame Application is able to inform the DTM by setting the ProtectedByChannelAssignent parameter via service WriteChannelData (see 7.5.1.2) that further changes shall be prohibited, because channel is already integrated into functional planning of the control system (HostChannel is assigned). In this case, DTM shall not accept any changes related to this channel.

4.2.5.3 Communication Channel

A Communication Channel represents the entry point to a fieldbus, a vendor specific backplane bus or point-to-point connection. It may even represent a logical communication with blocks within a device. A Communication Channel belongs to a Frame Application (see 4.2.2) or a DTM (see 4.2.3).

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

A Communication Channel provides communication hardware independent services. In general, the protocol specific services are one to one mapped to the services provided by the channel. The services may be used by the Frame Application or a DTM to exchange data with a connected device or to initiate a function (e.g. identification request, device reset, broad-cast, etc.). The general communication concept is further described in 4.6.

A Communication Channel shall be able to handle several connections to the same device and may be responsible for connections to different devices. It guarantees that a link to a device is established as a peer-to-peer connection. The services that shall be provided by a Communication Channel are defined in 7.6. The pure communication related services (see 7.6.1) just define the frame for interaction with a connected device. The data (parameters) which shall be passed or returned by the service is defined by the IEC 62453-3xy documents describing the protocol profile integration in FDT.

In case of vendor specific protocols (e.g. backplane bus or point-to-point) the pure communication related services (see 7.6.1) may be replaced by vendor specific services. Following this approach only DTMs supplied by this vendor are able to communicate with each other. Therefore a vendor specific bus category (see 4.3) shall be defined. For vendor-independent fieldbus protocols an IEC 62453-3xy document describing the protocol profile integration in FDT shall be provided and standard communication services shall be used.

4.2.5.4 Combined Process and Communication Channel

A combined Process and Communication Channel typically belongs to a Gateway DTM (see 4.2.3.4). It represents the I/O signal that is exposed via the fieldbus where the gateway device is connected to which corresponds to the cyclic communication on the linked fieldbus. At the same time such a channel represents the entry point to the linked fieldbus (or point-to-point connection) as shown in Figure 12.

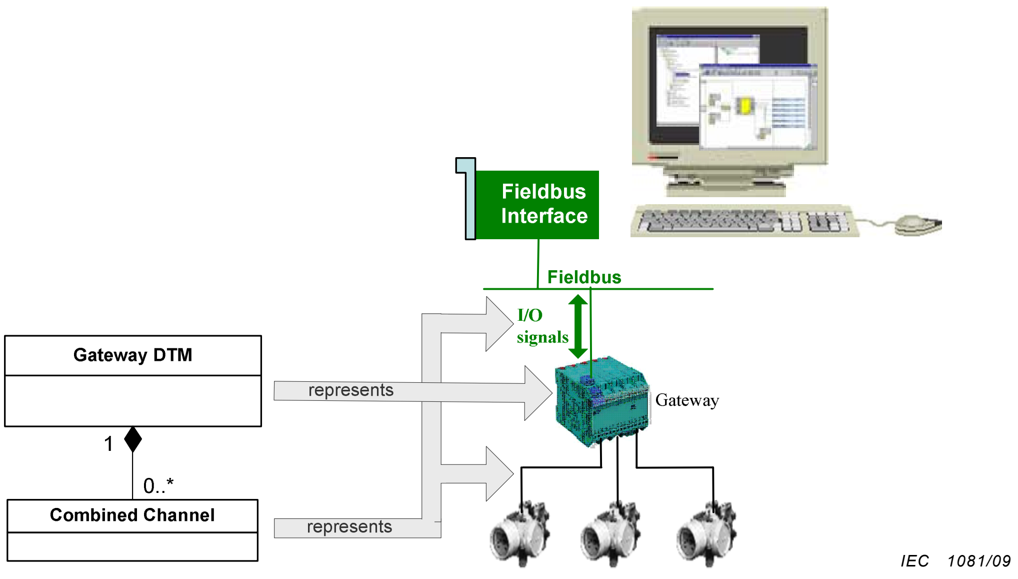


Figure 12 – Combined Process / Communication Channel

In other words, such a channel shall support the Process Channel services (see 7.5) for the fieldbus where the gateway is connected to and the Communication Channel Services (see 7.6) for accessing the devices connected to the linked fieldbus (or point-to-point connection).

4.3 Modularity

Different field bus protocols are using different device models. FDT supports following different approaches to describing the structure of device:

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

- DtmDeviceInstanceTopology,
- Module DTM, and
- BTM.

4.4 Bus categories

A DTM exposes information about encapsulated protocol specific definitions. The information returned by service GetTypeInformation (see 7.2.3.1) contains so called bus categories which are Universally Unique Identifiers for a fieldbus protocol (or a point-to-point communication).

The FDT concept distinguishes between supported and required bus categories:

- supported bus categories refer to the protocol specific communication services provided by a DTM (corresponding Communication Channels) to access a fieldbus;
- required bus categories refer to the protocol specific communication services required by a DTM to exchange data with its device.

4.5 System and FDT topology

The Frame Application is responsible for managing the system topology. That means the Frame Application shall organize the routing for accessing a device in the plan. A DTM shall not need to manage any information about the system topology.

System topology management is Frame Application specific. Some Frame Applications may require user interactions; others may support automatic operations such as topology importing or fieldbus scanning (see 6.2).

A DTM exposes all required information (see 7.2.3) which enables the Frame Application (and the user) to choose the appropriated DTM for a device, for example name, vendor, version of supported device types and corresponding identification properties.

The Frame Application initializes and links the DTMs according to the system topology. The linking of DTMs is called FDT topology. Figure 13 shows the FDT topology for a simple system topology with one fieldbus and two connected devices.

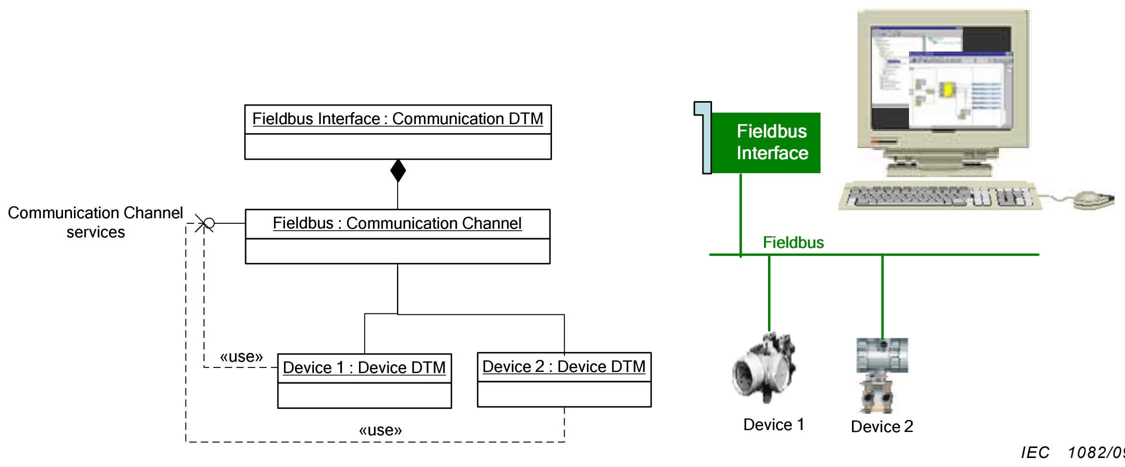


Figure 13 – FDT topology for a simple system topology

A Communication Channel is always used as the linking element between DTMs. As shown in Figure 13, the two Device DTMs are linked to the Communication Channel which provides access to the fieldbus.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The link between a Communication Channel and a DTM is created by the Frame Application. However, final decision whether a DTM shall be linked or not has to be made by the Communication Channel. The Frame Application has to call the service ValidateAddChild (see 7.6.2.1) before link is created. The Communication Channel shall at least check whether required bus categories of the DTM to be linked fits to its own supported bus categories. If this is not the case, then linking shall be rejected. In addition, the Communication Channel may perform additional checks, for example if the number of linked DTMs exceeds a limit. The sequence diagrams in 8.1 describe this sequence in more detail.

Neither the Communication Channel (or corresponding DTM) nor the linked DTM shall need to manage topology information. The Frame Application supports requesting topology information by service GetParentNodes (see 7.7.3.5) and service GetChildNodes (see 7.7.3.6).

The FDT topology management for a more complex system topology with gateways, modular devices and devices with blocks works exactly the same way.

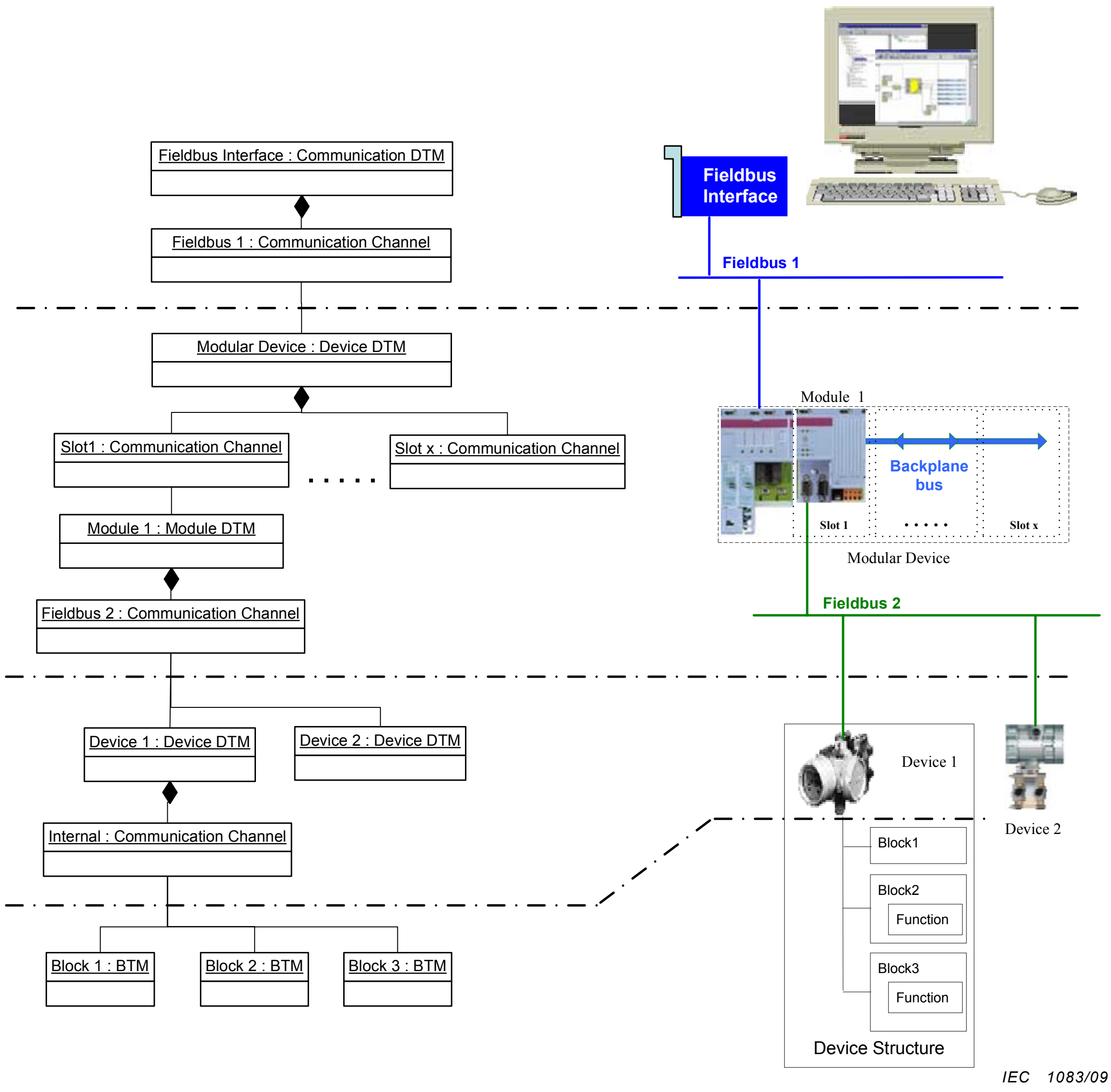


Figure 14 – FDT topology for a complex system topology

Starting from the root, all DTMs are linked via Communication Channels according to the physical system topology.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The root element DTM for a device which is represented by multiple DTMs (e.g. Modular Device and Device 1 DTMs in Figure 14) may support automatic creation of the FDT sub-topology that corresponds to the complete device structure. The Frame Application supports the service CreateChild (see 7.7.3.2), DeleteChild (see 7.7.3.3) and MoveChild (see 7.7.3.4) to enable a DTM to alter the FDT topology.

4.6 Peer to peer and nested communication

The Frame Application has to provide in each case a peer-to-peer connection between a DTM and a device.

It is under the control of the Frame Application to enable a DTM to communicate with its device. The Frame Application has to pass the Communication Channel to use the DTM by calling the service SetLinkedCommunicationChannel (see 7.2.4.2) and allow the Communication by calling the service EnabledCommunication with TRUE (see 7.2.4.3). In order to access the device, the DTM uses the Communication Channel services (see 7.6.1) as shown in Figure 15.

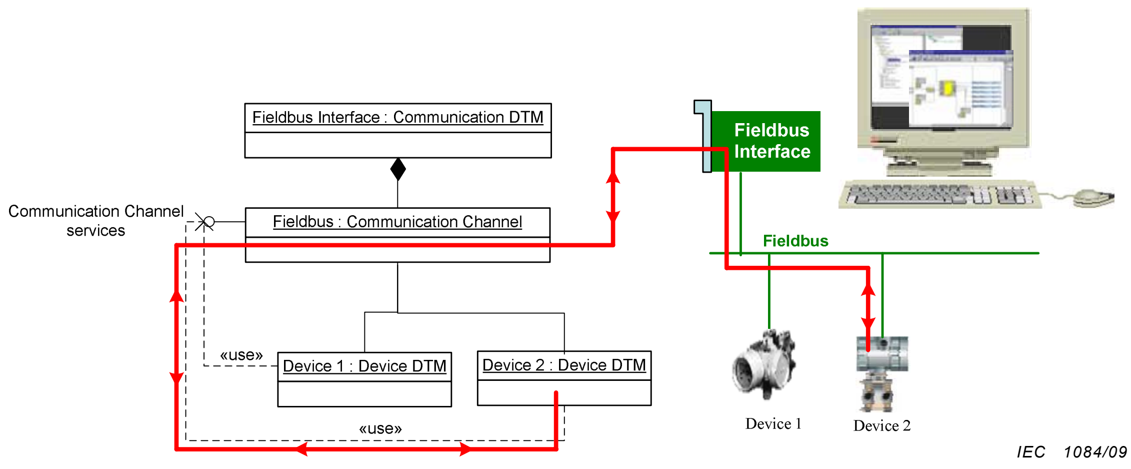


Figure 15 – Peer to peer communication

A DTM has to call the Communication Channel service Connect (see 7.6.1.2) to establish a communication link to the device. After the link is established, it is able to communicate with its device by calling the service Transaction (see 7.6.1.6). The diagram in 8.3.2 describes this sequence in more detail.

A DTM shall not know anything about the kind of the overlaying system. Thus the same DTM is able to communicate with devices attached to different fieldbus interfaces. Nested communication is used if the devices are connected to a sub-system, for example if a device which is connected to a channel of a Remote I/O (see Figure 16).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

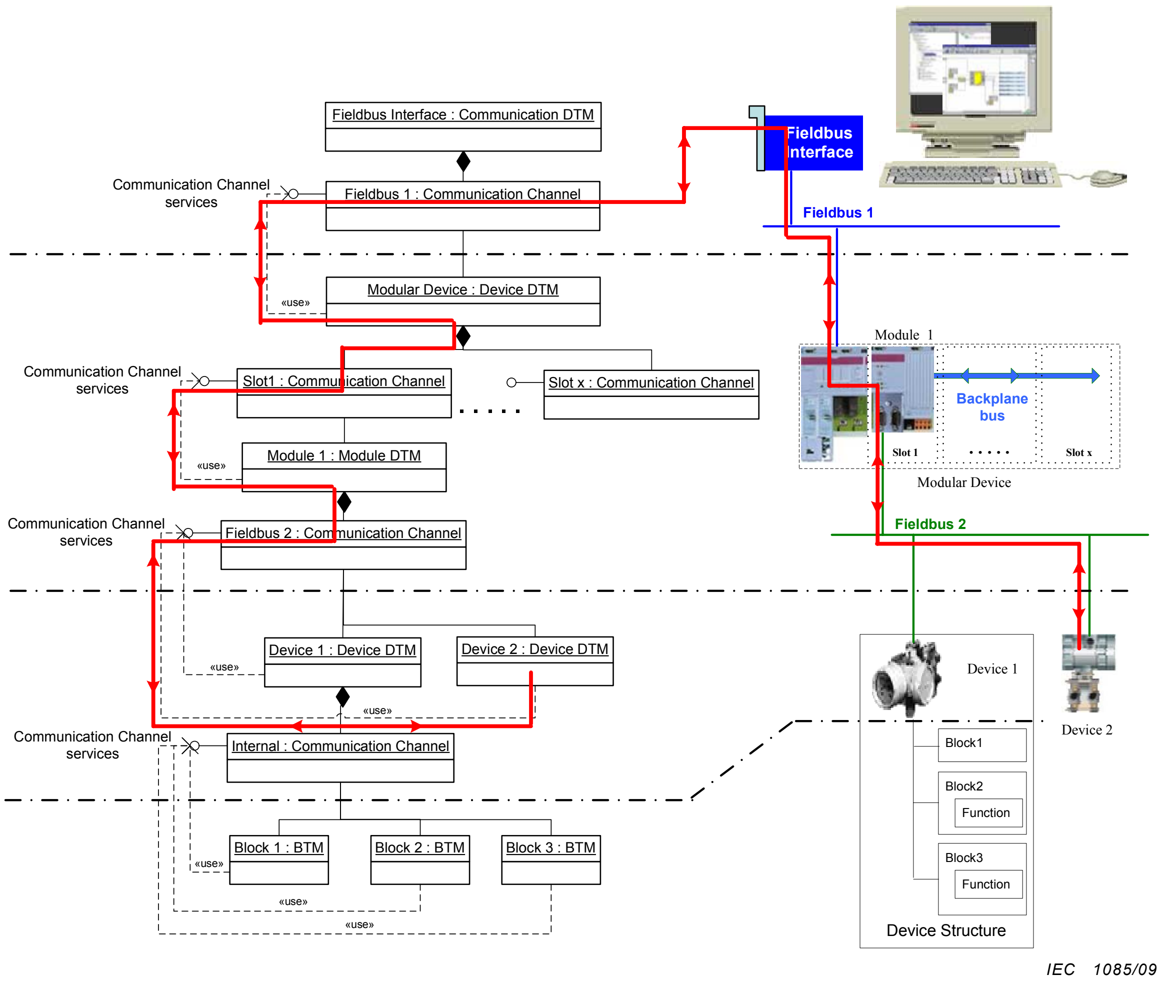

Fieldbus Interface : Communication DTM
Communication Channel services
Fieldbus 1 : Communication Channel
Modular Device : Device DTM
«use»
Communication Channel services
Slot1 : Communication Channel
Slot x : Communication Channel
Module 1 : Module DTM
Communication Channel services
Fieldbus 2 : Communication Channel
Device 1 : Device DTM
Device 2 : Device DTM
Communication Channel services
Internal : Communication Channel
Block 1 : BTM
Block 2 : BTM
Block 3 : BTM
Fieldbus Interface
Fieldbus 1
Module 1
Backplane bus
Slot 1
Slot x
Modular Device
Fieldbus 2
Device 1
Device 2
Block1
Block2
Function
Block3
Function
Device Structure
IEC 1085/09

Figure 16 – Nested communication

Starting from the root the Frame Application shall pass the linked Communication Channels to all DTMs which should be enabled for communication with its device. Like in the peer to peer communication, all DTMs simply use the Communication Channel services to communicate with their devices without the awareness of the nested communication (e.g. device 2 DTM in Figure 16). The diagram in 8.3.3 describes this sequence in more detail.

The communication in FDT follows the following concepts:

- each Communication Channel wraps the communication message from the linked DTM below, without knowing the contents;
- the linked DTM below does not have any knowledge about the system topology;
- the DTM shall only support its own communication protocols:
  – communication / routing through any system topology is possible without limitations.

4.7 DTM, DTM Device Type and Hardware Identification Information

4.7.1 DTM and DTM Device Type

A DTM supports two services to request information about itself and the supported device types (see Figure 17). Usually this information is used by the Frame Application to create libraries, select a DTM during creation of FDT topology, or for simple validations.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

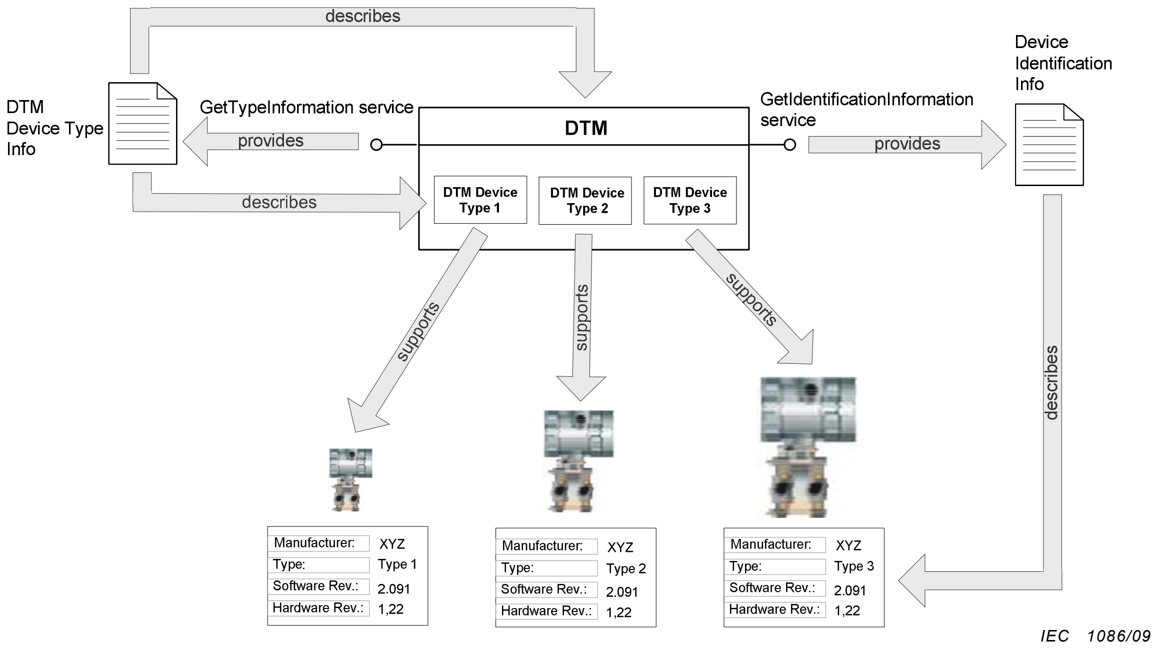


Figure 17 – DTM, DTM Device Type and Device Identification Information

A DTM is a software component containing device specific application software. The service GetTypeInformation (see 7.2.3.1) returns general information about this piece of software such as name, version, and vendor of the software, the FDT version (see also 4.12) as well as a list of supported device types.

The representation for a particular device type within the DTM is called DTM Device Type. A DTM may contain one or more DTM Device Types. The concrete design and implementation of the DTM Device Types is not within the scope of the FDT; but service GetTypeInformation also returns information about these pieces of software like name, version, vendor, supported protocols etc.

When a Frame Application adds a DTM to the FDT topology then it selects one of the supported DTM Device Types (see service Initialize, 7.2.4.1). The DTM shall persist the selection within its instance data, in order to restore it when a DTM representing same device is initiated next time. The DTM shall provide information about the selected DTM Device Type by the service GetActiveTypeInfo (see 7.2.3.4). The service shall always return the current DTM Device Type. If an update of the DTM occurred, the new DTM shall return the updated DTM Device Type information.

4.7.2 Supported hardware identification

Information about physical device types, which can be handled by the DTM Device Types is returned by the service GetIdentificationInformation (see 7.2.3.2). For example manufacturer, type, hardware and embedded software version of the device. The DTM may even return wildcards for some specific device identification elements to signal that the DTM Device Type can be used for all devices for which the wildcard matches (e.g. the character asterisk '*' for the hardware version may signal that the DTM Device Type supports all hardware versions). .

The information returned by service GetIdentificationInformation is fieldbus-specific and therefore defined by the IEC 62453-3xy document describing the protocol profile integration in FDT. However, FDT defines the means to transform the information into a protocol-independent format to enable Frame Applications without protocol-specific knowledge to use it. The transformation is implementation technology dependent and therefore defined in the IEC 62453-4z documents describing the mapping to specific implementation technologies.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

4.7.3 Connected Hardware Identification

Furthermore a DTM supports service HardwareInformation (see 7.2.3.3) that enables to read device information online from the connected device (see Figure 18).

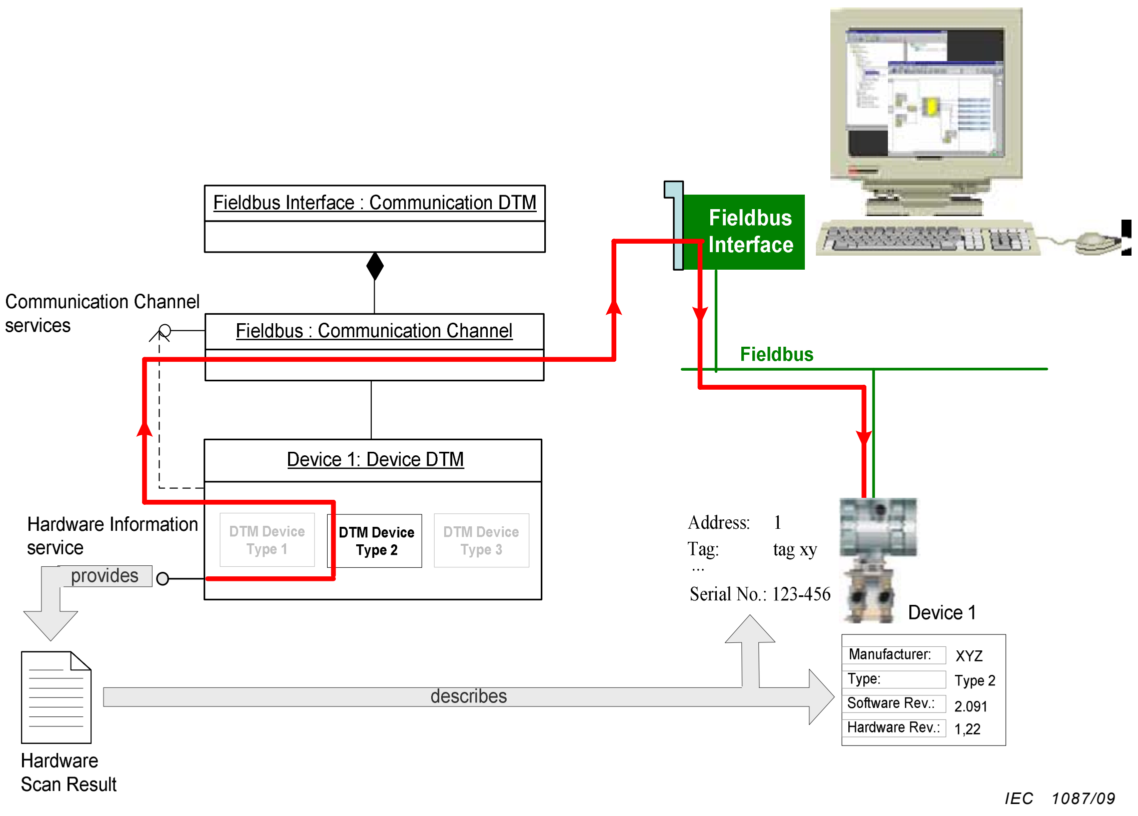


Figure 18 – Connected Hardware Identification

The service returns device type related information like manufacturer, type, hardware and embedded software version as well as device instance related information like fieldbus address, tag and serial number. This information is also fieldbus specific like the information returned by GetIdentificationInformation (see 7.2.3.2). Therefore, concrete format and transformation to protocol-independent format is also defined by the IEC 62453-3xy document describing the protocol profile integration in FDT. See examples in 6.2.4 and 6.2.5.

4.8 DTM data persistence and synchronization

Basically, three kinds of DTM related data can be seen:

- instance-related data (called “instance data”). Instance-related data belongs to the DTM itself. It is up to the DTM which data it stores but the DTM has to guarantee that it is able to represent the stored device instance by loading these data;
- non-instance-related data. Non-instance-related data belongs to a project or is global or DTM Device Type specific;
- bulk data. DTM specific data, for example historical data, temporary data.

The format of all kinds of data is DTM-specific, but a DTM shall expose an Universal Unique identifier specifying the format which is used to save its instance data. Furthermore, a DTM shall provide a list of compatible data formats it is able to load. These format identifiers are included in the DTM Device Type information returned by GetTypeInformation (see 7.2.3.1). A Frame Application may use the format identifiers to find correct DTM after a DTM software upgrade (see 8.9).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Two types of DTM data persistence mechanism are defined in FDT. A DTM shall either use the Frame Application project storage or a private DTM storage (see Figure 19).

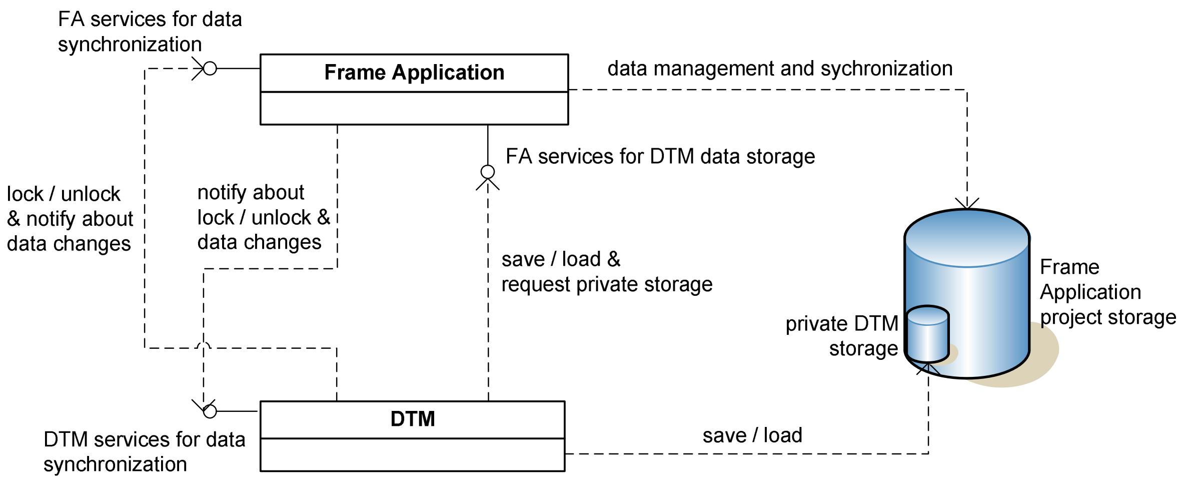

FA services for data synchronization
Frame Application
data management and sychronization
FA services for DTM data storage
lock / unlock & notify about data changes
notify about lock / unlock & data changes
save / load & request private storage
Frame Application project storage
private DTM storage
DTM
DTM services for data synchronization
save / load
IEC 1088/09

Figure 19 – FDT storage and synchronization mechanisms

The services LoadInstanceData (see 7.7.5.1) and service SaveInstanceData (see 7.7.5.2) enable the DTM to save and load instance-related data in the Frame Application project storage. The Frame Application has to guarantee the data consistency for multi-user and multi-client data access. The ‘FA services for DTM data synchronization’ (see 7.7.6) shall be used by the DTM to attempt locking of its instance-related data before alternation. The ‘DTM services related to data synchronization’ (see 7.2.13) shall be used by the Frame Application to notify a DTM about events, for example if data was locked by another DTM instance. See 8.5 Multi-user scenarios.

The services GetPrivateDtmStorageInformantion (see 7.7.5.3) returns information about private DTM storage which is directly accessible for the DTM without the need to use further services provided by the Frame Application. The private DTM storage enables the DTM to store instance-related, non-instance-related, and bulk data. The DTM itself has to guarantee the data consistence for multi-user and multi-client data access. It is in the responsibility of the Frame Application to create a backup strategy for the private DTM storages.

A DTM shall be able to work without any side effects if the private DTM storage is not available. It shall ensure that there is no impact to the instance data managed by servce SaveInstanceData (see 7.7.5.1) and LoadInstanceData (see 7.7.5.2).

The private DTM storage mechanism is implementation technology dependent and thus defined in the IEC 62453-4z documents defining mapping to specific technologies.

4.9 DTM device parameter access

A DTM supports services to read and write device parameters stored in the DTM instance data (see 7.2.8) and directly in the connected device (see 7.2.10).

It is up to a DTM and depends on the device- and fieldbus-type which device parameters are available. Semantic information (see SemanticId in Table A.4) is given to the device parameters. By using these, for example a Frame Application is able to get the information regarding the meaning and usage of a single parameter or data structure. The definition regarding the identifier is protocol-specific and thus defined in the IEC 62453-3xy specifications defining the protocol profile integration in FDT.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

4.10 DTM state machine

4.10.1 DTM states

The following diagram (Figure 20) defines the different states of a DTM.

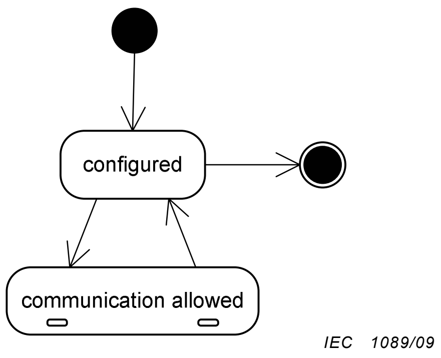


Figure 20 – DTM state machine

The state machine consists of two states: ‘configured’ and ‘communication allowed’. Which DTM services are available in a certain state is defined in the service descriptions (see Clause 7).

All state transitions are triggered by the Frame Application by calling corresponding DTM services. The state transition succeeds, if the service returns fdt:serviceSucceeded = TRUE. The DTM remains in its previous state, if state transition is rejected by returning fdt:serviceSucceeded = FALSE. The circumstances under which the DTM is allowed to reject the state transitions are defined in the service descriptions.

State Transition: Initial state – state ‘configured’

The service Initialize (see 7.2.4.1) shall be called to bring the DTM from its initial state into the state ‘configured’.

State Transition: State ‘configured’ – state ‘communication allowed’

The service EnableCommunication (see 7.2.4.3) shall be called with TRUE to bring the DTM from state ‘configured’ into the state ‘communication allowed’. A precondition for this call is that the Frame Application has informed the DTM about the communication infrastructure by calling the service SetLinkedCommunicationChannel (see 7.2.4.2).

Only in this state the DTM is allowed to establish a communication link to its device by calling the service Connect (see 7.6.1.2) of its linked Communication Channel. The sub-states within this state are described in the communication state machine (see 4.10.2).

State Transition: State ‘communication allowed’ – state ‘configured’

The service EnableCommunication (see 7.2.4.3) shall be called with FALSE to bring the DTM from state ‘communication allowed’ into the state ‘configured’.

State Transition: State ‘configured’ – final state

The service Terminate (see 7.2.4.6) shall be called to bring the DTM from state ‘configured’ into its final state.

Table 3 shows an overview of these transitions

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 3 – Transitions of DTM states

Start state	End state	Trigger	Condition
Initial state	Configured	Initialize	
Configured	Communication allowed	EnableCommunication(TRUE)	Communication infrastructure available to DTM
Communication allowed	Configured	EnableCommunication(FALSE)	
Configured	Final state	Terminate	

4.10.2 ‘Communication allowed’ sub-states

This state machine is describing the sub-states in the DTM state ‘communicationAllowed’ (see Figure 21).

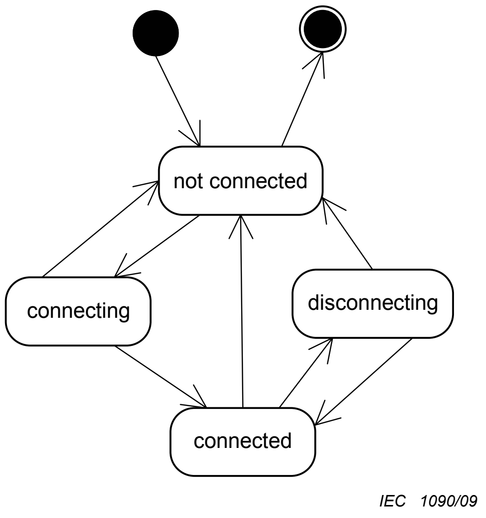


Figure 21 – Substates of communication allowed

Table 4 provides a description of the transitions of the DTM ‘communication allowed’ sub-states.

Table 4 – Transitions of DTM ‘communication allowed’ sub states

Start state	End state	Trigger	Condition	Action
not connected	connecting	Trigger of Online service of the DTM		Connect Request
connecting	not connected		Connection could not be established	
connecting	connected	Connect Response		
connected	not connected	Connection is terminated by communication		
connected	disconnecting	Online Service of DTM is finished		Disconnect Service
disconnecting	connected		Connection could not be terminated	
disconnecting	disconnected	DisConnect Response		

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

4.11 Basic operation phases

4.11.1 Roles and access rights

When a DTM is started by the Frame Application, the user role is passed to the DTM, which shall restrict the parameter access according to the role.

Also the availability of functions and appearance of user interfaces may be adapted.

Examples:

- an observer will not get access to calibration functions;
- an observer shall not modify parameters.

See 5.2.

4.11.2 Operation phases

If a Frame Application requests available functions from the DTM, it passes the operation phase, which shall be used by a DTM to adapt the availability of functions and the appearance of the user interface.

There are five operation phases:

- Engineering

  Planning and configuring of a plant,

  No online access to the plant;
- Commissioning, workshop

  Installing the plant infrastructure and the devices,

  Scanning the network to verify a planned network,

  Downloading project data into the plant,

  Programming, diagnosis of the devices,

  Adjusting parameters;
- Runtime

  The plant is completely commissioned and running,

  Very restricted access to configuration and parameterization data,

  Scan to verify the network,

  Replacing defect devices,

  Reading process values and diagnosis information;
- Service

  This operation phase is used for service tool Frame Applications. Scan to create a topology. Scan to verify a network. All service related operations. Unrestricted access to the device;
- Not supported

  The application does not support the operation phases defined above.

  A DTM shall offer all functions if the Frame Application passes “not supported”.

The following table (see Table 5) describes the usage of operation phases in different Frame Application types.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 5 – Operation phases

System Frame Application	Service tool Frame Application	Other Frame Applications
Engineering	Service	notSupported
Commissioning		
Runtime		

For example the DTM allows the maintenance actor during commissioning phase the complete Online parameterization, but during runtime phase it does not allow it or it allows it only for some of its parameters

4.12 FDT version interoperability

4.12.1 Version interoperability overview

FDT is a component based concept, which is constantly enhanced to new improved versions. Control system environments typically run for 10-15 years in contrast. If hardware and software components in a control system have to be exchanged, a mixture of components designed for different FDT versions may emerge.

This raises the question, how components based on different FDT versions can cooperate.

This clause mainly deals with two topics concerning FDT version interoperability:

a) Persistence

   New versions of FDT components (Frame Applications and DTMs) shall be able to load instance data of older versions.

b) Component Interoperability

   Components of different FDT versions shall interoperate properly. DTMs of newer FDT versions shall run in older Frame Applications and vice versa. Communication shall work even if FDT versions of Device DTMs and Communication DTMs differ. Interaction between these components is technology dependent and therefore details defined within IEC 62453-4z documents.

A limiting condition for future FDT-enhancements is to ensure maximum compatibility of different FDT versions. This results in a maximum interoperability of FDT components originally designed for different FDT versions.

Within IEC 62453-4z documents, FDT takes care about version interoperability on two different levels.

c) Specification Level

   The FDT specification targets full compatibility of different specification. Properly designed FDT components will achieve compatibility without extra implementations.

   Table 6 Implementation Level

   Within IEC 62453-4z documents the FDT version is composed by a major, a minor, a release and a build number (e.g. 1.2.0.3 where 1 is the major, 2 is the minor number, 0 the release and 3 the build number). Compatibility is defined slightly different with respect to the major number:

   Table 6 compatibility between FDT components with the same FDT major version number is assured by the IEC 62453-4z specification;

   Table 6 compatibility between FDT components with different FDT minor, release and build version numbers can be achieved by extra code paths. FDT will provide needed information and mechanisms to support full compatibility at this level. The minor or release number is increased if new functionality, dependent of its

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

appropriate importance, is added to the FDT specification. The build number is increased if a bugfix within the specification or the type library is necessary.

4.12.2 DTM and device versions

When providing a new DTM for a new major FDT version, it is highly recommended to use new registration information (see 7.2.2). If registration information of a DTM has changed, FDT IEC 62453-4z will provide a mechanism to upgrade to the newer.

Within its device catalog, a new DTM version is able to provide the same list or a subset of the devices of the old DTM version. In this case, a FDT Frame Application will handle the two versions of a DTM as any other DTMs able to handle the same field device as two distinguished DTMs.

A DTM of a higher FDT version which does not use new registration information shall support at least all DTM Device Types of that were supported by previous versions of this DTM.

4.12.3 Persistence

If the version of the Frame Application changes or if the version of the DTM changes persistence (loading of stored projects) may be a problem.

A Frame Application shall be able to load old instance data and to provide the unchanged instance data to the DTM even if the DTM version has changed. A DTM shall be able to load old instance data as long as the registration information of the DTM do not change. If registration information of a DTM has been changed, FDT will provide a technology dependent mechanism to upgrade to the newer DTM defined within IEC 62453-4z documents. By this way, it is possible to load an existing old instance data of a DTM into another DTM object. The new DTM is responsible to convert the instance data accordingly.

A new DTM shall provide information what DTM instance data it can load (compatibility of dataset). It shall support all DTM Device Types of the old DTM. If the Frame Application recognizes the new DTM, it may inform the user (e.g. “A new compatible DTM was installed, do you want to use the new version instead of the old version?”). If the user confirms the new DTM object is instantiated instead of the old and it loads and converts the old instance data.

4.12.4 Nested communication

4.12.4.1 Nested communication overview

Since DTMs may be chained with respect to the FDT topology, they need to communicate properly. Because the involved DTMs may support different FDT versions, the following restrictions shall be considered.

4.12.4.2 Information exchange

Information exchanged via communication services may contain version dependent parameters.

Therefore, a DTM shall only use communication service parameters according to the FDT version of its parent component. This version information of the parent component shall be provided within the service Connect (see 7.6.1.2) response information. A DTM shall then use only communication parameters compatible to this FDT version.

4.12.4.3 Communication Channel upgrade

A Communication Channel shall support the older FDT minor versions if it is upgraded to a higher minor version. This is due to the fact that some of its already existing children may not support the higher minor version interfaces.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

4.12.4.4 Version compatibility

Within nested communication the version number returned by a Gateway DTM depends on the functionality of its parent component.

If the Gateway DTM is at a higher version number than the parent component then the Gateway DTM:

- shall return the same version as the parent and provide the functionality according to this version or
- shall emulate the higher functionality if possible. Then the Gateway DTM shall guarantee that all required functionality is provided by its parent.

5 FDT session model and use cases

5.1 Session model overview

This clause describes the use cases that were considered when designing the FDT Specification. A Frame Application may support only a subset of these use cases. Also a Frame Application may support additional use cases that are not defined in this clause.

Figure 22 shows the main use cases.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

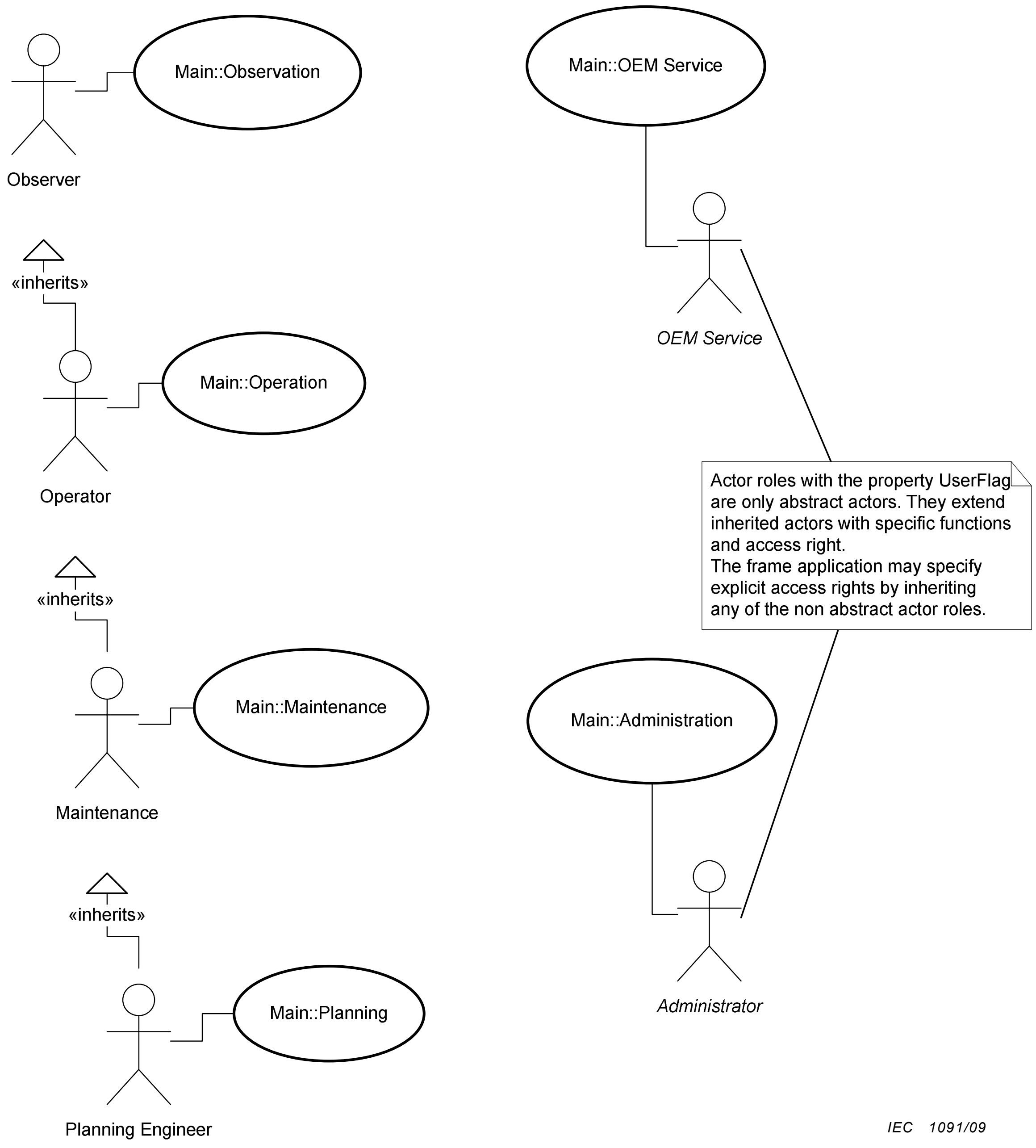

Main::Observation
Observer
«inherits»
Main::Operation
Operator
«inherits»
Main::Maintenance
Maintenance
«inherits»
Main::Planning
Planning Engineer
Main::OEM Service
OEM Service
Actor roles with the property UserFlag are only abstract actors. They extend inherited actors with specific functions and access right.
The frame application may specify explicit access rights by inheriting any of the non abstract actor roles.
Main::Administration
Administrator
IEC 1091/09

Figure 22 – Main Use Case Diagram

They are specified in more detail in the following subclauses, including textual descriptions and optionally some necessary scenarios.

5.2 Actors

The actor types in the above displayed use case diagram are structured in a hierarchical way.

The “Maintenance” actor inherits the “Operator” actor. The “Planning Engineer” actor inherits the use cases of the “Maintenance” actor. Both the “Administrator” and the “OEM Service” can extend the inherited actors by additional use cases. Use cases, which are passed on to an actor of higher level, may act in an extended way to match their intention.

The following table (Table 6) gives a brief description of the actors roles:

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 6 – Actors

Actor Name:	**Observer**
Brief Description:	This actor stands for a person that may only observe the current process
Inherits:	None
User Level:	FDTUserInformation.userLevel = observer
Access Verification:	The access as “Observer” actor may not have a password
Use Cases:	None
Actor Name:	**Operator**
Brief Description:	This actor stands for a person who has to observe and manage the current process. The “Operator” may therefore check the current status of the device, modify set values and check if the device is functioning well. The use cases for this actor role should enable the user to perform a complete diagnosis, watch the actual status and parameter set as well as the current process variables
Inherits:	Observer
User Level:	FDTUserInformation.userLevel = operator
Access Verification:	The access as “Operator” requires a password
Use Cases:	User Login, Online View, Audit Trail, Archive, Report Generation, Asset Management
Actor Name:	**Maintenance**
Brief Description:	A “Maintenance” person should have the possibility to perform all necessary maintenance operations including device exchange, teaching, calibration, adjustment, … The person may therefore download verified parameter sets, modify a subset of parameters online or offline, perform device-specific online operations and at the end of the processing, may have the possibility to upload the complete parameter set
Inherits:	Operator
User Level:	FDTUserInformation.userLevel = maintenance
Access Verification:	The access as “Maintenance” actor requires a password
Use Cases:	Simulation, Online Operation, Repair, Offline Operation, Bulk Data Handling User Login*, Online View*, Audit Trail*, Archive*, Report Generation*, Asset Management* (* Inherited use cases)
Actor Name:	**Planning Engineer**
Brief Description:	The actor “Planning Engineer” stands for person like a plant engineer, a specialist (s. VDI/VDE2187) or any fully authorized person. This Person has access to the complete set of functions of the Frame Application and can use DTM functions without any restrictions. Only OEM-specific operations are locked
Inherits:	Maintenance
User Level:	FDTUserInformation.userLevel = planningEngineer
Access Verification:	The access as “Planning Engineer” actor requires a password
Use Cases:	DTM Instance Handling, Simulation*, Online Operation*, Repair*, Offline Operation*, Bulk Data Handling*, User Login*, Online View*, Audit Trail*, Archive*, Report Generation*, Asset Management* (* Inherited use cases)

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Actor Name:	Administrator (abstract actor)
Brief Description:	The "Administrator" actor stands for person that has to perform administrative operations within an engineering environment He is responsible for adding and removing of software components
Inherits:	Depends on the Frame Application
User Flag*:	FDTUserInformation.administrator = TRUE
Access Verification:	The access as "Administrator" actor requires a password
Use Cases:	Integration and all inherited use cases
Actor Name:	**OEM Service (abstract actor)**
Brief Description:	The actor "OEM Service" may perform the complete set of DTM specific functions. OEM specific operation may be accessible to an "OEM Service" actor, like resetting of internal counters and exchanging firmware. The "OEM Service" has only inherited use cases, but the inherited use cases (especially the "Online Operation" use case) may have significant extensions
Inherits:	Depends on the Frame Application
User Flag*:	FDTUserInformation.oemService = TRUE
Access Verification:	The access verification for an "OEM Service" shall have two security levels: 1. Frame Application password: enables access to the Frame Application functionality 2. Device-specific password: enables access to OEM operation with the device. The verification of the device specific password lies in the responsibility of the DTM or even the device itself
Use Cases:	Inherited use cases only
* "User Flags" may be combined with any "User Level", to specify the inherited actor role of an "Administrator" or "OEM Service" actor.	

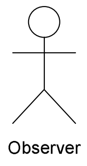
LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

5.3 Use cases

5.3.1 Use case overview

The following subclause describes all use cases of the use case diagram in tabular form. To reduce information, all attributes of included use cases, which are identical with the related main use case, are not displayed.

5.3.2 Observation



IEC 1092/09

Figure 23 – Observation Use Cases

Only the observation use cases can be executed by the actor "Observer" (see Figure 23). The "Observer" stands for a person that has the lowest access level. No use case is associated with this person.

5.3.3 Operation

The Operation use cases are available for the actor "Operator" (see Figure 24).

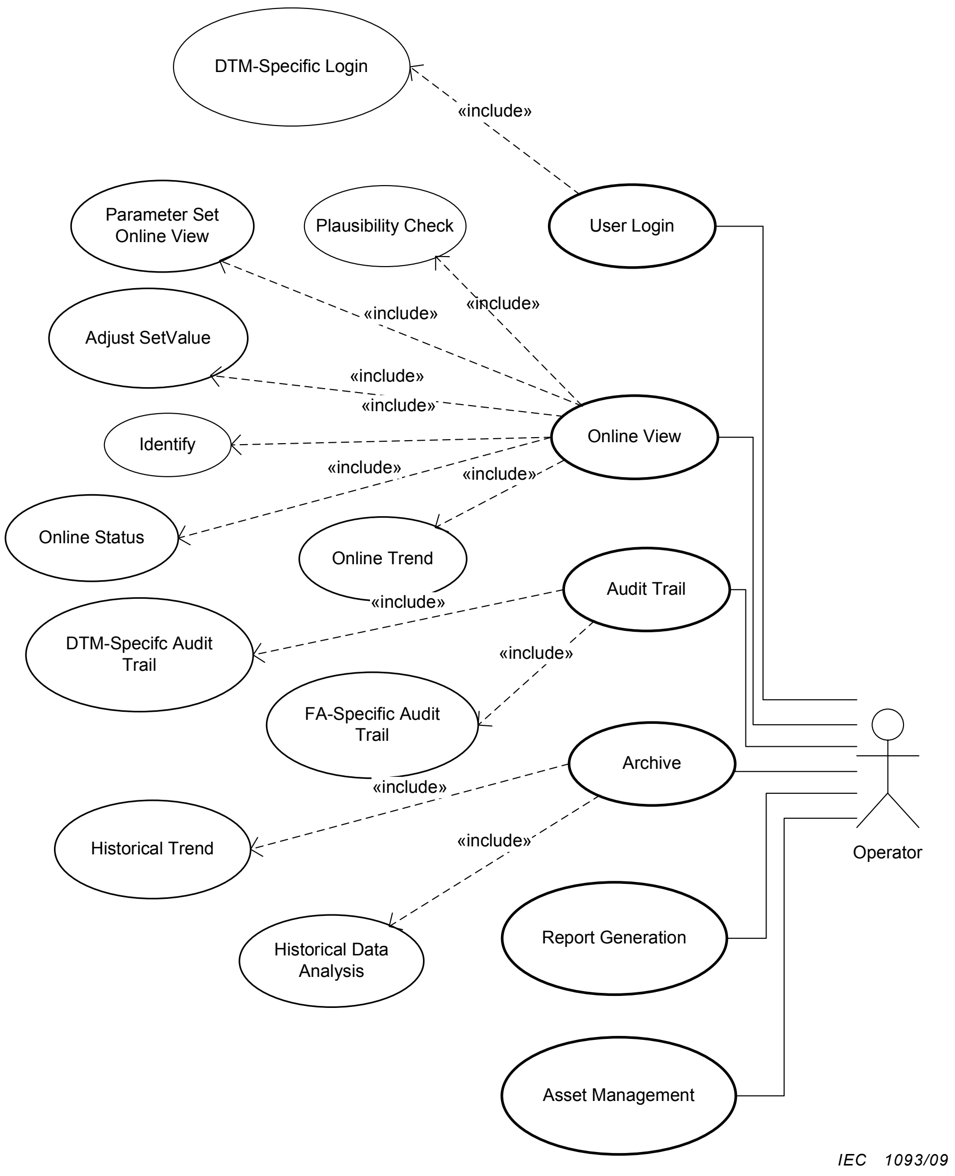


Figure 24 – Operation Use Cases

The following table (Table 7) provides a description of Operation use cases.

Table 7 – Operation Use Cases

Use Case:		User Login
Brief Description:		A person of specified actor type logs into the Frame Application. If verification fails, the person may act as an "Observer" actor only
GUI:		Part of the Frame Application
Print:		No support
Device Connection:		Not established
UserLevel	Observer:	Depends on the Frame Application
	Operator:	Accessible
	Maintenance:	
	Planning Eng.:	
UserFlag	Administrator:	
	OEM Service:	Accessible with extended functionality (see below)

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Included Use Case:		**DTM-Specific Login**
Brief Description:		Access to manufacturer specific information, e.g. production codes, changes log
GUI:		Part of the DTM
Print:		No support
Device Connection:		Typically needs an established connection
UserFlag	OEM Service:	Accessible only for OEM Service
Use Case:		**Online View**
Brief Description:		This use case offers a complete set of operations for viewing of device parameters and status
GUI:		The GUI is part of the DTM. The Frame Application can offer online views for a group of devices
Print:		Online View should always support print function to create documentation
Device Connection:		Needs an established connection to one or more devices (Adjust SetValue may influence device data)
UserLevel	Observer:	No access
	Operator:	Accessible
	Maintenance:	
	Planning Eng.:	
UserFlags:		No changes
Included Use Case:		**Parameter Set Online View**
Brief Description:		Enables the actor to load parameters from the device for viewing and documenting (printing). (DTM applicationId: fdtObserve)
Included Use Case:		**Adjust SetValue**
Brief Description:		Enables the actor to adjust the SetValues of a device (e.g. positioner, controllers). (DTM applicationId: fdtAdjustSetValue)
Included Use Case:		**Online Status**
Brief Description:		Use case for viewing and analyzing the current device status. (DTM applicationId: fdtDiagnosis)
Included Use Case:		**Online Trend**
Brief Description:		Use case for generating and watching trends of dynamic online values
Included Use Case:		**Plausibility Check**
Brief Description:		Every time the device parameters or the configuration data is changed, a plausibility check is automatically performed. As long as the plausibility check fails, the data cannot be written to the device
Included Use Case:		**Identify**
Brief Description:		Displays identification of device instance information, e.g. address, TAG, Fieldbus-Rev., DD-Rev., Firmware. This information is protocol dependent. It also may contain device type specific information. Further information may be provided: references to documentation, area to add comments to the device instance. Refer to Device Identification chapter in protocol specific FDT-JIG-Annex specifications. (DTM applicationId: fdtIdentify)

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Use Case:		Audit Trail
Brief Description:		Use case to enable the actor viewing and deleting audit trail data
GUI:		The component, which supports audit- trail, has to provide an adequate GUI
Print:		Printing for documentation purpose as a need for audit trails and therefore has to be supported
Device Connection:		No connection necessary
UserLevel	Observer:	No access
	Operator:	Accessible for viewing only
	Maintenance:	
	Planning Eng.:	View, export and delete
UserFlags:		No changes
Included Use Case:		**DTM-Specific Audit Trail**
Brief Description:		Every DTM might support an audit trail on its own. (DTM applicationId: fdtAuditTrail)
Included Use Case:		**FA-Specific Audit Trail**
Brief Description:		The Frame Application may implement a system global audit trail using internal data and named DTM properties exported from the DTM via the audit trail services
Use Case:		**Archive**
Brief Description:		The "Archive" use case describes the access to a Frame Application archive for viewing and data analysis
GUI:		The component, which supports archive functions, has to provide an adequate GUI
Print:		Every archive component should support printing too
Device Connection:		Not necessary
UserLevel	Observer:	No access
	Operator:	Accessible for viewing only
	Maintenance:	
	Planning Eng.:	View, export and delete
UserFlags:		No changes
Included Use Case:		**Historical Trend**
Brief Description:		The archive operations may support visualization of historical data (for example trending)
Included Use Case:		**Historical Data Analysis**
Brief Description:		Some Frame Applications and DTMs may support extended operations to analyze historical data
Use Case:		**Report Generation**
Brief Description:		The print function of a use case can normally be started from its GUI. In addition to printing the actual view of a GUI some Frame Applications may implement a configurable documentation mechanism which enables the actor to generate reports. This report may be generated by combining the prints of several use cases or several devices. For example a report could be generated containing "Online View", "Audit Trail" and "Online Operation" printouts for one device or a report may be generated containing the configuration of a HART multiplexer and its clients
GUI:		Frame Application
Print:		Frame Application in combination with one or more DTMs
Device Connection:		Depends on the type of report
UserLevel, UserFlag:		The printed information may depend on user level and user flags

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Use Case:		Asset Management
Brief Description:		Frame Application provides mechanism for asset management
GUI:		The component, which supports asset management functions, has to provide an adequate GUI. (DTM applicationId: fdtAssetManagement)
Print:		Frame Application supporting this use case should support printing, too
Device Connection:		Not necessary
UserLevel	Observer:	No access
	Operator:	Accessible for viewing only
	Maintenance:	
	Planning Eng.:	View, export and delete
UserFlags:		No changes

5.3.4 Maintenance

The maintenance use cases are available for the actor "Maintenance Engineer" (see Figure 25).

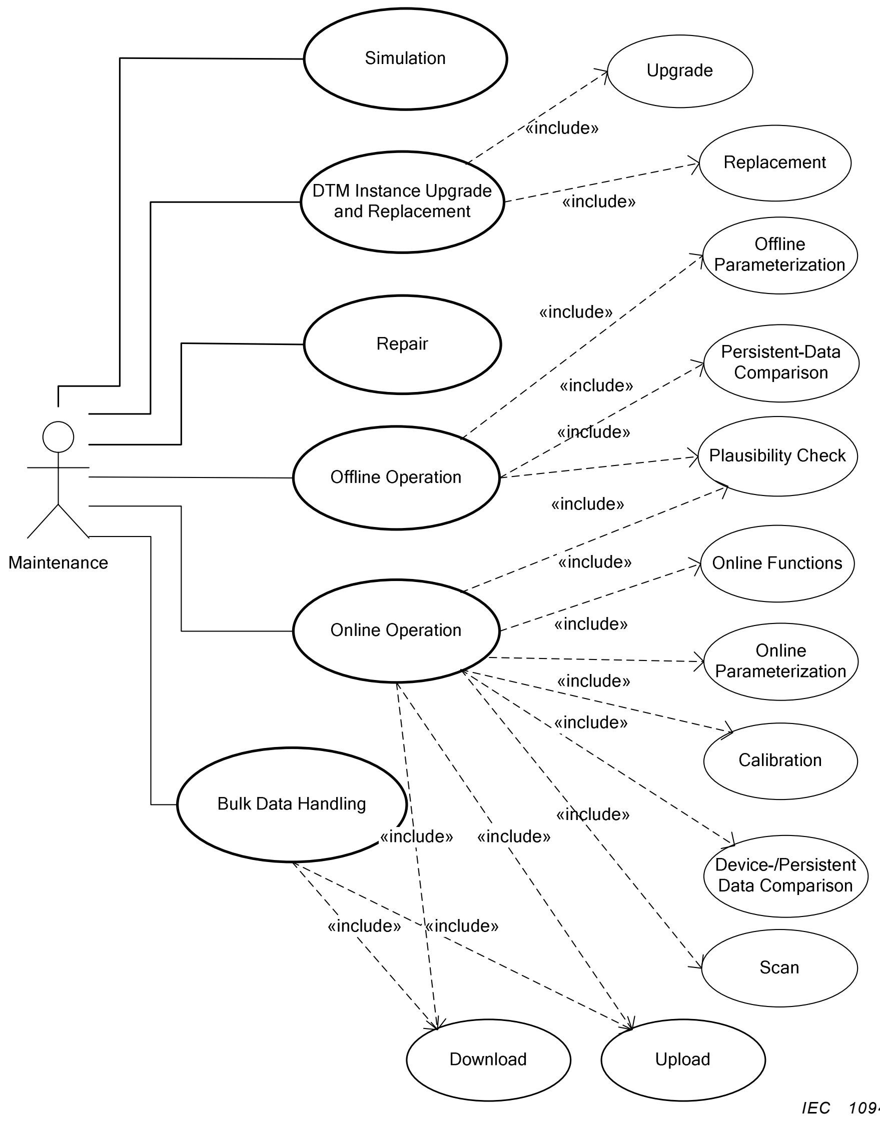


Figure 25 – Maintenance use cases

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The Maintenance use cases are described in the following table (Table 8).

Table 8 – Maintenance use cases

Use case:		**Simulation**
Brief description:		This use case simulates the behavior of a device. Therefore the actor is allowed to perform temporary changes within the device parameters, like forcing the device to set a fixed current analog output
GUI:		DTM-specific
Print:		DTM-specific
Device connection:		Shall have a connection established
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	Accessible
	Planning Eng.:	
UserFlags:		No changes
Use case:		**Offline operation**
Brief description:		This use case includes all operations which do not require an established connection to a device. Only for up- and download a connection to a device may be temporarily established
GUI:		DTM and Frame Application
Print:		DTM
Device Connection:		Depends on the included use case
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	Accessible
	Planning Eng.:	
UserFlags:		No changes
Included use case:		**Plausibility check**
Brief description:		Every time the parameter values are changed a plausibility check is automatically performed. As long as the plausibility check fails, the data cannot be saved to the database or written to the device. There may be two options depending on rules or application environment: After display of a warning, inconsistent data may be stored For safety reasons after display of a warning writing of data is prohibited
Users:		The plausibility check may be more tolerant for actors of higher level.
Included use case:		**Offline parameterization**
Brief description:		The user may change parameter values. The changes will only affect data in the Frame Application database after stored as persistent data. Data should be signed as transient data until they are stored
GUI:		DTM (applicationId: fdtOfflineParameterize)
Print:		DTM
Device connection:		No connection necessary

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Included use case:		Persistent data comparison
Brief description:		Allows the user to compare the offline parameter set with parameter sets of other instances, of device default or of project default parameter sets. The data sets may be editable and data may be transferred from one data set to the other
GUI:		DTM (applicationID: fdtOfflineCompare)
Print:		May be supported by the DTM
Device connection:		Not necessary
Use case:		**Repair**
Brief description:		This use case stands for operations which shall be performed to repair or change a device. Example: A Frame Application supports a temporary deletion of a device with automatically parameterization download when the device has been reinstalled
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	Accessible
	Planning Eng.:	
UserFlag	Administrator:	No changes
	OEM service:	Accessible with extended functionality (see below)
Use case:		**DTM instance upgrade and replacement**
Brief description:		This use case stands for operations which shall be performed to upgrade or to replace a DTM. Upgrade or replacement are used when new DTM is different to the previous DTM. The data format of the new DTM can be either compatible (upgrade) or incompatible (replacement) to the data format of the previous DTM
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	Accessible
	Planning Eng.:	
UserFlag	Administrator:	No changes
	OEM Service:	No changes
Included use case:		**Replacement**
Brief Description:		This use case stands for operations which shall be performed to replace a DTM in the case of incompatible data set format.
Included use case:		**Upgrade**
Brief Description:		This use case stands for operations which shall be performed to upgrade a DTM in the case of compatible data set format.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Use case:		**Online operation**
Brief description:		This use case includes all operations which are performed directly with the device.
GUI:		DTM-specific
Print:		DTM-specific
Device connection:		Connected or disconnected
Users:	Operator:	No access
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	Accessible
	Planning Eng.:	Fully accessible excluding OEM-specific parameter
UserFlag	Administrator:	No changes
	OEM service:	Accessible with extended functionality (see below)
Included use Case:		**Online functions**
Brief description:		The "online functions " use case offers all procedures which include direct communication with the device. The use case may include simple procedures like resetting but also more complex procedures with several sections like calibration or teach in
Included Use Case:		**Online Parameterization**
Brief Description:		When this use case starts all parameters are read from the device and exposed to the user within a GUI. Within the GUI, the user may change parameters, depending on the user level. The device is updated immediately if the changed parameterization is in a verified state. (DTM applicationId: fdtOnlineParameterize)
Included Use Case:		**Device-/Persistent Data Comparison**
Brief Description:		This use case gives the user the possibility to compare persistent and device data. The user may edit data and copy between these two sources. (DTM applicationId: fdtOnlineCompare)
Included Use Case:		**Calibration**
Brief Description:		This use case invokes calibration procedures to adjust the input for e.g. pressure transmitters or the current for the output. (DTM applicationId: fdtCalibration)
Included Use Case:		**Plausibility Check**
Brief Description:		Every time the device parameters or the configuration data is changed a plausibility check is automatically performed. As long as the plausibility check fails, the data cannot be written to the device.
UserLevel, UserFlag:		The plausibility check may be more tolerant for actors of higher level
Included Use Case:		**Upload**
Brief Description:		Reads the whole parameter set from the device into the Frame Application database. An upload started by an "OEM Service" actor may upload additional OEM parameters.
GUI:		If the Frame Application supports up- and download for a group of devices, the Frame Application may offer a GUI for a better control of the upload process.
Print:		No support
Device Connection:		Needs a temporary connection

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Included Use Case:			Download
Brief Description:			Like upload, but in the opposite direction.
GUI:			If the Frame Application supports up- and download for a group of devices, the Frame Application may offer a GUI for a better control of the download process.
Print:			No support
Device Connection:			Needs a temporary connection
Included Use Case:			**Scan**
Brief Description:			Retrieves the list of available devices from a channel object. The channel object may be provided by a DTM or by Frame Application.
GUI:			The Frame Application may offer a GUI for representation of the list of devices.
Print:			No support
Device Connection:			Needs a temporary connection. (The channel shall have online access.)
Use Case:			**Bulk Data Handling**
Brief Description:			This use case stands for the possibility to handle (e.g. up- and download, parameter adjustment or report generation) a group of instruments at once. This use case handles bulk data on system level.
GUI:			Frame Application
Print:			Not supported
Device Connection:			Needs a connection
UserLevel	Observer:		No access
	Operator:		
	Maintenance:		Accessible
	Planning Eng.:		
UserFlags:			No changes
Included Use Case:			**Upload**
Brief Description:			Reads the whole parameterization from the devices of the group into the Frame Application database
Included Use Case:			**Download**
Brief Description:			Like upload, but in the opposite direction

5.3.5 Planning

The planning use cases are available for the actor “Planning Engineer” (see Figure 26).

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

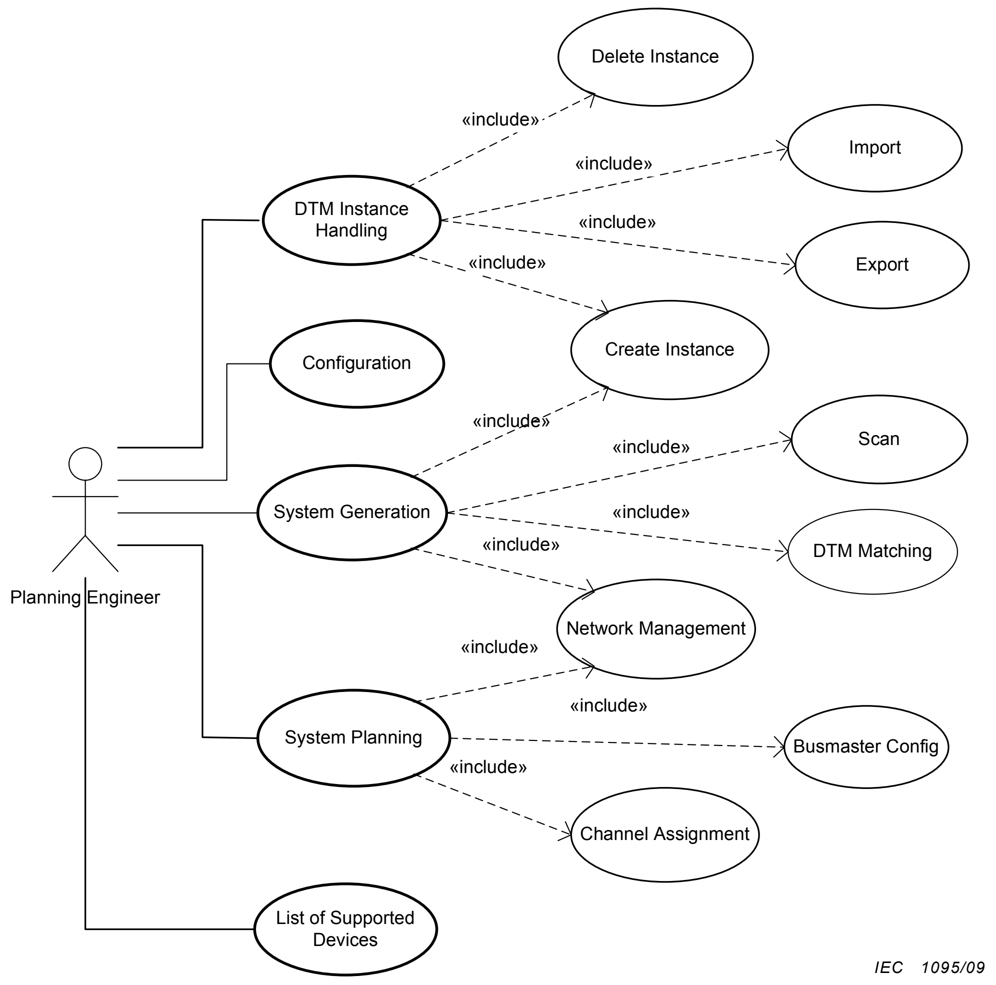


Figure 26 – Planning use cases

Table 9 provides a description of the planning use cases.

Table 9 – Planning use cases

Use case:		**DTM instance handling**
Brief Description:		With this use case a "Planning Engineer" actor can handle (e.g. instantiate, import, export, delete) a device instance in the Frame Application.
GUI:		Frame Application and DTM
Print:		Not supported
Device Connection:		Not necessary
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	
	Planning Eng.:	Accessible
UserFlag		No changes
Included use case:		**Create instance**
Brief description:		In this use case a new device instance can be created and placed into the project. After creation of a new device instance, the device parameter set shall be instantiated. This action is performed by the DTM.
GUI:		Frame Application and DTM (if it supports device default selection)

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Included use case:		**Import**
Brief description:		The parameter set may be instantiated by importing data from a database (for example a template database) or by copying the parameter set from an already instantiated and verified device.
GUI:		FA
Included use case:		**Export**
Brief description:		To copy data from one device instance to another, the device instance data can be exported.
GUI:		FA
Included use case:		**Delete instance**
Brief description:		Deletion of a device instance
GUI:		FA
Use case:		**Configuration**
Brief description:		This use case enables the “Planning Engineer” actor to configure a complex device.
GUI:		DTM (DTM applicationId: fdtConfiguration)
Print:		May be supported
Device connection:		Shall not have a connection established
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	
	Planning Eng.:	Accessible
UserFlags:		No changes
Use case:		**System generation**
Brief description:		Use case to perform all necessary actions for the creation of topology based on the result of scan.
GUI:		Frame Application and DTM
Print:		Frame Application and DTM
Device connection:		Necessary
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	
	Planning Eng.:	Accessible
UserFlags:		No changes
Included use case:		**Scan**
Brief description:		Retrieves the list of available devices with service scan from a channel object. The channel object may be provided by a DTM or by Frame Application.
GUI:		The Frame Application may offer a GUI for representation of the list of devices.
Print:		No support
Device connection:		Needs a temporary connection. (The channel shall have online access.)
Included use case:		**Network management**
Brief description:		Use case for network management purposes, e.g. address setting as a function of a Communication-DTM. (DTM applicationId: fdtNetworkManagement)
Included use case:		**DTM matching**
Brief Description:		Use case for finding a DTM that matches best the information retrieved in the Scan use case.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Included use case:		Create instance
Brief description:		In this use case a new device instance can be created and placed into the project. After creation of a new device instance, the device parameter set shall be instantiated. This action is performed by the DTM.
GUI:		Frame Application and DTM (if it supports device default selection)
Use case:		**System planning**
Brief description:		Use case to perform all necessary actions for the editing of topology information.
GUI:		Frame Application and DTM
Print:		Frame Application and DTM
Device connection:		Not necessary
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	
	Planning Eng.:	Accessible
UserFlags:		No changes
Included use case:		**Network management**
Brief description:		Use case for network management purposes e.g. address setting as a function of a communication-DTM. (DTM applicationId: fdtNetworkManagement)
Included use case:		**Bus master configuration**
Brief description:		Use case for configuration of bus master DTMs
Included use case:		**Channel assignment**
Brief description:		Use case for editing the FDT channel assignments
Use case:		**List of supported devices**
Brief description:		In this use case a list of all supported devices within a project can be viewed and edited. A “ Planning Engineer” actor can select a device from this list to create a new device instance. An actor with the “Administrator” actor flag set has access to change this list.
GUI:		FA
Print:		No support
Device connection:		No connection
UserLevel	Observer:	No access
	Operator:	
	Maintenance:	
	Planning Eng.:	Accessible for viewing only
UserFlag	Administrator:	Accessible for editing
	OEM Service:	No changes

5.3.6 OEM service

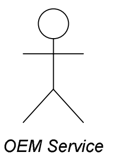


Figure 27 – OEM service

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The “OEM service” uses cases may provide the “OEM service” actor (see Figure 27) extended access to use cases of other actor roles.

5.3.7 Administration

The administration use cases are available to users that have additional Administrator permissions (see Figure 28)

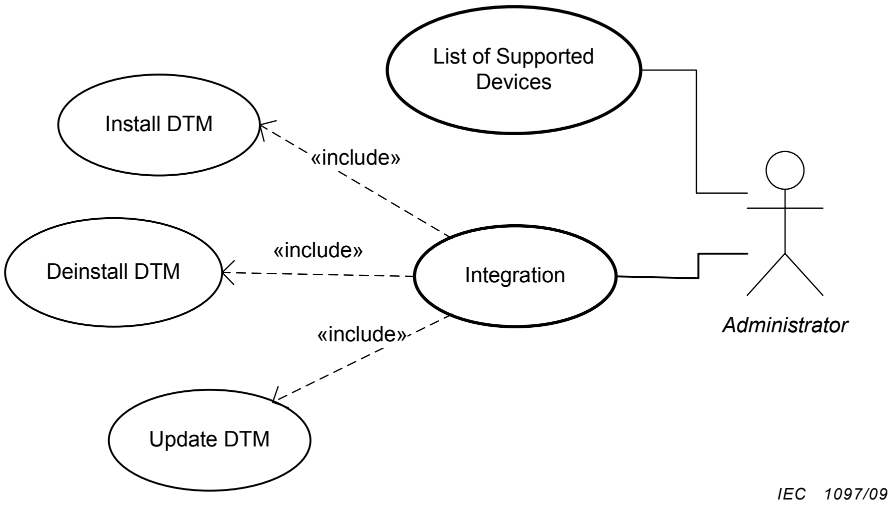


Figure 28 – Administrator use cases

Table 10 provides a description for the Administrator use cases.

Table 10 – Administrator use cases

Use case:		**Integration**
Brief description:		This use case enables the actor to install, deinstall or update new DTMs. Installation and deinstallation are not controlled by the Frame Application.
GUI:		The Frame Application may support the management of DTMs.
Print:		Not supported
Device connection:		Shall not have a connection established
Users:	Operator:	Not accessible
	Maintenance:	
	Planning Eng.:	
	Administrator:	Accessible
	OEM Service:	Not accessible
Included use case:		**Install**
Brief description:		Installation of a new DTM.
GUI:		Responsibility of the DTM
Included use case:		**Deinstall**
Brief description:		Deinstallation of a DTM
GUI:		Responsibility of the DTM
Included use case:		**Update DTM**
Brief description:		Update/modification of a DTM installation
GUI:		Responsibility of the DTM

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

6 General concepts

6.1 Address management

Fieldbus systems generally use an addressing concept to distinguish between the different connected devices. Therefore, a DTM generally needs the information about the address of its associated device in order to establish an online connection to it (see service Connect, 7.6.1.2).

A DTM for a device type which can be connected to a system which defines an addressing concept has to provide the NetworkManagementInfo services (see 7.2.11). These services enable to read and write the device addressing information to the DTM. The addressing schema is fieldbus specific, therefore concrete information which shall be passed to the service is defined by the IEC 62453-3xy documents describing the protocol profile integration in FDT.

Always the component providing the Communication Channel (DTM or Frame Application) is responsible for address setting in the linked DTMs. A DTM providing Communication Channels shall support a Presentation object that enables the user to set the addresses (ApplicationID = fdtNetworkManagement, see Table A.4) as shown in following figure (Figure 29).

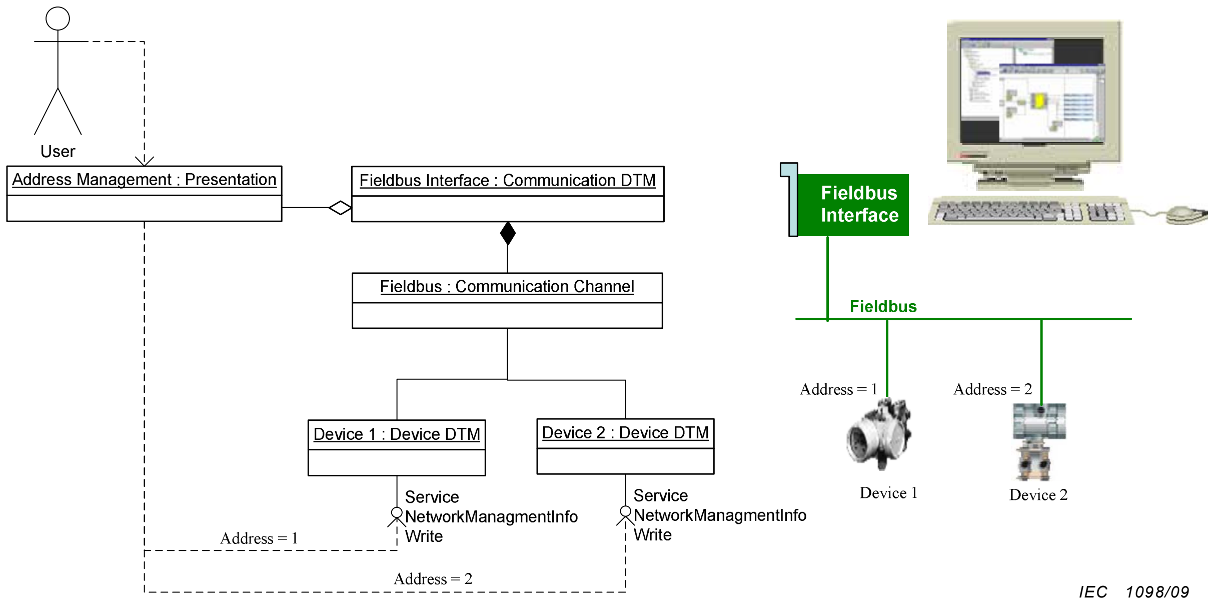


Figure 29 – Address setting via DTM presentation object

Furthermore the Communication Channel itself shall support the service SetChildrenAdresses (see 7.6.2.5) to enable the Frame Application to initiate the address setting. The sequence diagrams in 8.2 describe different addressing scenarios in more detail.

How address management is realized if the Communication Channel is provided by the Frame Application is not within the scope of this standard.

6.2 Scanning and DTM assignment

6.2.1 Scanning introduction

Many fieldbus protocols (or point-to-point communication) define scanning services supporting to request a live list that contains information about connected devices. The information and services described in subsequent clauses enable the Frame Application to

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

generate or validate corresponding FDT topology without the knowledge of the protocol-specific definitions.

6.2.2 Scanning

Scanning is supported by the Communication Channel service scan (see 7.6.4.1).

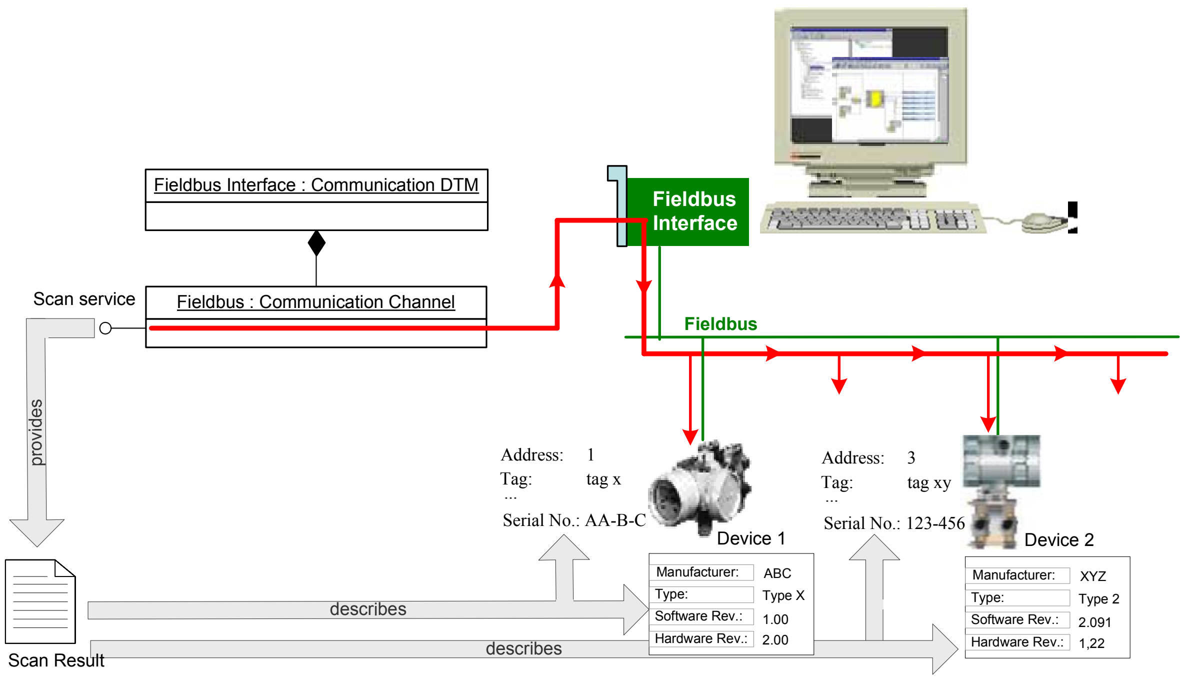


Figure 30 – Fieldbus scanning

The service returns device type and instance related information for all connected devices which can be determined by the means defined by the fieldbus protocol (see Figure 30). For example device manufacturer, type, hardware and embedded software version or fieldbus address, tag and serial number. This information is fieldbus specific. The concrete format and transformation to the protocol-independent format which can be used by Frame Application without fieldbus specific knowledge is defined by the IEC 62453-3xy document describing the protocol profile integration in FDT.

6.2.3 DTM assignment

A Frame Application may use the information returned by a scan (see 7.6.4.1) to find proper DTMs that can be added to the FDT topology by comparing it with the device type identification information return by service GetIdentificationInformation (see 7.2.3.2) of known DTMs.

Furthermore, a Frame Application may also use the scan result information to validate whether DTMs in an existing FDT topology are the proper ones by comparing it with the device type identification information returned by those DTMs.

The comparison is done by the Frame Application based on its internal rules. Each element from the scan information can be compared with the DTM device identification information.

6.2.4 Manufacturer specific device identification

Sometimes devices cannot be uniquely identified by the means defined by the fieldbus protocol. For example because same type ID is used for several different device types of one manufacturer. In this case, the Frame Application may find several DTM device types or even

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

DTMs that appear to be the proper ones for devices found during a scan. However, such a device may be identifiable by means defined by the device manufacturer.

FDT enables the manufacturer to implement these specific device identification within a DTM and make it usable by the Frame Application in a manufacturer and protocol independent way.

The DTM for such a device shall add manufacturer specific identification properties to the device identification information that is returned by GetIdentificationInformation (see 7.2.3.2) If now the Frame Application tries to find a proper DTM then it can only find device identifications which match, but have additional elements which are not included in the scan result, because they are only known by the device manufacture. In such a situation, the Frame Application shall look for a DTM device type for which the DTMSupportLevel is set to 'identSupport' (see Table A.11) and temporarily add this to the FDT topology to request additional information from the device by the service HardwareInformation (see 7.2.3.3). The DTM then shall apply the manufacturer specific device identification means and return the additional device identification elements as a result of the service HardwareInformation call to enable the Frame Application to replace itself with the proper DTM and DTM device type.

6.2.5 Scan for communication hardware

FDT also defines the means to scan for communication hardware acting as the root element in the FDT topology to access the top-level fieldbus in the system topology or a point-to-point connection.

This mechanism is based on a trial and error principle. The Frame Application just instantiates every registered or pre-selected Communication DTM, invokes HardwareInformation (see 7.2.3.3) and checks whether the service returns information about found communication hardware or an error.

6.3 Configuration of fieldbus master or communication scheduler

Many fieldbuses use dedicated devices to control the communication between the different participants, for example the Master-Slave or the Token-Passing concept.

The devices controlling the communication over the fieldbus are called Fieldbus Master or Communication Scheduler. These devices usually need to be configured with the bus parameter information of the devices connected to the fieldbus. In general, the configuration is done with a device-specific configuration tool. The FDT component representing the Fieldbus Master or Communication Scheduler shall provide this configuration tool. That means the configuration tool is either part of the Frame Application or as in Figure 31 part of a DTM.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

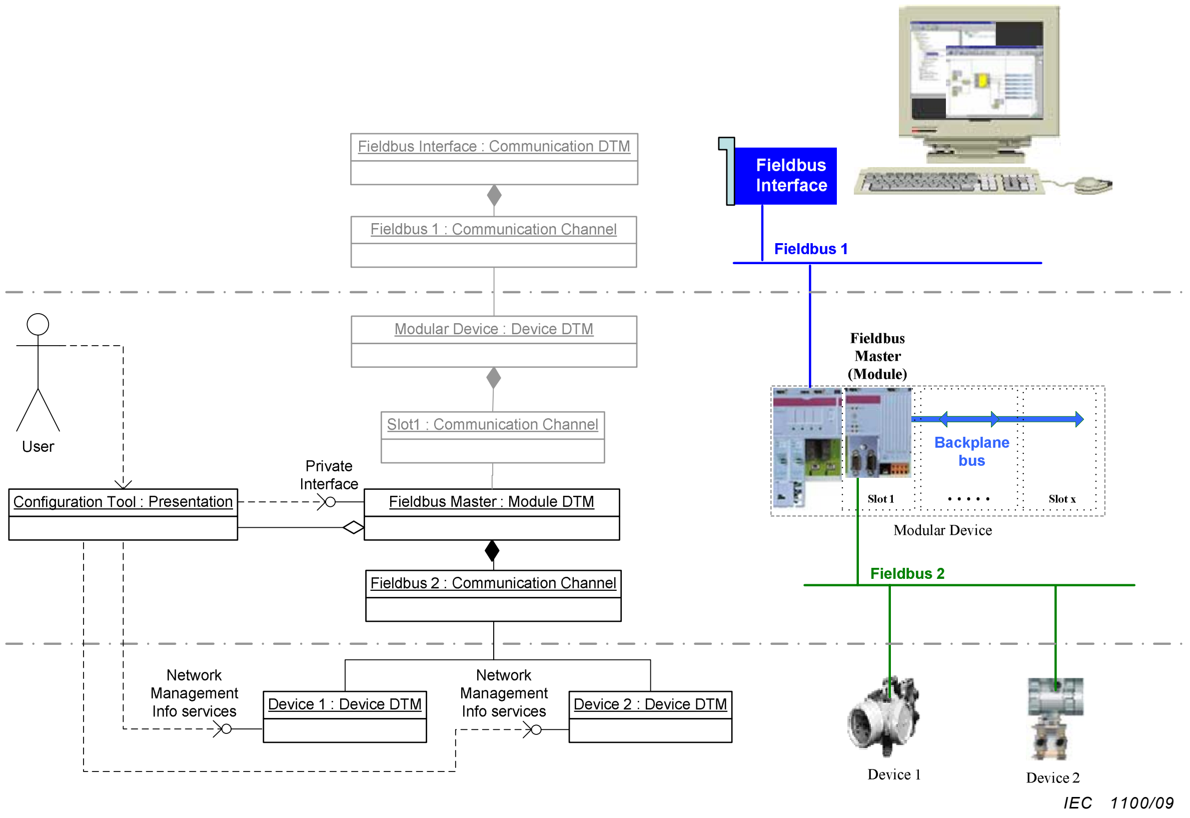

Fieldbus Interface : Communication DTM
Fieldbus 1 : Communication Channel
Modular Device : Device DTM
Slot1 : Communication Channel
User
Private Interface
Configuration Tool : Presentation
Fieldbus Master : Module DTM
Fieldbus 2 : Communication Channel
Network Management Info services
Device 1 : Device DTM
Network Management Info services
Device 2 : Device DTM
Fieldbus Interface
Fieldbus 1
Fieldbus Master (Module)
Backplane bus
Slot 1
Slot x
Modular Device
Fieldbus 2
Device 1
Device 2
IEC 1100/09

Figure 31 – Fieldbus master configuration tool as part of a DTM

The configuration tool is able to read and write the bus parameter information of the fieldbus participants via the NetworkManagementInfo services (see 7.2.11) of corresponding DTMs. The master or communication scheduler bus parameter information may be accessed by private means defined between the configuration tool and corresponding DTM.

After the configuration tool has set all bus parameter information, each DTM is responsible for writing the information to its device via standard FDT communication operations, for example if a download is requested via service WriteDataToDevice (see 7.2.12.3).

The concrete bus parameter information which can be read or written via NetworkManagementInfo services is defined by the IEC 62453-3xy documents describing the protocol profile integration in FDT. Furthermore, the general bus configuration is also described in more detail in the IEC 62453-3xy document, which describes the protocol.

6.4 Slave redundancy

6.4.1 Redundancy overview

The duplication of critical components of a system with the intention of increasing reliability of the system, usually in the case of a backup or fail-safe, is called redundancy. Figure 32 shows most common redundancy scenarios in the automation industry.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

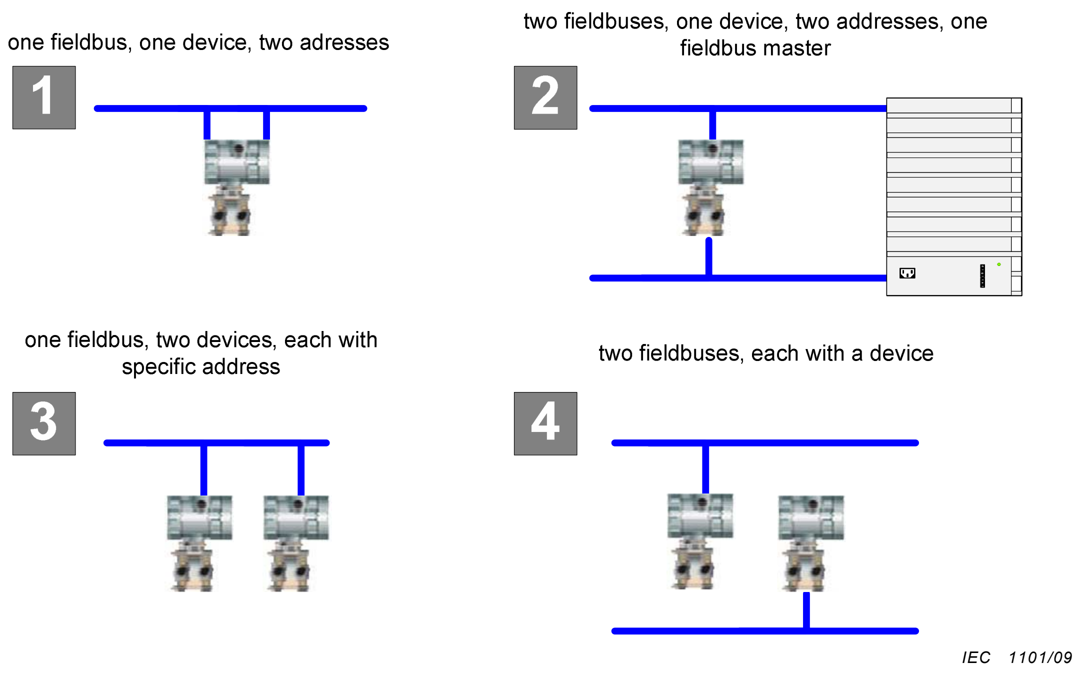

one fieldbus, one device, two adresses
1
two fieldbuses, one device, two addresses, one fieldbus master
2
one fieldbus, two devices, each with specific address
3
two fieldbuses, each with a device
4
IEC 1101/09

Figure 32 – Redundancy scenarios

The scope of slave redundancy within FDT is one DTM for configuration of one device connected according to either one of following scenarios:

- scenario 1: one fieldbus using two different addresses;
- scenario 2: connected to two different fieldbuses controlled by one bus master.

Separate redundant devices (scenarios 3 and 4) cannot be handled within FDT, these scenarios would require additional Frame Application functionality, for example with regard to data synchronization.

In scenarios 1 and 2, redundant fieldbuses shall be managed by one redundancy aware parent component (e.g. a Communication DTM) specific for the appropriate redundant field bus technology. This parent component should typically also be able to handle DTMs for redundant and non-redundant field devices in parallel.

6.4.2 Redundancy support in Frame Application

The DTMs able to handle slave redundancy can be used by any FDT Frame Application without need for specific redundancy support or knowledge. A FDT Frame Application may optionally support the services OnAddedRedundantChild (see 7.7.4.1) and OnRemovedRedundantChild (see 7.7.4.2) to get notified and be able to display redundant slaves in its topology view, for example a field device connected to two different lines or by using specific device icons within a topology view.

Within a Frame Application which is not aware of redundancy, a DTM representing a redundant slave cannot be distinguished within the topology view. But the DTM and the parent component themselves typically display additional information, for example additional fieldbus addresses. Nevertheless within such a Frame Application these DTMs provide full functionality with regard to fieldbus redundancy.

As a redundant slave is represented by one DTM instance no additional functionality with regard to dataset synchronization or multiuser is required.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

6.4.3 Parent component for redundant fieldbus

Fieldbus communication for redundant slaves is handled by a fieldbus and redundancy technology specific parent component, for example a specific Communication DTM. All fieldbuses used to connect redundant slaves shall be handled by one instance of such a parent component. In scenario 2 for example, each fieldbus line is represented by an appropriate Communication Channel of the parent component. Therefore, a Frame Application is able to use such a parent component without knowledge about the physical redundant fieldbus structure.

During a service ValidateAddChild (see 7.6.2.3) and service ChildAdded (see 7.6.2.2) call the parent component and is able to detect if a Device DTM to be added is able to handle redundancy by examining the information that is returned by service NetworkManagementInfoRead (see 7.2.11.1) of the instantiated DTM. In this case, a specific address selection dialog within the parent component can be used. This address selection is fieldbus and redundancy specific. It is up to the parent component if redundancy is handled automatically or only after user interaction.

If the Frame Application is aware of redundancy the parent component has to inform the Frame Application about the redundant DTM via OnRemovedRedundantChild (see 7.7.4.2).

The parent component shall be able to handle all possible communication paths to the redundant device. Within a service NetworkManagementInfoWrite (see 7.2.11.2) call complete redundant address information can be provided to the DTM instance handling this redundant device. This DTM is then able to use this information to display all available communication paths. During a service Connect (see 7.6.1.2) the DTM provides this complete address information to the parent component. The parent component is then able to select the currently active communication path. The communication path can be changed within the parent component without need for interaction with the Device DTM.

A parent component may also be able to handle a DTM representing a non-redundant field device. In this case, this DTM is handled without using the redundancy specific data definitions in FDT.

6.4.4 Redundancy support in Device DTM

A DTM able to handle a redundant device provides additional information in service NetworkManagementInfoRead (see 7.2.11.1).

After a service ChildAdded (see 7.6.2.2) call complete redundant address information can be given by the parent component to the Device DTM. The Device DTM is then able to detect if its appropriate slave is used as a redundant slave. This complete address information shall be used by the Device DTM within a Connect (see 7.6.1.2), but can also be used for diagnosis and status information. Typically additional redundancy handling is required within the Device DTM itself.

A DTM handling a redundant device shall check all addresses provided by a NetworkManagementInfoWrite (see 7.2.11.2) call. If the Device DTM is not able to handle these addresses, for example because only a vendor-specific offset can be used, an error message should be created and negative success should be returned to the calling parent component. In this case, the parent component is able to detect Device DTMs which cannot be used at the redundant fieldbus. A Device DTM shall save all redundancy information in its instance data.

A DTM representing a redundant slave can be used as child of a non-redundant aware parent component. In this case, the Device DTM shall not use the additional FDT redundancy functionality, GUIs or redundant specific data definitions in FDT.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

6.4.5 Scan and redundant slaves

Fieldbus scan provides for each address of a redundant slave one entry, a redundant slave cannot be detected.

In scenario 1, mapping of redundant addresses to one instance is only possible for a DTM itself. In scenario 2, mapping can only be done based on project knowledge. Typically, a scan is not executed on a redundant system.

7 FDT service specification

7.1 Service specification overview

This clause specifies the services defined for FDT objects. The service definitions are abstract descriptions and do not represent a specification for implementation. The mapping between the abstract descriptions and the actual implementations derived from these services are defined in technology specific implementation parts.

The services for each object are organized into service sets. Each service set defines a set of related services. The organization in service sets is a logical grouping used in the specification and may not be used in the implementation.

Each service is documented with

- service abstract;
- table for request and response values;
- service description;
- state availability statement.

The service abstract gives a brief overview.

Within the table for request and response values, the arguments which will be used for the service are described. For each argument (see Annex A for a description of the argument data types) it is defined whether the argument is mandatory (‘M’) or optional (‘O’) used for the service call (Request) or as return value (Response) or not needed (‘-‘).

The description of the service defines the behaviour and usage of the service.

The state availability statement defines for each service whether the service is available in state ‘configured’ or ‘communication allowed’

This part of IEC 62453 does not define whether it is optional or mandatory to implement the defined services. This definition is provided by the technology specific implementation parts. Each technology specific part provides an annex describing the actual implementation of the services (for example mapping of services to interface methods).

NOTE The mechanism for informing about not implemented services may also be technology specific. For some technology, not implemented service may be recognized by the client if a server does not provide certain interfaces. For other technologies, not implemented services may be recognized by special service responses.

Whether the services are executed in a synchronous or asynchronous model depends on the implementation technology. ‘Synchronous’ means that the service caller is blocked during the complete execution time of the service. ‘Asynchronous’ means that the caller is able to continue with other operations until service execution is complete. The used service execution model is defined in the IEC 62453-4z document describing the mapping to certain implementation technology. Both execution models may be used mixed for implementation of services. In addition, canceling of an active service execution may be defined.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.2 DTM services

7.2.1 General services

7.2.1.1 Service PrivateDialogEnabled

This service is used by the Frame Application to notify a DTM whether it is allowed to open a private dialog window.

Table 11 – Arguments for service PrivateDialogEnabled

	Request	Response
Enabled (Boolean)	M	-

This special condition is set by a Frame Application during runtime. This may be necessary for instance if

- a running user action shall not be disturbed;
- it is not allowed that a dynamically opened window gets the focus or hides other windows, or
- the DTM is executed on a different host than the GUI.

Enabled set to FALSE means that the DTM shall not open any GUI other than GUIs that are opened by FA services (e.g. service OpenPresentationRequest and service UserDialog service). If under this condition any error occurs, the DTM is still able to send a notification to the Frame Application via service OnErrorMessage (see 7.7.2.1).

If the DTM needs a user action, and therefore has to open a user dialog, it should call the Frame Application service UserDialog (see 7.7.7.3). If no GUI can be shown, the default behavior at this request is not to open the dialog and to provide the DTM with a default return value. So it is up to the Frame Application to allow opening the dialog, delay the request until the current user action is finished, or to refuse the request by sending the default response.

DTMs should inform user via service UserDialog if a specific functionality cannot be performed because private dialogs are not allowed.

Examples for private dialogs are:

- message boxes (i.e. standard message box);
- file or printer selection dialogs (i.e. provided by operating system);
- (default) web browsers;
- (default) mail clients;
- help file view;
- manual viewer;
- splash screens;
- external stand-alone applications,
- any other windows.

If the dialogs are opened under control of the Frame Application, they are defined not to be private dialogs.

Examples are:

- dialogs opened by service UserDialog;

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

- DTM GUI opened by service OpenPresentationRequest;
- applications started by service OpenPresentation;
- windows opened by the Frame Application due to a <Document> entry in the response received from GetFunctions.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.1.2 Service SetLanguage

This service is used by the Frame Application to notify a DTM during initialization about the language to use in all Presentation objects and for user or error messages.

Table 12 – Arguments for service SetLanguage

	Request	Response
LanguageID	M	-

The Frame Application sets the language during initialization of the DTM, so that all Presentation objects of the same instance have the same language. Also, the messages on the event service like service OnErrorMessage (see 7.7.2.1) and the human readable contents returned by services GetTypeInformation (see 7.2.3.1) or GetDocumentation (see 7.2.7.1) shall use the requested language. If a DTM does not support the requested language, it shall use the current language (if it is already initialized) or on the first initialization set the current language to its default language.

The supported languages of a DTM are listed in the information returned by service GetTypeInformation (see 7.2.3.1). One DTM instance is always initialized with one language. It is up to a DTM whether it is able to change the current language of a Presentation object while it is shown. The output format for numbers, currencies, times, and dates shall be based on the regional options of the operation system.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.1.3 Service SetSystemGuiLabel

This service is called by the Frame Application to set a unique human readable identifier of the DTM instance in the context of the Frame Application.

Table 13 – Arguments for service SetSystemGuiLabel

	Request	Response
windowTitle	M	-

This service is called by the Frame Application in order to set a system label, for example for a message box or a Presentation object which is part of the DTM and is not to be embedded within a Frame Application user interface. The Frame Application shall set the unique human readable identifier of the DTM instance in the context of the Frame Application which guarantees a unique identification between the device and, for example a message box of a DTM. The DTM shall use this system label for all kinds of windows that will be opened by the DTM themselves. The DTM may extend this title with specific information. The Frame Application has to call this service as early as possible. As long as the service is not called, the DTM has to use the tag of the device as human readable string by default.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The human readable identifier shall not be stored by the DTM. It is mandatory to update the labels of all open windows when this service is called.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.2 DTM services related to installation

Each DTM which should be visible for integration within the Frame Application shall be registered within the system. In addition to the registration, the DTM shall give information whether the DTM represents a device, module or block.

DTMs shall inform the system whenever the DTM is:

- installed;
- uninstalled;
- modified, (e.g. new device types are supported or the DTM was updated (bug fix)).

By using this information, a Frame Application is able to build a database with all available DTMs on the system. The detailed implementation depends on the used technology and is described in an IEC 62453-4z document.

7.2.3 DTM services related to DTM/device information

7.2.3.1 Service GetTypeInformation

Returns general DTM type information like name, version, vendor and supported DTM device types.

Table 14 – Arguments for service GetTypeInformation (for DTM)

	Request	Response
fdt:VersionInformation	-	M
fdt:Version	-	M
fdt:DtmDeviceTypes	-	M

This service provides access to DTM specific information. This data are available as soon as the DTM is instantiated. The DTM does not need any instance specific data to provide this information.

Usually this information is used by the Frame Application to create libraries, to select a DTM, or for simple validations.

If service is called at a BTM, then the following service request and response parameters are used.

Table 15 – Arguments for service GetTypeInformation (for BTM)

	Request	Response
fdt:VersionInformation	-	M
fdt:Version	-	M
btm:BtmBlockTypes	-	M

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service is available in states:
[X] ‘configured’
[X] ‘communication allowed’

7.2.3.2 Service GetIdentificationInformation

Requests device identification information for specified DTM device type and protocol.

Table 16 – Arguments for service GetIdentificationInformation (for DTM)

	Request	Response
fdt:DtmDeviceType	M	-
fdt:ProtocolId	M	-
fieldbus:DeviceTypeIdentifications or devIdent:DeviceTypeIdentificationList	-	M

This service returns device identification information for a requested device or block type. The response may contain also protocol specific information.

If service is called at a Communication-DTM, then response contains devIdent: DeviceTypeIdentification List.

If service is called at a BTM, then the following service request and response parameters are used.

Table 17 – Arguments for service GetIdentificationInformation (for BTM)

	Request	Response
btm:BtmBlockType	M	-
fdt:ProtocolId	M	-
fieldbus: DeviceTypeIdentifications	-	M

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.3.3 Service HardwareInformation

Requests scan for availability of hardware and corresponding identification information.

Table 18 – Arguments for service Hardware information (for DTM)

	Request	Response
ScanIdentificationList	-	M
NOTE ScanIdentificationList may be either ident:ScanIdentificationList or fieldbus:ScanIdentifications.		

The Frame Application executes this function to check if corresponding hardware is responsive and to request identification information.

This service is available in states:
[_] ‘configured’;
[X] ‘communication allowed’.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.2.3.4 Service GetActiveTypeInfo

The DTM shall provide information about the selected DTM device type.

Table 19 – Arguments for service GetActiveTypeInfo

	Request	Response
fdt:DtmDeviceType	-	M
net:DtmDeviceInstanceTopology	-	O

The service shall always return the current DTM device type. That means in case of an update of the DTM, the changed DTM device type shall be returned (e.g. higher software version) instead of the DTM device type given during service Initialize (see 7.2.4.1) call.

Optionally, the service returns information about the internal structure of the corresponding device. The returned information is protocol- and device specific.

If the service is called at a BTM, then the following service request and response parameters are used.

Table 20 – Arguments for service GetActiveTypeInfo (for BTM)

	Request	Response
btm:BtmBlockType		M -

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.4 DTM services related to the DTM state machine

7.2.4.1 Service Initialize

The service initializes the DTM in order to bring it to a working state.

Table 21 – Arguments for service Initialize (for DTM)

	Request	Response
fdt:DtmDeviceType	M	-
user:FDTUserInformation	M	-
fdt:FrameVersion	O	-
fdt:SystemTag	M	-
fdt:FrameObjectReference	M	-
fdt:serviceSucceeded	-	M

The DTM receives information regarding the device type that it will represent, regarding the user that is going to access the DTM, regarding the runtime environment, the identifier of the DTM and a reference to the Frame Application itself.

In general, it is expected that a DTM adapts the provided functionality according to the role of the current user.

Also it is expected that the DTM adapts its behaviour regarding the FDT functionality of the runtime environment (according to FrameVersion).

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

If service is called at a BTM, then the following service request and response parameters are used.

Table 22 – Arguments for service Initialize (for BTM)

	Request	Response
btm:BtmBlockType	M	-
user:FDTUserInformation	M	-
fdt:FrameVersion	O	-
fdt:SystemTag	M	-
fdt:FrameObjectReference	M	-
fdt:serviceSucceeded	-	M

This service is available in states:
[X] Initial State;
[_] ‘configured’;
[_] ‘communication allowed’.

Calling this service leads to a state transition to state ‘configured’.

7.2.4.2 Service SetLinkedCommunicationChannel

The service informs the DTM about the linked Communication Channel within the FDT topology which shall be used to communicate with its device.

Table 23 – Arguments for service SetLinkedCommunicationChannel

	Request	Response
fdt:CommunicationObjectReference	M	-
fdt:serviceSucceeded	-	M

The Communication Channel is set by the Frame Application: The DTM is able to check the supported communication protocols by calling the service GetSupportedProtocols (see 7.6.1.1) and the GatewayBusCategory information by calling the the ReadChannelData (see 7.5.1.1). In general, the Frame Application is responsible for establishing a valid link between a Communication Channel and a DTM (see 4.3), but this check may be for example necessary if a DTM supports multiple protocols and wants to find out which shall be used.

This service is available in states:
[X] ‘configured’;
[_] ‘communication allowed’.

7.2.4.3 Service EnableCommunication

The service allows or permits a DTM to communicate with its device.

Table 24 – Arguments for service EnableCommunication

	Request	Response
Boolean	M	-
fdt:serviceSucceeded	-	M

The service is called by the Frame Application with TRUE to bring the DTM from state ‘configured’ into the state ‘communication allowed’. Afterwards, the DTM is allowed to establish a communication link to its device by calling the Communication Channel service Connect (see 7.6.1.2).

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The service is called with FALSE to bring the DTM from state ‘communication allowed’ back into the state ‘configured’. If the DTM has accepted the call (fdtServiceSucceeded = TRUE), then all established communication links shall be closed (see services Disconnect (7.6.1.3) or AbortRequest (7.6.1.4)). The DTM is allowed to reject the (fdt:serviceSucceeded = FALSE) if communication service calls are active which cannot be terminated.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.4.4 Service ReleaseLinkedCommunicationChannel

The service informs the DTM that that it shall release the Communication Channel set by service SetLinkedCommunicationChannel (see 7.2.4.2).

Table 25 – Arguments for service ReleaseLinkedCommunicationChannel

	Request	Response
fdt:serviceSucceeded	-	M

This service is available in states:
[X] ‘configured’;
[_] ‘communication allowed’.

7.2.4.5 Service ClearInstanceData

Used to inform the DTM that it has to clean up, for example its log files or protocols. The instance data will be deleted by the Frame Application.

Table 26 – Arguments for service ClearInstanceData

	Request	Response
fdt:serviceSucceeded	-	M

The service is used to inform a DTM that it has to clean up, for example its log files or protocols in its private storage (see 4.8). After this service call, the instance data will be deleted by the Frame Application. The Frame Application is responsible for ensuring the pre-conditions for the delete. That means the Frame Application shall ensure, that all started DTM instances related to this instance data are terminated.

This service is available in states:
[X] ‘configured’;
[_] ‘communication allowed’.

7.2.4.6 Service Terminate

This service informs the DTM that it has to release its references to other components. The DTM will be terminated by the Frame Application.

Table 27 – Arguments for service Terminate

	Request	Response
fdt:serviceSucceeded	-	M

The DTM has to release all references to other components and has to terminate all pending or running functions. It shall also close all user interfaces. It is up to a DTM to decide, if transient data should be stored or not.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service is available in states:
[X] ‘configured’;
[_] ‘communication allowed’.

7.2.5 DTM services related to functions

7.2.5.1 Service GetFunctions

Returns information about standard (defined by ApplicationID) or additional functionalities (defined by FunctionId) supported by a DTM.

Table 28 – Arguments for service GetFunctions

	Request	Response
fdt:OperationPhase	M	-
func:Functions	-	M

This service provides information about DTM functions that enable the Frame Application for example to create a DTM specific menu. This data is available as soon as the DTM is instantiated but the information may change if it is instance specific.

A DTM shall inform the Frame Application about any function changes (e.g. number, status, labe,l etc.) by calling the service OnFunctionChanged (see 7.7.2.4),

Information about how to execute functions with user interface shall be requested by service GetGuiInformation (see 7.2.5.3) function calls without user interface shall be started by service InvokeFunction (see 7.2.5.2).

The service additionally provides access to documents provided by a DTM, which can be displayed by the Frame Application or a special viewer program.

It is expected that a DTM adapts the provided list of functions according to the role of the current user. As long as the DTM does not have valid user information, it should behave like the user with the least rights (“Observer”) is using the DTM.

Functions with associated module Ids (so-called module functions) may not be shown directly in the standard context menu of the DTM node. They may appear in a submenu which is associated to a DTM module or in the context menu of a DTM module node. In order to ensure that module functions can be called by the user, the Frame Application should provide a submenu “Modules” in the context menu of the DTM node or provide appropriate context menus for DTM modules or offer these functions by other means.

The DTM has to ensure that all accessible module functions (enabled and not hidden) are valid. That means the Frame Application is not responsible for matching the given module Ids to the list of the existing modules returned by service GetActiveTypeInfo (see 7.2.3.4).

If the DTM disables a group of functions (Functions data type), it shall also disable all functions (Function data type) below that group if this is the intended behavior. If the DTM does not disable sub-functions, the Frame Application can still make use of them.

If a Frame Application does not support the operation phases (‘notSupported’) the DTM should provide all functions based on the current state and the current user role.

If the DTM wants to limit the number of opened user interfaces, it should disable the corresponding functions. If a user interface is closed, the DTM has to enable the function again.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.5.2 Service InvokeFunctions

Starts a function of a DTM without user interface.

Table 29 – Arguments for service InvokeFunctions

	Request	Response
func:FDTFunctionCall List	M	-

This service executes a DTM function provided without a user interface . Information about functions supported by a DTM is returned by service GetFunctions (see 7.2.5.1).

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.5.3 Service GetGuiInformation

This service provides information on how to start the user interface of a DTM.

Table 30 – Arguments for service GetGuiInformation

	Request	Response
func:FDTFunctionCall	M	-
func:GUIInformation	-	M

It returns information that enables the Frame Application to instantiate the user interface. Dependent on the returned information, the Presentation object may be a type of object that allows integration into the GUI of the Frame Application (see 7.7.7.1) or it may be an object that runs outside of the GUI of the Frame Application (see 7.2.5.4).

Integration and Interaction between these objects is technology dependent and defined within IEC 62453-4z documents

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.5.4 Service OpenPresentation

It requests the DTM to open a user interface for a specific function call.

Table 31 – Arguments for service OpenPresentation

	Request	Response
fdt:invokeID	M	
func:FDTFunctionCall	M	
fdt:windowTitle	M	
fdt:DisplayContext	O	
fdt:serviceSucceeded	-	M

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The FDTFunctionCall requests opening of corresponding Presentation object. In general, it is up to the Frame Application to determine the passed FDTFunctionCall and the DTM decides the kind of presentation.

This service shall always bring a user interface to the foreground, or at least an error message. Already started Presentation objects, identified by the InvokeID, shall be popped to the foreground. The request of an already started Presentation object with a new InvokeID shall be rejected by the DTM.

The InvokeID is used by a Frame Application for the association within service ClosePresentation (see 7.2.5.5) request. Furthermore, it allows the Frame Application to handle a list of open user interfaces.

This service is available in states:
[X] 'configured';
[X] 'communication allowed.'

7.2.5.5 Service ClosePresentation

Request to a DTM to close a user interface identified by InvokeID.

Table 32 – Arguments for service ClosePresentation

	Request	Response
fdt:invokeID	M	
fdt:serviceSucceeded	-	M

The DTM just checks whether it is able to close the user interface or not. In the case it is able, it returns success and shall start its shut down procedure for the user interface. During this shut down it has to unlock its instance data and release the online connection to its device if necessary. The DTM itself is not terminated.

In case of errors, the DTM shall supply further details by calling the service OnErrorMessage (see 7.7.2.1)

The InvokeID is used by a Frame Application for the association to the appropriate service OpenPresentation (see 7.2.5.4) request.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.6 DTM services related to channel objects

7.2.6.1 Service GetChannels

Returns the channel objects of a DTM.

Table 33 – Arguments for service GetChannels

	Request	Response
ChannelObjectReference List	-	M

This service returns the channels of a DTM.For simple devices, a channel object provides only the information for channel assignment.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.7 DTM services related to documentation

7.2.7.1 Service GetDocumentation

This service provides the DTM specific documentation for a specific DTM function.

Table 34 – Arguments for service GetDocumentation

	Request	Response
func:FDTFunctionCall	M	-
doc:Documentation	-	M

This service shall return information which can be used directly for documentation purposes. The content of the returned information is defined by the passed function call id, which is available via service GetFunctions (see 7.2.5.1). Only functions with the attribute Printable shall be supported. The Frame Application shall transform the returned information to a user readable format.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.8 DTM services to access the instance data

These services enable a Frame Application the access to device parameters. The DTM provides its current in-memory representation of its instance data. It is up to a DTM and depends on the device- and fieldbus-type which parameters are available.

7.2.8.1 Service InstanceDataInformation

Returns information about the device specific parameters.

Table 35 – Arguments for service InstanceDataInformation

	Request	Response
fdt:DatasetState	-	M
fdt:StorageState	-	M
param: DtmItemInfoList	-	M
fdt:ModifiedInDevice		M

This service provides information about the device specific parameters and state. The information may contain information related to configuration parameters as well as asset management related data or can be empty for devices without public data. The contents of the provided information may also depend on the current configuration of the device and the user roles.

Several responses may be provided by the DTM if the list or status of parameters changes.

The service provides the transient data of a DTM. That means, if the DTM is active or has for example an open user interface the provided parameter can differ from the actual stored instance data. The state of the received data is classified.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Parameters provided may also be available as channel objects (provided by service ReadChannelData (see 7.5.1.1). The relation of these items can be identified by the information ‘SemanticId’ (see Table A.4).

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.8.2 Service InstanceDataRead

The service provides read access to DTM instance data.

Table 36 – Arguments for service InstanceDataRead

	Request	Response
param:ItemId list	M	-
param:DtmItem list	-	M

Via this service a Frame Application is able to request data from DTM instance data.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.8.3 Service InstanceDataWrite

The service provides write access to DTM instance data.

Table 37 – Arguments for service InstanceDataWrite

	Request	Response
param:DtmItem list	M	M

Via this service, the Frame Application requests a DTM to write the specified data to its instance data. Furthermore, the DTM has to apply the business rules in order to keep the instance data consistent.

The response contains the device data of the successfully written data specified by the request (may differ compared to the written value due to, for example rounding procedures within the device).

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.9 DTM services to evaluate the instance data

7.2.9.1 Service Verify

Validates the complete instance data by internal business rules of the DTM.

Table 38 – Arguments for service Verify

	Request	Response
fdt:serviceSucceeded	-	M

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

A Frame Application calls this service typically to ensure a consistent dataset, for example after persistent load or before going online.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.9.2 Service CompareDataValueSets

Compares the instance data of an external DTM supporting the same DatasetId with its own.

Table 39 – Arguments for service CompareDataValueSets

	Request	Response
fdt:SystemTag	M	-
fdt:serviceSucceeded	-	M

The structure and the parameter values for configuration, parameterization and identification are relevant for the comparison. Runtime dependent parameters (e.g. operation hours) of the data are not relevant for comparison.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.10 DTM services to access the device data

These services provide a Frame Application online access to specific parameters of a device. The DTM shall be prepared for multiple asynchronous requests in parallel. The requests shall be processed in the order received.

The service shall not modify the instance data.

7.2.10.1 Service DeviceDataInformation

This service provides a list of the available device specific parameters and process values.

Table 40 – Arguments for service DeviceDataInformation

	Request	Response
param:DtmItemInfo list	-	M

Provides a list of the available device specific parameters and process values. Within a DTM this list may contain items related to configuration parameters, process values as well as asset management related data like stroke counter. In DTM state ‘configured’ the returned item list is based on the current instance data, which could be different from the configuration of the device. In this case, services DeviceDataRead (see 7.2.10.2) and DeviceDataWrite (see 7.2.10.3) may fail.

The service provides a list of items that can be read or written from/to the DTM via DeviceDataRead or written to the DTM via DeviceDataWrite The source for this data is the device itself.

Items provided within this list may also be available by channel service, service ReadChannelData (see 7.5.1.1) or DTM service InstanceDataInformation (see 7.2.8.1). The related items can be identified via the ‘SemanticId’ (see Table A.4)

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The information may depend on the current configuration of the device. Several responses may be provided by the DTM if list or status of parameters is changed.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.2.10.2 Service DeviceDataRead

This service provides read access to device data.

Table 41 – Arguments for service DeviceDataRead

	Request	Response
param:ItemId list	M	-
param:DtmItem list	-	M

This service enables a Frame Application to request data from a device. Error information shall be handed over to the Frame Application by the related response.

Execution of the service shall not change the data of the instance data.

The DTM shall always accept the request. If the request cannot be processed, the reason for failure shall be provided as part of the response.

The DTM shall be able to handle more than one request at a time.

This service is available in states:
[_] ‘configured’;
[X] ‘communication allowed’.

7.2.10.3 Service DeviceDataWrite

This service provides write access to device data.

Table 42 – Arguments for service DeviceDataWrite

	Request	Response
param:DtmItem list	M	M

This service enables the Frame Application to request a DTM to write the specified data to its device according to the device specific rules. Error information shall be handed over to the Frame Application by the related response.

Execution of this service shall not change the data of the instance data.

The DTM has to check, whether it could manipulate the flag ‘ModifiedInDevice’ (see Table A.4) in the instance data by requesting a lock. If the lock request fails the DTM has also to refuse the DeviceDataWrite call - an appropriate response shall be provided.

The DTM shall always accept the request. If the request cannot be processed, the reason for failure shall be provided asynchronously as part of the response.

The DTM should be able to handle more than one request at a time.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Response as DtmItem list contains the device data of the successfully written data (may differ to the written value due to, for example rounding procedures within the device).

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.2.11 DTM services related to network management information

7.2.11.1 Service NetworkManagementInfoRead

This service provides read access to network management information.

Table 43 – Arguments for service NetworkManagementInfoRead

	Request	Response
net:NetworkInfo	-	M

This service provides write access to network management information. The returned information is protocol specific. It for example contains device bus-address, tag and further bus specific configuration settings.

The usage of this service is further described in 6.1.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.11.2 Service NetworkManagementInfoWrite

This service provides write access to network management information.

Table 44 – Arguments for service NetworkManagementInfoWrite

	Request	Response
net:NetworkInfo	M	-

This service provides write access to network management information which can be read by service NetworkManagementInfoRead (see 7.2.11.1).

The usage of this service is further described in 6.1.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.12 DTM services related to online operation

7.2.12.1 Service DeviceStatus

This service provides information about the status of the device.

Table 45 – Arguments for service DeviceStatus (for DTM)

	Request	Response
status:DTMDeviceStatus	-	M

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

The DTM shall load the current status from the device. Depending on the fieldbus protocol, the DTM may additionally upload its current diagnosis information. Depending on this information, the DTM shall provide status within human readable format. The function shall not open a user interface to allow the check of complete networks.

A BTM shall return the status of the related block.

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.2.12.2 Service CompareDataValueSetWithDeviceData

This service compares DTM instance data with device data. .

Table 46 – Arguments for service CompareInstanceDataWithDeviceData (for DTM)

	Request	Response
onlineComp:DTMOnlineCompare	-	M

This service is used for batch processing and shall work without user interface. It compares its instance data with the device data uploaded from its device. If the DTM is able to upload the data from the device, then it shall compare the data and return whether the data are equal or not. Otherwise, the response shall contain communication error information.

If the DTM has no comparable online data, it shall return 'noComparableData' as value of the 'StatusFlag'.

Comparison should only include data significant for the configuration, parameterization and identification of the device. Data related to runtime (e.g. operation hours) should not be included.

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.2.12.3 Service WriteDataToDevice

This service requests to write online data to the device.

Table 47 – Arguments for service WriteDataToDevice (for DTM)

	Request	Response
fdt:serviceSucceeded	-	M

This service is initiated by the Frame Application. It is mandatory for all DTMs, which shall be loaded during commissioning. The service is used to write all parameters needed for commissioning of the device from DTM's instance data to the device.

In case of errors, the DTM shall inform the Frame Application via the service OnErrorMessage (see 7.7.2.1).

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.2.12.4 Service ReadDataFromDevice

This service requests to read online data from the device.

Table 48 – Arguments for service ReadDataFromDevice(for DTM)

	Request	Response
fdt:serviceSucceeded	-	M

This service is initiated by the Frame Application. It is mandatory for all DTMs, which shall be loaded during commissioning. The service is used for DTM to read all parameters needed for commissioning of the device from the device into DTM's instance data.

In case of errors, the DTM shall inform the Frame Application via the service, service OnErrorMessage (see 7.7.2.1)

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.2.13 DTM services related to data synchronization

7.2.13.1 Service OnLockInstanceData

In case of a multi-user environment, it is necessary to inform the DTM about its current access rights to its instance data especially if another DTM gets the write access to the instance data. See 8.5.

Table 49 – Arguments for service OnLockInstanceData

	Request	Response
fdt:SystemTag	M	-
user:UserName	M	-

If another DTM has locked instance data via service LockInstanceData (see 7.7.6.1) the Frame Application has to send OnLockInstanceData to all DTM instances, which have a reference to the same instance data.

Receiving this notification, a DTM shall not allow any operations to modify the instance related data, for example by disabling its input fields in case of an open user interface.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.13.2 Service OnUnlockInstanceData

In case of a multi-user environment, it is necessary to inform a DTM about its current access mode especially if another DTM gets the read/write access for its instance related data.

Table 50 – Arguments for service OnUnlockInstanceData

	Request	Response
fdt:SystemTag	M	-
user:UserName	M	-
ServiceSucceeded	-	M

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

If a DTM has unlocked an instance data via service UnlockInstanceData (see 7.7.6.2) the Frame Application has to send information via OnUnlockInstanceData to all DTMs which have a reference to the same instance data. When receiving this notification, a DTM supporting data synchronization returns positive response and shall allow alternation of instance data. DTMs which do not support data synchronization shall return negative response value and shall not allow any data modifications. See 8.5.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.13.3 Service OnInstanceDataChanged

In case of a multi-user environment, it is necessary to inform the DTM about data changes.

Table 51 – Arguments for service OnInstanceDataChanged

	Request	Response
fdt:SystemTag	M	-
Information	M	-

If a DTM has changed and stored data, it has to call FA service InstanceDataChanged (see 7.7.6.3) with information of changed data. The Frame Application will forward it to all other DTMs that have a reference to the same instance data and support synchronized locking by calling this DTM service OnInstanceDataChanged.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.2.13.4 Service OnChildInstanceDataChanged

In case of a FDT topology, it is necessary to inform the parent DTM about changed data.

Table 52 – Arguments for service OnInstanceChildDataChanged

	Request	Response
fdt:SystemTag	M	-

If a DTM has changed any data, it has to call FA service InstanceDataChanged (see 7.7.6.3) with information containing the instance specific changes. The Frame Application will forward this document by calling DTM service OnInstanceDataChanged (see 7.2.13.3) from all DTMs which have reference to the same instance data.

Concerning the according parent DTMs (primary parent, secondary parents (see service GetParentNodes, 7.7.3.5), the Frame Application shall implement the following behavior: the Frame Application shall send a notification to the according parent DTM via this DTM service OnChildInstanceDataChanged (see 7.2.13.4) within a multi user environment, this notification will only be sent to one parent DTM; this notification shall be sent to the parent DTM which has the lock concerning the related instance data.

If currently no parent DTM is started, the Frame Application shall start the parent DTM.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.2.14 DTM services related to import and export

7.2.14.1 Service Export

This service enables to build an export image of complete DTM persistent data.

Table 53 – Arguments for service Export

	Request	Response
fdt:StorageObject	-	M

This service enables to build an export image of complete DTM persistent data, independently whether it is stored in the Frame Application project storage or in a private DTM storage (see 4.8). The data returned by the service shall include DTM instance-related and non-instance related data.

This service is available in states:
[X] 'configured';
[_] 'communication allowed'.

7.2.14.2 Service Import

This service enables to import data earlier exported by service Export (see 7.2.14.1) back into the DTM.

Table 54 – Arguments for service Import

	Request	Response
fdt:StorageObject	M	-

This service enables to import data exported by service Export (see 7.2.14.1) back into the DTM.

This service is available in statesss:
[X] 'configured';
[_] 'communication allowed'.

7.3 Presentation object services

Interaction and state control between Frame Application, Presentation object and DTM is technology dependent and defined within an IEC 62453-4z document.

7.4 Channel object service

7.4.1 Channel object service introduction

This subclause defines the basic channel services which are common for Process and Communication Channels.

7.4.2 Service ReadChannelInformation

This service returns general (protocol-independent) channel information.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 55 – Arguments for service ReadChannelInformation

	Request	Response
fdt:Tag		M
fdt:ChannelPath		M
fdt:ChannelId		M
fdt:FrameApplicationTag		O

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.4.3 Service WriteChannelInformation

This service writes general (protocol-independent) channel information.

Table 56 – Arguments for service WriteChannelInformation

	Request	Response
fdt:ProtectedByChannelAssignment	M	.
fdt:FrameApplicationTag	O	-
fdt:serviceSucceeded		M

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.5 Process Channel object services

7.5.1 Services for IO related information

7.5.1.1 Service ReadChannelData

This service returns information with fieldbus dependent channel parameters.

Table 57 – Arguments for service ReadChannelData

	Request	Response
fdt:ProtocolId	M	
fieldbus:ChannelData		M

The service returns description with the channel specific parameters. The fieldbus is selected by the parameter ProtocolId.

The caller of service ReadChannelData should use the protocolId from the member NetworkInfo in the DtmDeviceInstanceTopology information available via the service GetActiveTypeInfo.

The returned parameters are especially used for channel assignment.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.5.1.2 Service WriteChannelData

This service sets changes of channel specific parameters.

Table 58 – Arguments for service WriteChannelData

	Request	Response
fdt:ProtocolId	M	
fieldbus:ChannelData	M	
fdt:serviceSucceeded		M

The service receives the channel specific parameters according to a fieldbus specific description. The fieldbus is defined by the parameter ProtocolId.

The service returns if the channel has accepted the complete information with all changes, else the channel has rejected all transferred changes. In this case, the channel informs the Frame Application about the error in detail via service OnErrorMessage (see 7.7.2.1)

The service works on the transient data of a DTM. That means that the new data are not stored implicitly.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.6 Communication Channel object services

7.6.1 Services related to communication

7.6.1.1 Service GetSupportedProtocols

This service returns information about the supported protocols of the Communication Channel.

Table 59 – Arguments for service GetSupportedProtocols

	Request	Response
fdt:BusCategory List	-	M

The service returns fieldbus protocol UUIDs. The supported protocols may be static or depend on the device configuration. If a channel supports more than one protocol during runtime, it has to support all protocols in parallel.

The service is not allowed to change the supported protocols if a DTM is linked to the channel because a change may cause an invalid topology (see 4.3). Which protocols are supported may be determined during topology planning. So if the Communication Channel can be configured to support different protocols, this is only allowed if there are no linked DTMs.

This service is available in states:
[_] ‘configured’;
[X] ‘communication allowed’.

7.6.1.2 Service Connect

This service establishes a new communication link to a device.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 60 – Arguments for service Connect

	Request	Response
fdt:CommunicationClientObjectReference	M	-
fdt:SystemTag	O	-
fdt:BusCategory	M	-
fieldbus:ConnectRequest	M	-
fieldbus:ConnectResponse	-	M
fdt:CommunicationReferenceID	-	M

The service takes information with fieldbus specific communication parameters. The fieldbus protocol to be used is identified by the parameter BusCategory.

The request contains additional fieldbus-protocol-specific information for the Communication Channel how to establish the connection. For example, information like repeat counts or preamble counts, etc. It is up to the environment to decide whether this information fits.

Furthermore, the request contains fieldbus- and protocol-specific information how to address the device connected to a specific fieldbus.

The SystemTag (see Table A.4) provided in the connect request is the SystemTag of the communication client. It may be used to get access to the DTM by calling service GetDtm (see 7.7.3.7) If the SystemTag is not provided then the communication client is not a DTM (may be the Frame Application or some other component).

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.6.1.3 Service Disconnect

This service releases a communication link to a device. For more than one connection to the same device, the link is identified by the communication reference returned by service Connect (see 7.6.1.2).

Table 61 – Arguments for service Disconnect

	Request	Response
fdt:CommunicationReferenceID	M	-
fieldbus:DisConnectRequest	M	-
fieldbus:DisConnectResponse	-	M

Via this service the caller of the service receives from the Communiction Channel the information whether the connection is released.

The service causes the release of all pending response data and at least the release of the CommunicationClientObjectReference passed by service Connect (see 7.6.1.2).

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.6.1.4 Service AbortRequest

This service aborts a communication link to a device without any response.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 62 – Arguments for service AbortRequest

	Request	Response
fdt:CommunicationReferenceID	M	-

The service terminates all pending requests and returns without waiting for a result. The termination of the connection will not be confirmed.

This service is available in states:
[_] ‘configured’;
[X] ‘communication allowed’.

7.6.1.5 Notification service AbortIndication

This service provides a notification that a connection (identified by the CommunicationReferenceID) has been aborted.

Table 63 – Arguments for service AbortIndication

	Request	Response
fdt:CommunicationReferenceID		M

The service returns the abort notification to a connected communication component or to a connected DTM, that a connection is terminated. All pending requests on this connection are also terminated.

The termination of the connection will not be confirmed.

A termination of a connection can be the result of an invoked service or can occur “spontaneously” (e.g. if the absence of a device is noted automatically by the underlying communication infrastructure).

This service is available in states:
[_] ‘configured’;
[X] ‘communication allowed’.

7.6.1.6 Service Transaction

The service performs exchange of a data structure with a device specified by the communication client.

Table 64 – Arguments for service Transaction

	Request	Response
fdt:CommunicationReferenceID	M	-
fieldbus:TransactionRequest	M	-
fieldbus:TransactionReponse	-	M

The service returns protocol specific communication responses for protocol specific communication requests.

Developers of Communication Channels should not expect that there will be only one pending request for a certain device at one time. For instance, several clients (e.g. Frame Application and DeviceDTM’s) are trying to retrieve information from the same device.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Even if the underlying fieldbus protocol allows sending only one request at a time to one or more devices, Communication Channels shall be able to manage a number of requests.

It is expected that the requests are processed by the Communication Channel in the order received if not specified otherwise by the protocol.

Developers of DeviceDTM's should consider that the used communication infrastructure creates delays in the communication. So the DeviceDTM's should limit the number of communication requests.

Also the DeviceDTM shall be able to handle a refused request, since there may be a variety of reasons to refuse a transaction request.

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.6.1.7 Service SequenceDefine

The service SequenceDefine defines a sequence of communication transactions and prepares them for execution. The communication transactions will be completed when the SequenceStart service is requested.

There is only one sequence defined from a DTM and all transactions requested by the previous sequences are terminated.

Table 65 – Arguments for service SequenceDefine

	Request	Response
fdt:CommunicationReferenceID	M	
fieldbus:SequenceDefine	M	-
fdt:serviceSucceeded		M

The Communication Channel has to preserve the defined sequence untill SequenceStart service is requested.

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.6.1.8 Service SequenceStart

SequenceStart service completes the execution of the sequence of communication transactions defined by SequenceDefine service (see 7.6.1.7).

Table 66 – Arguments for service SequenceStart

	Request	Response
fdt:CommunicationReferenceID	M	
fdt:serviceSucceeded		M

The sequence of communication transactions is completed in the sequence the transactions are defined without interruptions untill the last transaction service is executed. The Communication Channel informs the client for the result of each transaction when it is completed.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.6.2 Services related to sub-topology management

7.6.2.1 Service ValidateAddChild

The service validates whether a given DTM, specified by its SystemTag, can be added to the sub-topology (see 4.3).

Table 67 – Arguments for service ValidateAddChild

	Request	Response
fdt:SystemTag	M	-
fdt:serviceSucceeded	-	M

The channel validates the link. If the link is valid it will return success.

If the DTM needs more information about the child, it is able to access the DTM via service GetDtm (see 7.7.3.7) passing the received SystemTag.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.6.2.2 Service ChildAdded

The channel is informed that a new DTM was added to the sub-topology.

Table 68 – Arguments for service ChildAdded

	Request	Response
fdt:SystemTag	M	-

The channel is informed that a new DTM, specified by its SystemTag, has been added to the sub-topology.

If the channel needs more information about the child DTM, it is able to access the DTM via service GetDtm (see 7.7.3.7) passing the received SystemTag.

This service shall only be used in order to inform that the topology was changed. In case of reloading, for example project related data, this service shall not be called.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.6.2.3 Service ValidateRemoveChild

This service validates if a given child DTM, specified by its SystemTag, can be removed from the sub-topology.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 69 – Arguments for service ValidateRemoveChild

	Request	Response
fdt:SystemTag	M	-
fdt:serviceSucceeded	-	M

The channel has to validate if the DTM can be removed from the topology.

If the DTM needs more information about the child DTM, it is able to access the DTM via service GetDtm (see 7.7.3.7) passing the received SystemTag.

The validation shall include a check for active connection and return failure if a connection is active via this channel.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.6.2.4 Service ChildRemoved

With this service the channel is informed that a linked DTM was removed from the sub-topology.

Table 70 – Arguments for service ChildRemoved

	Request	Response
fdt:SystemTag	M	-

The channel is informed that a linked DTM, specified by its SystemTag, has been removed from the sub-topology.

If the channel needs more information about the child DTM, it is able to access the DTM via service GetDtm (see 7.7.3.7) passing the received SystemTag.

This service shall only be used in order to inform that the topology was changed. In case of destruction, for example project related data, this service shall not be called.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.6.2.5 Service SetChildrenAddresses

This service requests bus address setting for specified device list and returns device list including success information (see 6.1).

Table 71 – Arguments for service SetChildrenAddresses

	Request	Response
devList DeviceList	M	-
devList DeviceList	-	M

Requests setting of bus address for one device or a list of devices. The request may specify that the called Communication Channel shall open a user interface to request device address

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

settings from user. To get a qualified response, error information is included in the returned information.

As part of executing this service, the service OpenPresentationRequest (7.7.7.1) may be used to request address selection from the user.

The bus address on a DTM is set via the service NetworkManagementInfoWrite (see 7.2.11.2). This DTM shall not use this information for execution of communication transactions, which set the address in the field device.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.6.3 Services related to GUI and functions

7.6.3.1 Service GetChannelFuntions

This service returns information about functions of a DTM channel object.

Table 72 – Arguments for service GetChannelFunctions

	Request	Response
fdt:OperationPhase	M	-
func:Functions	-	M

This service provides information about functions provided by channel.

A channel shall inform the Frame Application about any function changes (e.g. number, status, label, etc.) by calling the service OnFunctionChanged (see 7.7.2.4).

Several responses may be provided. Usually this information is used by the Frame Application to create menus.

This service is available in states:
[X] ‘configured’;
[X] ‘communication allowed’.

7.6.3.2 Service GetGuiInformation

This service provides information how to start the user interface of a channel.

Table 73 – Arguments for service GetGuiInformation

	Request	Response
func:FDTFunctionCall	M	-
func:GUIInformation	-	M

Returns information that enables the Frame Application to instantiate the user interface. Channels shall only support Presentation objects that allows integration into the GUI of the Frame Application.

Integration and Interaction between these objects is technology dependent and defined within IEC 62453-4z documents.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service is available in states:
[X] 'configured';
[X] 'communication allowed'.

7.6.4 Services related to scan

7.6.4.1 Service Scan

This service performs the scan to the sub-topology (see 6.2.2).

Table 74 – Arguments for service Scan

	Request	Response
devList:BusAddressRange	O	-
fieldbus:ScanIdentifications	-	M

Returns information which contains a list of fieldbus related information to identify the connected devices. The optional passed BusAddressRange may limit the range of the scanning to specific device addresses or ranges.

This service is available in states:
[_] 'configured';
[X] 'communication allowed'.

7.7 Frame Application services

7.7.1 General state availability

All Frame Application services are available in DTM states:
[X] 'configured';
[X] 'communication allowed'.

7.7.2 FA services related to general events

7.7.2.1 Service OnErrorMessage

With this service a Frame Application receives notification by a DTM about errors during a service call.

Table 75 – Arguments for service OnErrorMessage

	Request	Response
fdt:SystemTag	M	-
fdt:ErrorMessage	M	-

The service is necessary if the DTM works without a user interface. If a DTM works without user interface it is not allowed to display any error message within its own dialog windows.

It is up to the Frame Application to handle the information. In case of errors or warnings, the human readable string may be displayed in a dialog box of the Frame Application.

In order to provide users with helpful information (e.g. regarding errors), it is recommended to use this service in cases, where a service call failed and service UserDialog (see 7.7.7.3) is not be used.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.7.2.2 Service OnProgress

With this service a Frame Application receives notification by a DTM about the progress on handling of a service call.

Table 76 – Arguments for service OnProgress

	Request	Response
fdt:SystemTag	M	-
fdt:ProgressInformation	M	-

This service shall be used by DTMs during active service calls, which take a longer time, to inform the Frame Application and at least the user about ongoing activities. If a DTM is not able to determine the real progress, it may be useful to change, for example the title.

It is up to the Frame Application how to handle the information.

7.7.2.3 Service OnOnlineStatusChanged

With this service a Frame Application receives notification by a DTM about status of communication link.

Table 77 – Arguments for service OnOnlineStatusChanged

	Request	Response
fdt:SystemTag	M	-
fdt:OnlineStatus (previous)	M	-
fdt:OnlineStatus (current)	M	
fdt:CommunicationError	O	

This service shall be used by DTMs to inform the Frame Application about changes of communication link status. The DTM shall specific new and previous online status and additional error information, if status change was triggered by a communication error.

7.7.2.4 Service OnFunctionsChanged

With this service a Frame Application receives notification by a DTM that the information about its current available additional functionality has changed.

Table 78 – Arguments for service OnFunctionsChanged

	Request	Response
fdt:ChannelPath	Ö	-
fdt:SystemTag	M	-

This service shall be used by DTMs to inform the Frame Application to update its menus or function calls which reference on this extended functionality. The Frame Application gets the currently available DTM functions by the service GetFunctions (see 7.2.5.1).

If the parameter ChannelPath is provided, then functionality of corresponding channels have changed. In this case, Frame Application gets the currently available channel functions by the service GetChannelFunctions (see 7.6.3.1).

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.7.3 FA services related to topology management

7.7.3.1 Service GetDtmInfoList

This service returns a list of DTM related information.

Table 79 – Arguments for service GetDtmInfoList

	Request	Response
dtmInfo:DTMInfo list	-	M
dtmInfo.BTMInfo list		M

Returns a list of DTM related information. This information enables a DTM to add another DTM to the FDT topology by usage of service CreateChild (see 7.7.3.2).

It is up to the Frame Application to decide which DTM information is returned in the list. For example, the list could contain only information concerning HART DTMs even if PROFIBUS DTMs are installed.

7.7.3.2 Service CreateChild

This service enables a DTM to add another DTM to the FDT topology.

Table 80 – Arguments for service CreateChild (DTM)

	Request	Response
fdt:DtmDeviceType	M	-
fdt: ChannelObjectReference	M	-
fdt:SystemTag	-	M

This service enables a DTM to add another DTM identified by its DTM device type to the sub-topology identified by the ChannelObjectReference.

The DTM is instantiated by the Frame Application. The service returns the SystemTag of the DTM If the operation failed, no SystemTag shall be returned. The Frame Application has to implement the behavior described in the sequence diagram in 8.1.2..

If service is called to add a BTM, then BTM block type is used in the request as defined in following table (Table 81).

Table 81 – Arguments for service CreateChild (BTM)

	Request	Response
btm:BtmBlockType	M	-
fdt: ChannelObjectReference	M	-
fdt:SystemTag	-	M

7.7.3.3 Service DeleteChild

This service enables a DTM to remove another DTM from the FDT topology.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 82 – Arguments for service DeleteChild

	Request	Response
fdt:SystemTag	M	-
fdt: ChannelObjectReference	M	-
fdt:serviceSucceeded	-	M

This service enables a DTM to remove another DTM specified by SystemTag from the sub-topology identified by the ChannelObjectReference.

The Frame Application has to call the service ValidateRemoveChild (see 7.6.2.3) at the Communication Channel from which the DTM is removed. If this was the last reference within the topology, the Frame Application has to delete the instance data. The Frame Application has also to call service ClearInstanceData (see 7.2.4.5) with respect to the behavior. The operation shall also fail, if a sub-topology exists.

7.7.3.4 Service MoveChild

This service enables a DTM to move another DTM in the FDT topology.

Table 83 – Arguments for service MoveChild

	Request	Response
fdt:SystemTag	M	-
fdt:SystemTag	M	
fdt: ChannelObjectReference	M	-
fdt:serviceSucceeded	-	M

This service enables a DTM to move another DTM specified by its SystemTag to another sub-topology identified by the destination DTM SystemTag and ChannelObjectReference.

The Frame Application has to call ValidateAddChild and ValidateRemoveChild as defined for the services CreateChild (see 7.7.3.2) and DeleteChild (see 7.7.3.3).

7.7.3.5 Service GetParentNodes

This service enables a DTM to request a list of DTMs which are directly above in the FDT topology.

Table 84 – Arguments for service GetParentNodes

	Request	Response
fdt:SystemTag	M	-
fdt:SystemTag List	-	M
fdt:SystemTag		M

The DTM for which the list shall be created is identified by the SystemTag in the request. The service returns a SytemTag list of all DTMs above in the FDT topology and a single SystemTag identifiying the primary parent DTM.

7.7.3.6 Service GetChildNodes

This service enables a DTM to request a list of DTMs which are directly below in the FDT topology.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 85 – Arguments for service GetChildNodes

	Request	Response
fdt:SystemTag	M	-
fdt: ChannelObjectReference	M	-
fdt:SystemTag List	-	M

The DTM and Communication Channel for which the list shall be created are identified by the SystemTag and a ChannelObjectReference. The service returns a SytemTag list of all DTMs below in the FDT topology.

7.7.3.7 Service GetDtm

This service enables a DTM to request a reference to another DTM identifiyed by its SystemTag.

Table 86 – Arguments for service GetDtm

	Request	Response
fdt:SystemTag	M	-
fdt: DtmObjectReference	-	M

This service enables a DTM to request a reference to another DTM identifiyed by its SystemTag. Additional calls within a FrameApplication instance shall return the identical DTM ObjectReference.

The DTM which has requested the references has to call the service ReleaseDtm (see 7.7.3.8) to release the reference. The Frame Application has to implement a kind of reference counting which is required to handle multiple references to the same DTM instance.

7.7.3.8 Service ReleaseDtm

This service shall be used to release reference to the associated DTM that was requested by the service GetDtm (see 7.7.3.7)

Table 87 – Arguments for service ReleaseDtm

	Request	Response
fdt:SystemTag	M	-
fdt:serviceSucceeded	-	M

This service shall be used to release the reference to the associated DTM, identified by its SystemTag, which was requested by the service GetDtm (see 7.7.3.7).

7.7.4 FA services related to redundancy

7.7.4.1 Service OnAddedRedundantChild

A parent DTM shall send this information to the Frame Application if another DTM handling a redundant device is added to the topology. The Frame Application is then able to display the instance at an additional redundant Communication Channel.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 88 – Arguments for service OnAddedRedundantChild

	Request	Response
fdt:SystemTag	M	-
fdt:ChannelObjectReference	M	-
fdt:serviceSucceeded	-	M

The parent DTM has added the DTM identified by its SystemTag, handling a redundant slave, to the Communication Channel identified by ChannelObjectReference. The Frame Application shall not call service ValidateAddChild (see 7.6.2.1) or ChildAdded (see 7.6.2.12) on this channel.

7.7.4.2 Service OnRemovedRedundantChild

A parent DTM shall send this information to the Frame Application if another DTM handling a redundant device is removed from the topology. The Frame Application is able to hide the instance at the additional redundant Communication Channel.

Table 89 – Arguments for service OnRemovedRedundantChild

	Request	Response
fdt:SystemTag	M	-
fdt:ChannelObjectReference	M	-
fdt:serviceSucceeded	-	M

The parent DTM removes the DTM identified by its SystemTag, handling a redundant slave, from the Communication Channel identified by ChannelObjectReference. The Frame Application shall not call service ValidateRemoveChild (see 7.6.2.3) or service OnRemoveChild (see 7.6.2.4) services on this channel.

7.7.5 FA services related to storage of DTM data

7.7.5.1 Service SaveInstanceData

This service enables a DTM to save its instance data into the project storage managed by the Frame Application (see 4.8).

Table 90 – Arguments for service SaveInstanceData

	Request	Response
fdt:StorageObject	M	-
fdt:serviceSucceeded	-	M

This service enables a DTM to save its instance data into the project storage managed by the Frame Application. It is up to the Frame Application to decide whether data passed in the requests is immediately stored in the project storage or cached in the memory until another event occurs (e.g. explicit user request).

The DTM calling this service shall hold the lock to the instance data (see service LockInstanceData, 7.7.6.1), otherwise the service call fails (returns fdtServiceSucceeded = FALSE).

7.7.5.2 Service LoadInstanceData

This service enables a DTM to loads instance data from the project storage managed by the Frame Application (see 4.8).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table 91 – Arguments for service LoadInstanceData

	Request	Response
fdt:StorageObject	-	M

The DTM calling this service gets exactly the same data which was stored by service SaveInstanceData (see 7.7.5.1).

7.7.5.3 Service GetPrivateDtmStorageInformation

This service enables a DTM to request information about its private data storage from the Frame Application (see 4.8).

Table 92 – Arguments for service GetPrivateDtmStorageInformation

	Request	Response
InstanceRelatedStorageInfo	-	M-
ProjectRelatedStorageInfo	-	M

The response parameter InstanceRelatedStorageInfo contains the information which enables a DTM to access the storage that belongs to the calling instance. The response parameter ProjectRelatedStorageInfo contains the information which enables to access the storage shared by all instances of this DTM type in a project. The private DTM storage mechanism is implementation technology dependent and thus defined in the IEC 62453-4z document defining mapping to specific technologies.

If a Frame Application is not able to provide access to private DTM data storages then it returns no information. A DTM shall be able to work without any side effects in this case. It shall ensure that there is no impact to the instance data managed by SaveInstanceData (see 7.7.5.1) and LoadInstanceData (see 7.7.5.2).

A DTM shall clean up the instance related storage if service ClearInstanceData (see 7.2.4.5) is called. If the DTM holds any references between project and instance related storages, then it also shall clean up these in the project related storage.

7.7.6 FA services related to DTM data synchronization

7.7.6.1 LockInstanceData

This service enables a DTM to request an exclusive write access to its instance data.

Table 93 – Arguments for service LockInstanceData

	Request	Response
fdt:serviceSucceeded	-	M

The Frame Application validates the request and grants the access within the multi-user environment. In single user systems, the exclusive access is always granted. The DTM shall make modifications to instance related data only when access is granted via this service. See 8.5

The DTM shall request the lock for instance data only when required for data modification purpose. The DTM shall possess an exclusive write access when allowing:

- the configuration of instance data via presentation objects;

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

- the synchronization of the data between device and instance data via service ReadDataFromDevice (see 7.2.12.4);
- the device data modification via service DeviceDataWrite (see 7.2.10.3), or via parameterization GUI, to update information data modified in device (ModifiedInDevice).

If the write access is not granted, the DTM shall not make modifications into instance data and shall inform user via service OnErrorMessage (see 7.7.2.1) notification service.

7.7.6.2 UnlockInstanceData

This service enables the DTM to notify the Frame Application that it wants to unlock the write access to instance data.

Table 94 – Arguments for service UnlockInstanceData

	Request	Response
fdt:serviceSucceeded	-	M

When instance data is not further under modification, the DTM shall immediately save it and call this service. Within the multi-user environment the Frame Application shall notify other DTM instances having a reference to the same instance data about the changed data via service InstanceDataChanged (see 7.7.6.3).

If the service response is unsuccessful, the DTM shall inform the user via service OnErrorMessage (see 7.7.2.1) notification service to clean up the system database. Within the single user systems, the service response is always successful.

7.7.6.3 InstanceDataChanged

This service enables a DTM to inform the Frame Application about data changes

Table 95 – Arguments for service OnInstanceDataChanged

	Request	Response
Information	M	-

If a DTM has changed and saved data, it has to call this service with information of changed data. The Frame Application shall forward this information to all DTM instances that have a reference to the same instance data by calling DTM service OnInstanceDataChanged (see 7.2.13.3).

7.7.7 FA services related to presentation

7.7.7.1 Service OpenPresentationRequest

This service enables a DTM or a channel to request a presentation object in the user interface of the Frame Application.

Table 96 – Arguments for service OpenPresentationRequest

	Request	Response
func: FDTFunctionCall	M	-
ui: RunAsModal	O	
fdt:ChannelPath	O	
fdt:invokeID	-	M
fdt: ServiceSucceeded	-	M

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

This service enables a DTM or a channel to open a presentation, identified by a FDTFunctionCall, in the user interface of the Frame Application. The Frame Application shall use the DTM service GetGuiInformation (see 7.2.5.3) or channel service GetGuiInformation (see 7.6.3.2) to obtain more information about the presentation.

If the parameter RunAsModal is set to TRUE then this service behaves modal: The Frame Application at least has to ensure that all presentations of calling DTM are disabled, so that no further user input is possible.

If the parameter ChannelPath is provided then presentation of corresponding channels shall be opened.

The service returns an InvokeID that enables to close the opened presentation by calling the service ClosePresentationRequest.

7.7.7.2 Service ClosePresentationRequest

This service enables a DTM or a channel to close an opened presentation object.

Table 97 – Arguments for service ClosePresentationRequest

	Request	Response
fdt:invokeID	M	
fdt: ServiceSucceeded	-	M

This service enables a DTM or a channel to close an opened presentation. The passed parameter InvokeID identifies the presentation that was for example opened by service OpenPresentationRequest (see 7.7.7.1).

7.7.7.3 Service UserDialog

This service enables a DTM to open a user message within the user interface of the Frame Application.

Table 98 – Arguments for service UserDialog

	Request	Response
ui:UserMessage	M	-
ui:UserMessage	-	M

A DTM shall use this service for standard user messages like error or information messages. Especially if a DTM is not allowed to open a private user dialog (see service PrivatDialogEnabled, 7.2.1.1).

This service returns the selection of the user action or the specified default answer of the message. It is up to the Frame Application to display the user message or to send the default answer.

In case of a distributed system the Frame Application shall ensure displaying the user dialog and the user interface of the DTM at the same workplace.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

7.7.8 FA Services related to audit trail

7.7.8.1 Service RecordAuditTrailEvent

This service shall be used by all DTMs to send their audit trail information to the Frame Application. It is up to the Frame Application to implement the audit-trail-application itself which records the device specific information and supplies the user interface.

Table 99 – Arguments for service RecordAuditTrailEvent

	Request	Response
fdt:SystemTag	M	-
audittrail:LogEntry	M	-
audittrail:TransactionInfo	O	-

Notification by a DTM about changed data to be recorded by the Frame Application's audit trail tool.

The content of LogEntry depends on the DTM. The implementation of the audit trail tool is Frame Application-specific.

audittrail:TransactionInfo with value startTransaction shall be used to request audit trail for the following actions (e.g. configuration or simulation). All calls of this service will be recorded until the record is closed by calling it with endTransaction as value of audittrail:TransactionInfo parameter.

When the record has been closed, the Frame Application may use it for its own specific audit trail functions like adding comments, etc.

8 FDT dynamic behavior

8.1 Generate FDT topology

8.1.1 FDT topology generation triggered by the Frame Application

The Frame Application is responsible to generate and manage the topology. The sequence (Figure 33) shows how the Frame Application manages the link between DTMs and Communication-Channels.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

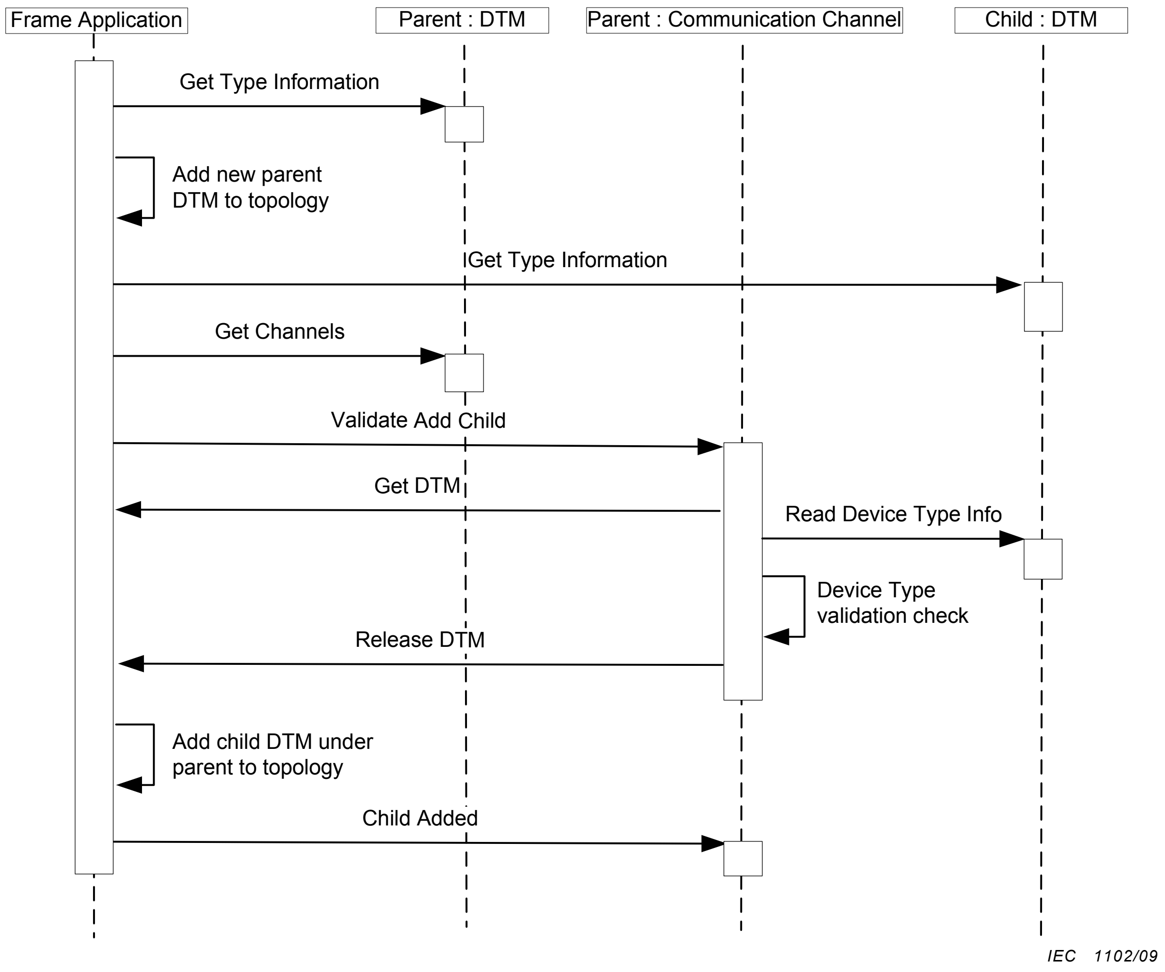


Figure 33 – FDT topology generation triggered by the Frame Applications

8.1.2 FDT topology generation triggered by the DTM

This sequence (Figure 34) shows the generation of the FDT topology triggered by a DTM.

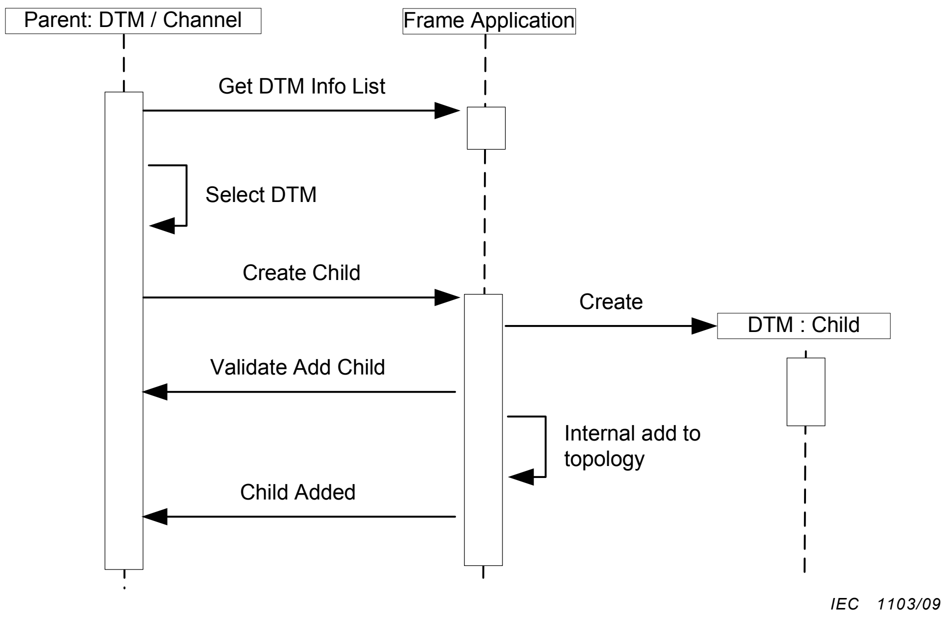


Figure 34 – FDT topology generation triggered by a DTM

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

8.2 Address setting

8.2.1 Address setting introduction

This subclause describes different address setting scenarios (see also 6.1).

8.2.2 Set or modify device address – with user interface

In this scenario, the Frame Application requests a set of specific child device addresses at DTMs. This sequence (see Figure 35) is for example started if new DTM is added to the topology. Similar sequence can be used if a Frame Application offers manual change of address in context of a DTM.

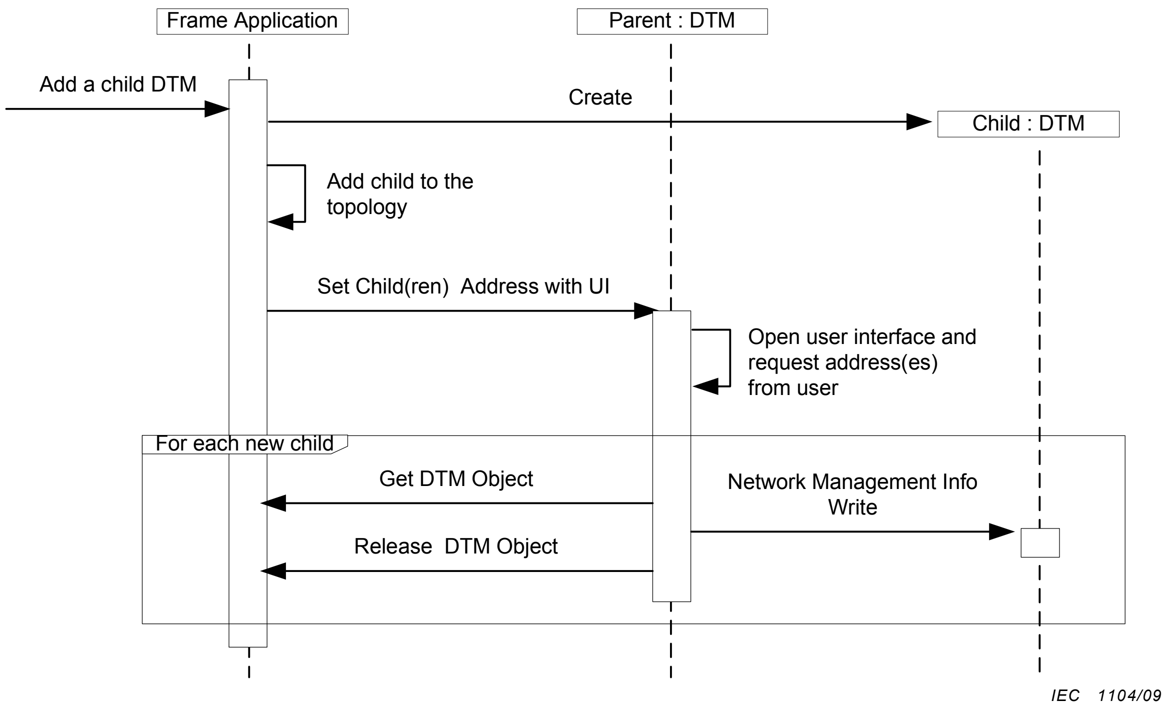

Frame Application
Parent : DTM
Add a child DTM
Create
Child : DTM
Add child to the topology
Set Child(ren) Address with UI
Open user interface and request address(es) from user
For each new child
Get DTM Object
Network Management Info Write
Release DTM Object
IEC 1104/09

Figure 35 – Set or modify device address – with user interface

8.2.3 Set or modify device address – without user interface

In this scenario, the Frame Application requests to set device addresses at DTMs, for example after a scanning or import operation (see Figure 36).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

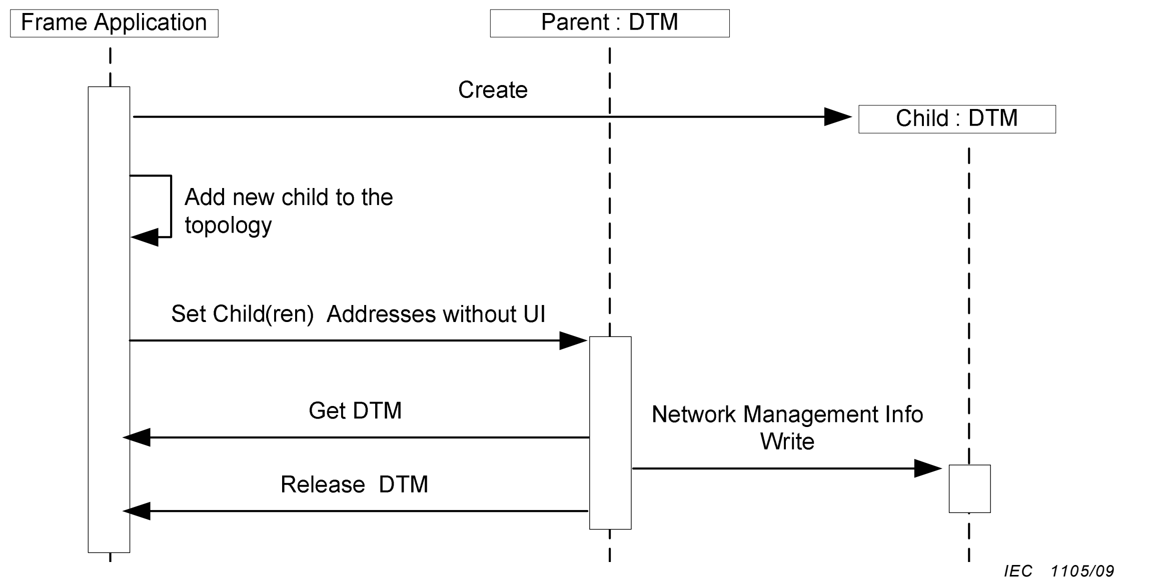

Frame Application
Parent : DTM
Create
Child : DTM
Add new child to the topology
Set Child(ren) Addresses without UI
Get DTM
Network Management Info Write
Release DTM
IEC 1105/09

Figure 36 – Set or modify device address – with user interface

The Frame Application submits the list of addresses to the parent DTM by the service NetworkManagementInfoWrite (see 7.2.11.2) .

8.2.4 Display or modify all child device addresses with user interface

In this scenario (see Figure 37) Frame Application requests display or modification of all child device addresses at a DTM. This sequence is for example started if user selects corresponding menu in context of a communication or gateway-DTM.

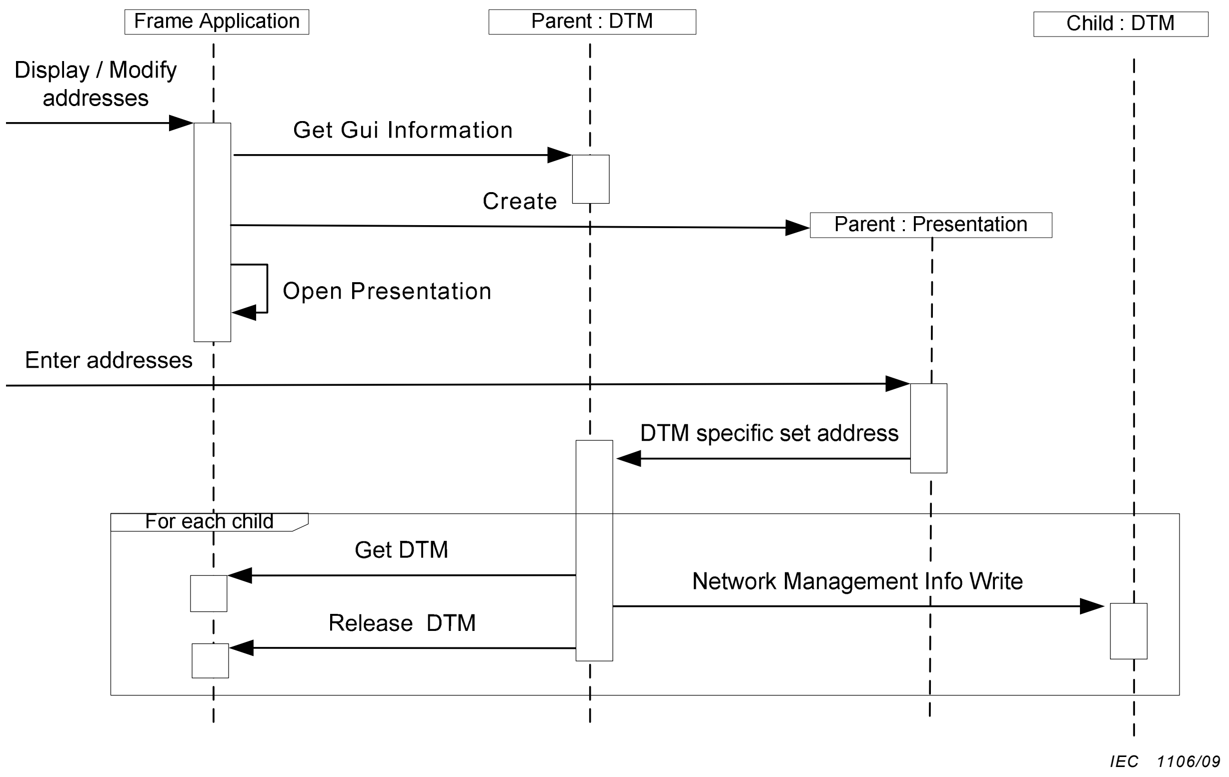


Figure 37 – Set or modify all device addresses – with user interface

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

8.3 Communication

8.3.1 Communication overview

This subclause describes different communication scenarios of a DTM with its device. .

8.3.2 Peer to peer communication

This sequence diagram (Figure 38) shows the peer to peer communication of a DTM by using the services provided by a linked Communication Channel.

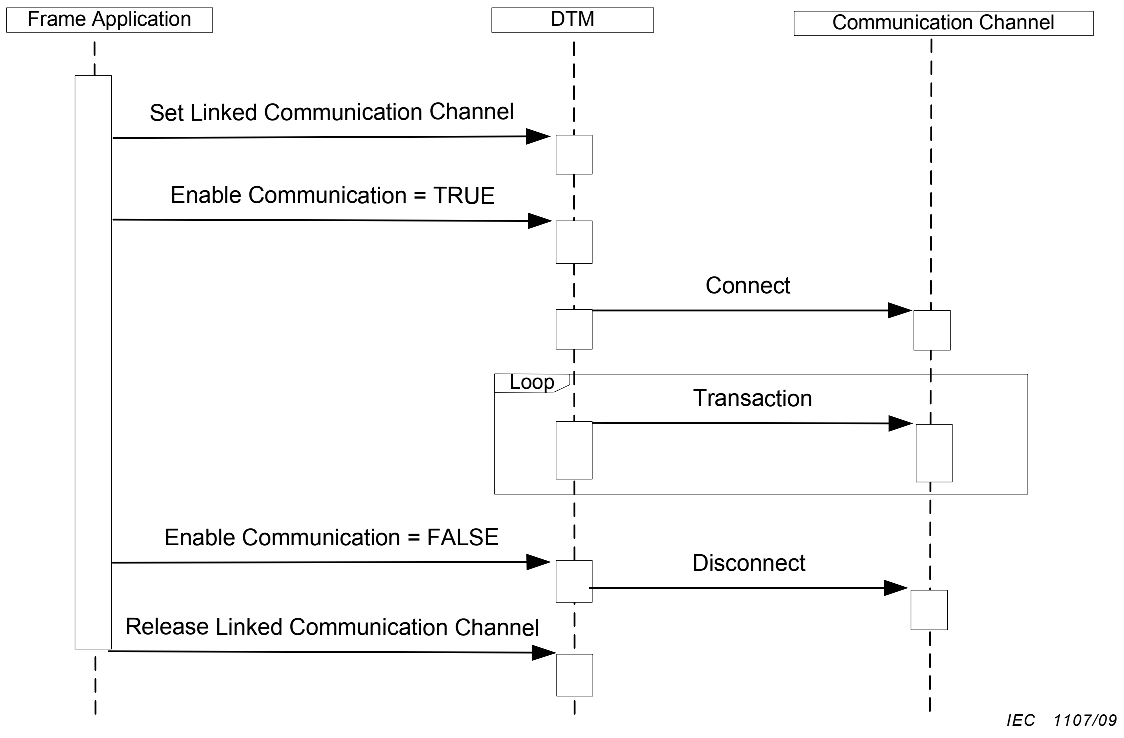
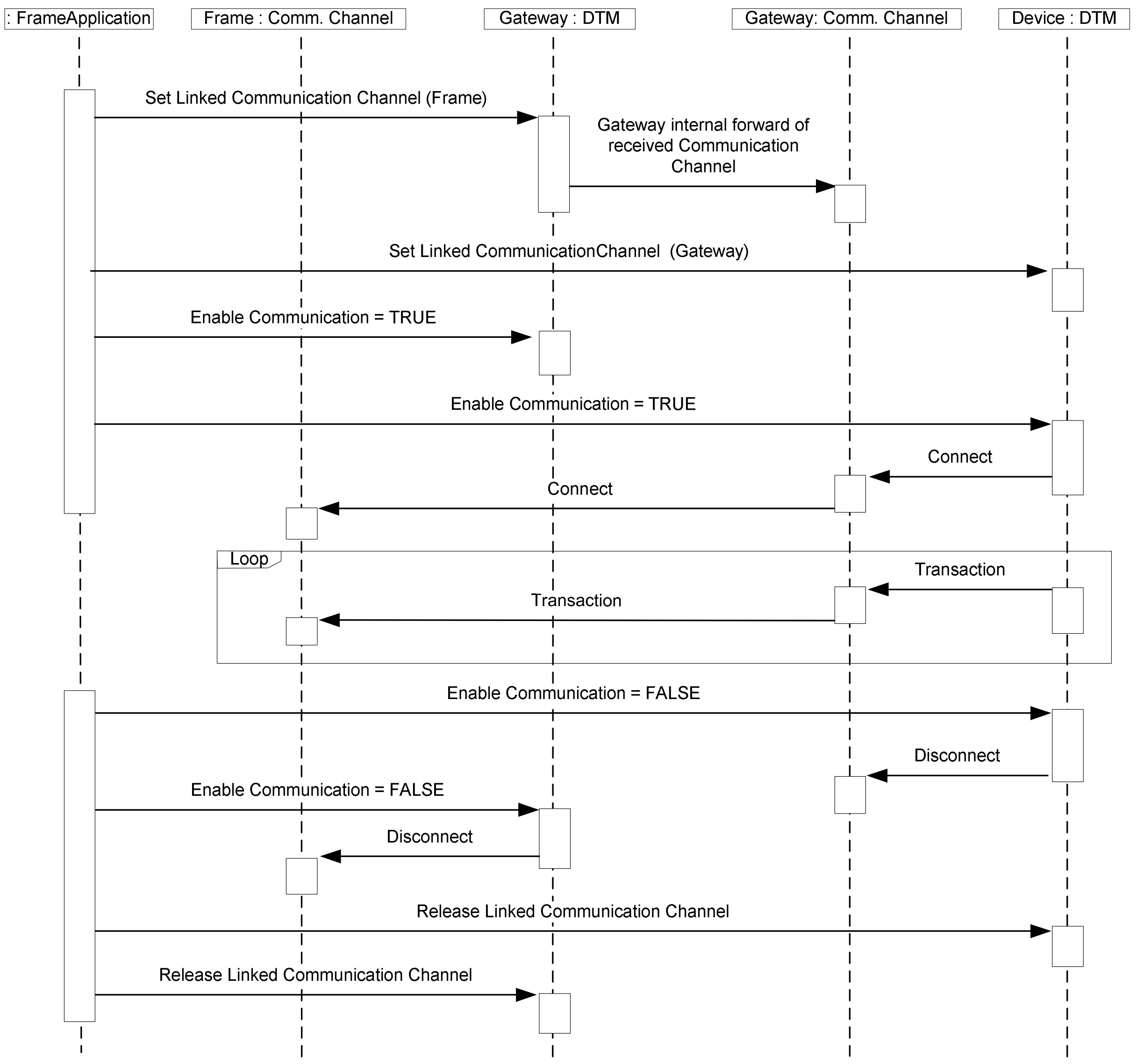

Frame Application
DTM
Communication Channel
Set Linked Communication Channel
Enable Communication = TRUE
Connect
Loop
Transaction
Enable Communication = FALSE
Disconnect
Release Linked Communication Channel
IEC 1107/09

Figure 38 – Peer to peer communication

8.3.3 Nested communication

This sequence diagram (Figure 39) shows the communication of a DTM through a gateway and a Communication Channel provided by the Frame Application.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.


: FrameApplication
Frame : Comm. Channel
Gateway : DTM
Gateway: Comm. Channel
Device : DTM
Set Linked Communication Channel (Frame)
Gateway internal forward of received Communication Channel
Set Linked CommunicationChannel (Gateway)
Enable Communication = TRUE
Enable Communication = TRUE
Connect
Connect
Loop
Transaction
Transaction
Enable Communication = FALSE
Disconnect
Enable Communication = FALSE
Disconnect
Release Linked Communication Channel
Release Linked Communication Channel

IEC 1108/09

Figure 39 – Nested communication

8.3.4 Device initiated data transfer

Some protocols support data transfer services that are initiated by the device and not by the DTM. In order to support such services, following general behavior is defined in FDT:

- the infrastructure (filter, service queue) for such services is initiated by a protocol-specific transaction request of the DTM, dependant on the protocol this service may be acknowledged or not;
- the device initiated data transfer is transported by multiple transaction responses to a single initiating transaction request;
- the infrastructure for these services is terminated by a protocol-specific transaction request of the DTM.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

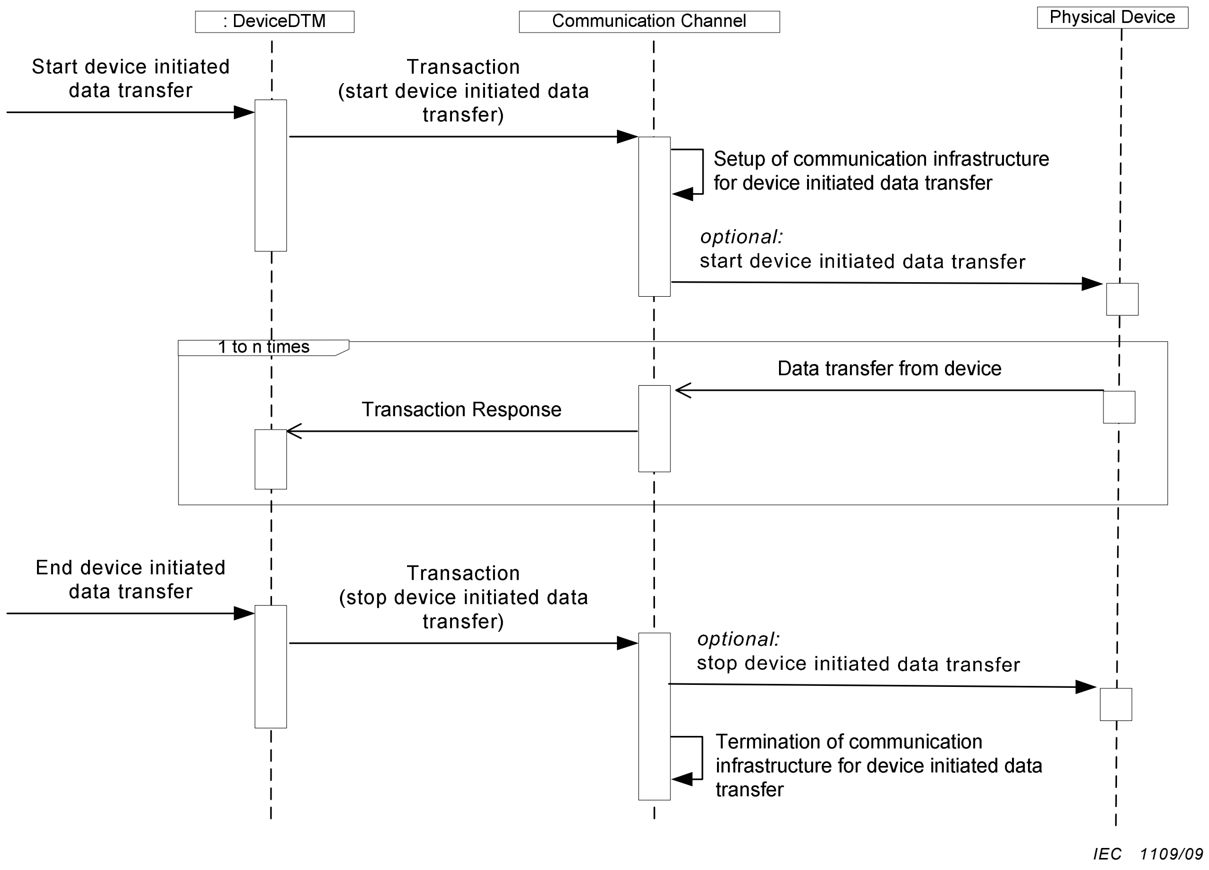

: DeviceDTM
Communication Channel
Physical Device
Start device initiated data transfer
Transaction (start device initiated data transfer)
Setup of communication infrastructure for device initiated data transfer
optional: start device initiated data transfer
1 to n times
Data transfer from device
Transaction Response
End device initiated data transfer
Transaction (stop device initiated data transfer)
optional: stop device initiated data transfer
Termination of communication infrastructure for device initiated data transfer
IEC 1109/09

Figure 40 – Device initiated data transfer

Device initiated data transfer needs management by the communication infrastructure. That is why it is necessary to unambiguously identify the request to initialize and terminate this type of behavior.

The transaction service (see 7.6.1.6) to initiate or terminate device initiated data transfer is transport protocol specific. The transaction service to terminate the data transfer contains all the information necessary to terminate the device initiated data transfer (if necessary it also references the transaction service which started the data transfer).

If a DTM starts device initiated data transfer service on the device (see Figure 40), it shall also terminate this service. The termination has to occur, before the DTM goes offline (see service Disconnect (7.6.1.3) or AbortRequest (7.6.1.4)). If the connection is terminated by the notification AbortIndication (see 7.6.1.5) this service is terminated automatically.

The support of such services is protocol-specific and device-specific. If a Communication Channel is not able to support this type of service, it shall return an error indicating that it is not able to support this feature.

The concrete protocol specific transactions are defined in the IEC 62453-3xy publications defining the protocol profile integration in FDT.

8.4 Scanning and DTM assignment

This sequence (Figure 41) shows how a Frame Application generates a FDT sub-topology based on information returned by the service Scan (see 7.6.4.1).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

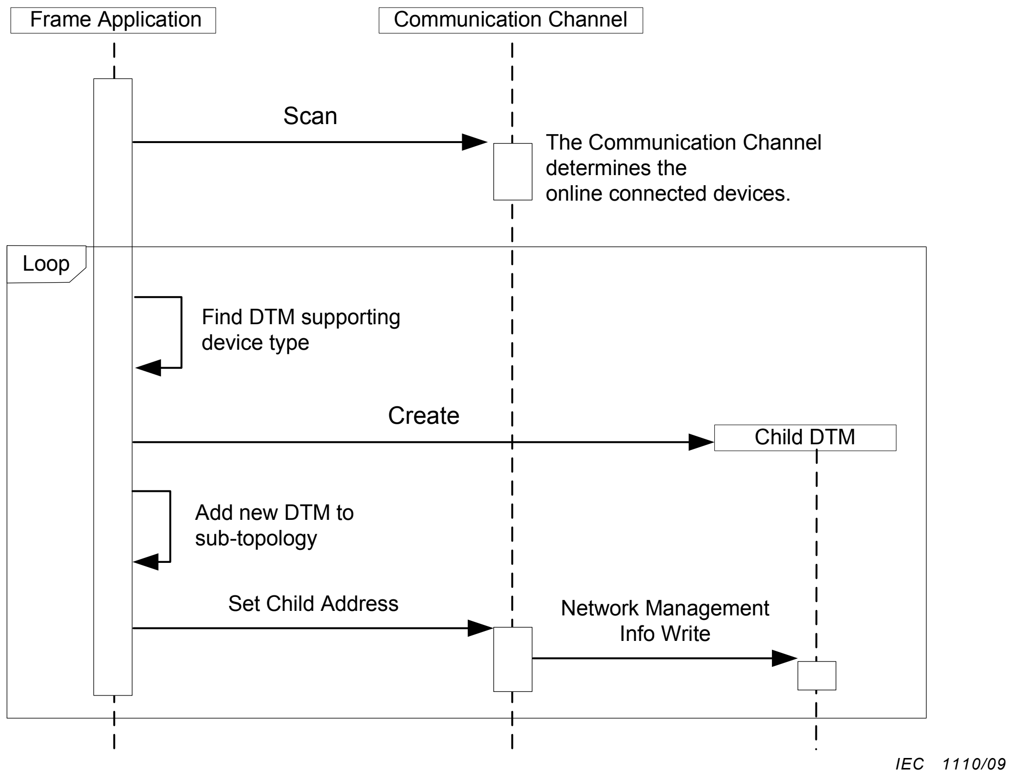

Frame Application
Communication Channel
Scan
The Communication Channel determines the online connected devices.
Loop
Find DTM supporting device type
Create
Child DTM
Add new DTM to sub-topology
Set Child Address
Network Management Info Write
IEC 1110/09

Figure 41 – Scanning and DTM assignment

The Frame Application is able to request device type and instance related information by calling the service Scan (see 7.6.4.1), use the information to find proper DTMs that can be added to the FDT topology (see 6.2.3 DTM assignment), and finally set the addresses in the DTMs to enable them to communicate with its device (see 6.1 Address Management).

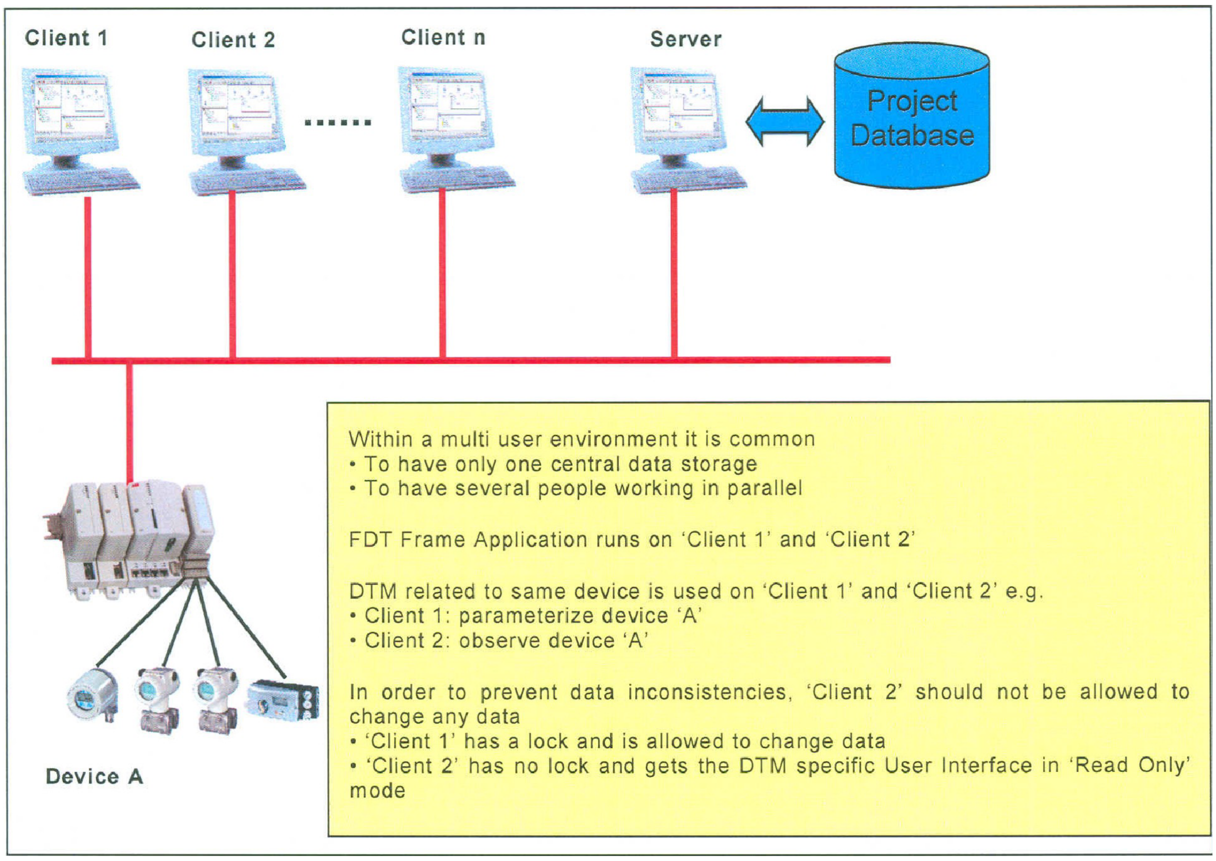
8.5 Multi-user scenarios

8.5.1 General

In order to reduce time, for example the commissioning time of a large installation, systems are often designed for multi-user access. The goal is to allow a number of users to work in parallel and ensure data consistency, while preventing unintentional change of data, for example configuration related data based on parallel work. Figure 42 gives an overview of a multi-user system.

The mechanism to handle changes within the device (for example local operation on the device) is described in 8.7.4.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.


Client 1
Client 2
Client n
Server
Project Database
Device A
Within a multi user environment it is common
• To have only one central data storage
• To have several people working in parallel
FDT Frame Application runs on 'Client 1' and 'Client 2'
DTM related to same device is used on 'Client 1' and 'Client 2' e.g.
• Client 1: parameterize device 'A'
• Client 2: observe device 'A'
In order to prevent data inconsistencies, 'Client 2' should not be allowed to change any data
• 'Client 1' has a lock and is allowed to change data
• 'Client 2' has no lock and gets the DTM specific User Interface in 'Read Only' mode
IEC 1111/09

NOTE This figure shows one Frame Application managing multiple users. (E.g. multiple instance of one Frame Application running on several clients working on the same data base.) Multiple Frame Applications for multiple users is out of scope of this specification.

Figure 42 – Multi-user system

Within a multi-user environment it is common, that more than one DTM have access to the instance data. To synchronize DTMs which are started by several users on different workplaces FDT provides a locking mechanism. Target for this event mechanism is, that only one DTM has read/write access to instance related data. All other DTMs have only a read access.

Access to the stored data set shall be managed during all modifications of the data set. This includes modification via the GUI of the DTM and modification via the InstanceData services of the DTM.

In order to prevent multi-DTM-access to the device, online writable access to a device shall be provided only if the instance data set is locked. This means the data set shall be locked only if services are executed that change device data (e.g. Online Parameterization GUI and service WriteDataToDevice) but not when device data is read only (e.g observation).

For this reason, a DTM shall lock its data set only if required and only during modification of the data. After the instance data is saved by the Frame Application and is not further under modification the DTM shall unlock its instance data immediately to enable concurrent access to the device data by other DTM instances within a multi-user environment.

Before opening GUI for modification of data, the DTM shall try to lock the instance data. If service is successful all modification to instance data is allowed, for example parameterization, synchronization. If lock is not granted, the DTM is not allowed to make any alterations to its instance data.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

When there is no need to modify the data (e.g. all GUIs with write access are closed, synchronization is completed), the DTM shall request to save the instance data and unlock it when saved.

DTM is not allowed to hold write access to its instance related data unnecessarily. Otherwise, DTM is not suitable for use in a multi-user environment. For example, DTM should not lock instance data immediately when it is initialized and unlock it only when terminated.

8.5.2 Synchronized and non-synchronized locking mechanism for DTMs

The DTM shall always support a locking mechanism. There are two options for it: synchronized and non-synchronized mechanism. The supported mechanism type is informed by Frame Application in a response of service OnUnlockInstanceData (see 7.2.13.2).

The following sequence diagram (Figure 43) shows the general flow of events in case of a DTM supporting the synchronized locking mechanism.

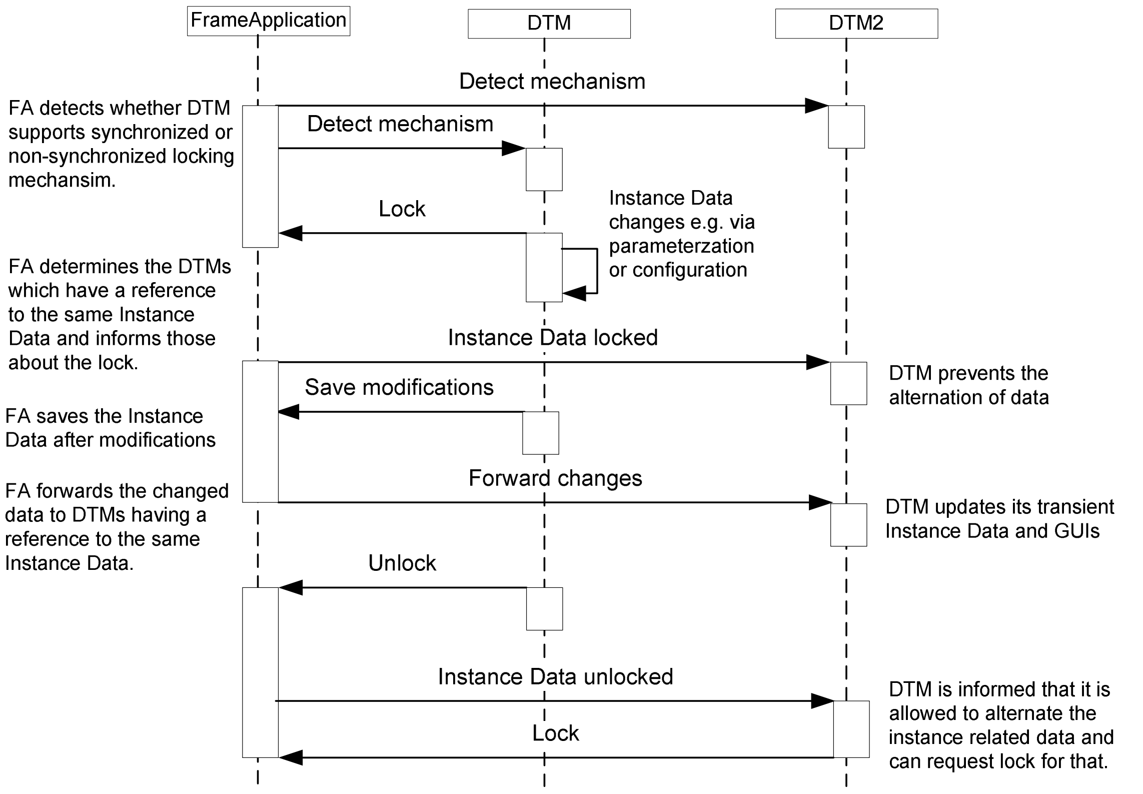

FrameApplication
DTM
DTM2
Detect mechanism
FA detects whether DTM supports synchronized or non-synchronized locking mechansim.
Detect mechanism
Lock
Instance Data changes e.g. via parameterzation or configuration
FA determines the DTMs which have a reference to the same Instance Data and informs those about the lock.
Instance Data locked
DTM prevents the alternation of data
Save modifications
FA saves the Instance Data after modifications
Forward changes
FA forwards the changed data to DTMs having a reference to the same Instance Data.
DTM updates its transient Instance Data and GUIs
Unlock
Instance Data unlocked
Lock
DTM is informed that it is allowed to alternate the instance related data and can request lock for that.
IEC 1112/09

Figure 43 – General synchronized locking mechanism

The following sequence diagram (Figure 44) shows the general flow of events in case of a DTM supporting the non-synchronized locking mechanism.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

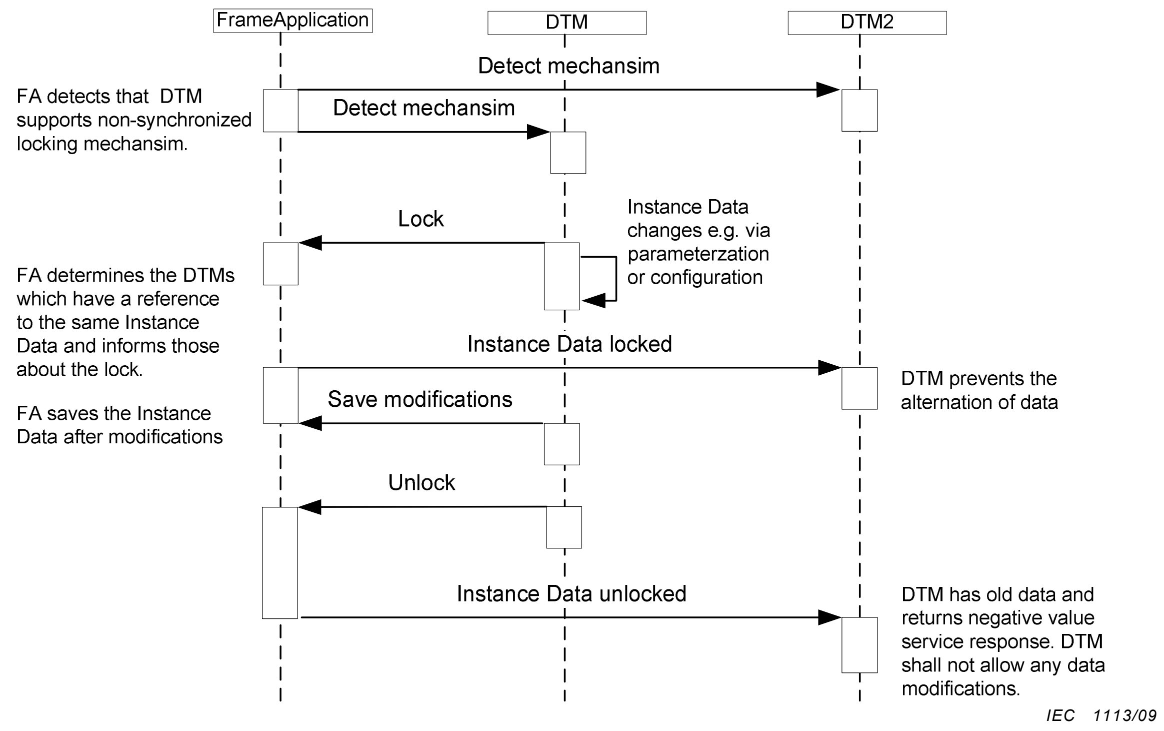

FrameApplication
DTM
DTM2
Detect mechansim
FA detects that DTM supports non-synchronized locking mechansim.
Detect mechansim
Lock
Instance Data changes e.g. via parameterzation or configuration
FA determines the DTMs which have a reference to the same Instance Data and informs those about the lock.
Instance Data locked
DTM prevents the alternation of data
Save modifications
FA saves the Instance Data after modifications
Unlock
Instance Data unlocked
DTM has old data and returns negative value service response. DTM shall not allow any data modifications.
IEC 1113/09

Figure 44 – General non-synchronized locking mechanism

The following sequence diagram (Figure 45) shows a parameterization scenario in case of a DTM supporting the synchronized locking mechanism.

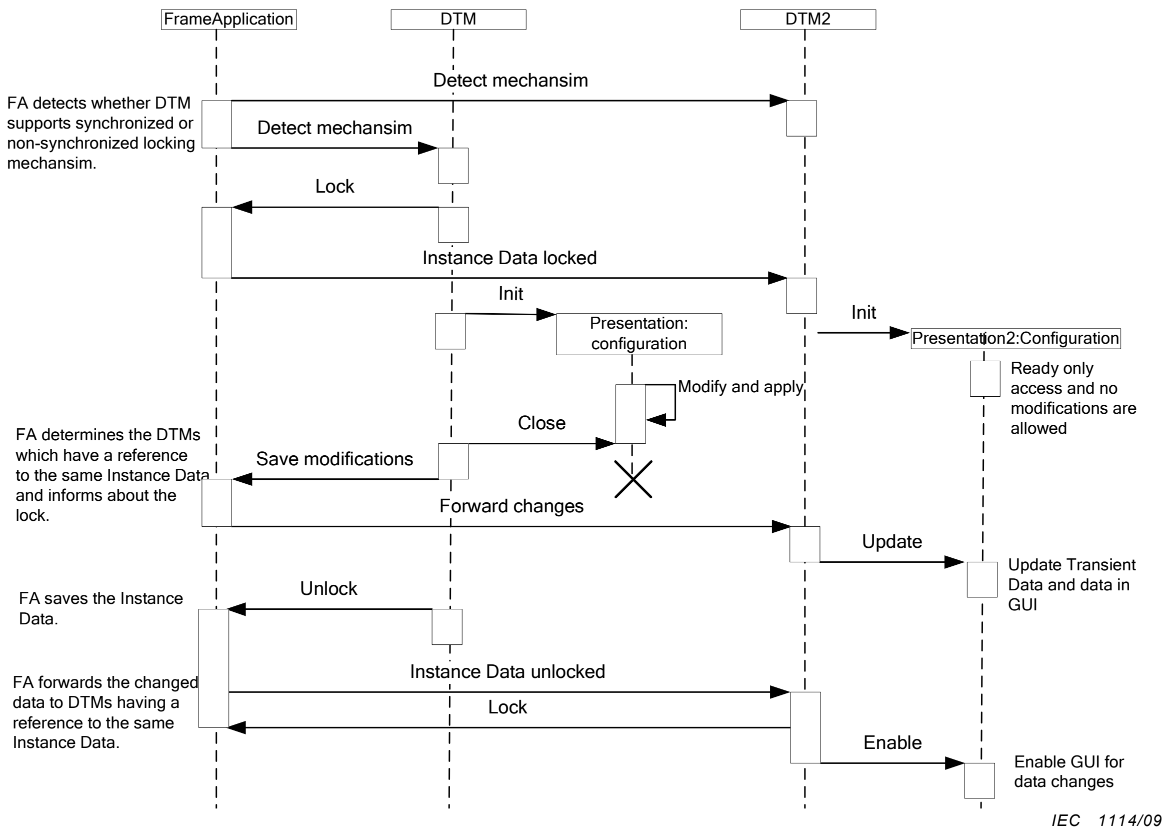

FrameApplication
DTM
DTM2
Detect mechansim
FA detects whether DTM supports synchronized or non-synchronized locking mechansim.
Detect mechansim
Lock
Instance Data locked
Init
Presentation: configuration
Init
Presentation2:Configuration
Ready only access and no modifications are allowed
Modify and apply
Close
FA determines the DTMs which have a reference to the same Instance Data and informs about the lock.
Save modifications
Forward changes
Update
Update Transient Data and data in GUI
FA saves the Instance Data.
Unlock
Instance Data unlocked
FA forwards the changed data to DTMs having a reference to the same Instance Data.
Lock
Enable
Enable GUI for data changes
IEC 1114/09

Figure 45 – Parameterization in case of synchronized locking mechanism

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

8.5.3 Additional rules

Additional rules are:

- before opening a Presentation Object containing changeable parameters, the DTM shall try to lock the data set. If successful, all user input fields may be enabled. If unsuccessful, the user input fields shall be disabled. After closing all Presentation Objects in the case of a locked data set, the DTM should ask to store the data set and unlock it after Frame Application has stored the data set (after the Frame Application has called storage services). The data set shall not be locked if the Presentation Object does not contain changeable parameters, for example for applications like observe;
- before loading data from the device the DTM shall also try to lock the dataset, after loading the data from the device, the DTM should request the FA to store its data and unlock dataset after FA has stored the dataset. If the data set could not be locked, the DTM shall inform the user by an error message. In that case, the action will not be performed;
- in order to support multi-user environments locks shall be kept as short as possible. This means, for example if a DTM locks its data set after instantiation and unlocks the data set only if it is terminated, this DTM is not suitable for use within a multi-user environment. A DTM shall lock its data set only if necessary and only during modification of the data. After the instance data is stored by the frame-application and is not modified further, the DTM shall unlock its data set immediately. This enables concurrent access to the device data by other DTM instances within a multi-user environment;
- when the DTM receives a notification regarding an unlock of the data set, a Presentation Object is active, which would allow to change the data set and if the access right of the current user allows changing the instance data set, the DTM shall ask the user if he wants to have write access. When the user wants to have write access, the DTM has to lock the data set. The DTM shall enable the input controls only if the DTM gets the lock.

8.6 Notification of changes

The FDT defines a notification mechanism to inform other DTM instances for the same device about changes in the data set and to refresh those (synchronized DTMs). If the synchronization mechanism is not implemented (non-synchronized DTMs), changing the data set of a DTM on one PC can block other users on a different PC, because the DTM on the other PC needs to shutdown to reload the changed data. Shutting down the DTM on the other PC and loading the changed data set from project data base will take more time and will require more computing resources than an update by the notification mechanism.

Therefore, it is recommended that DTMs implement notification mechanism for synchronized DTMs (see 8.5.2).

If a DTM has stored any changed data (after the Frame has called storage service), the DTM shall call the service InstanceDataChanged. The DTM shall not call InstanceDataChanged before storing the instance data set.

If a DTM supports the notification mechanism for synchronized DTMs and when an event is received regarding changed parameters (service OnInstanceDataChanged) the DTM shall apply the new parameter values into the instance data set without requesting a lock. In this case, the storage state of the data set equals to persistent. Also all input controls in all opened graphical user interfaces shall be updated.

8.7 DTM instance data state machines

8.7.1 Instance data set introduction

This subclause describes the different states of DTM instance data with regard to modifications and persistence. The persistence mechanism itself is described in 4.8.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

8.7.2 Modifications state machine

This state machine (see Figure 46) reflects the possible states of instance data concerning modifications.

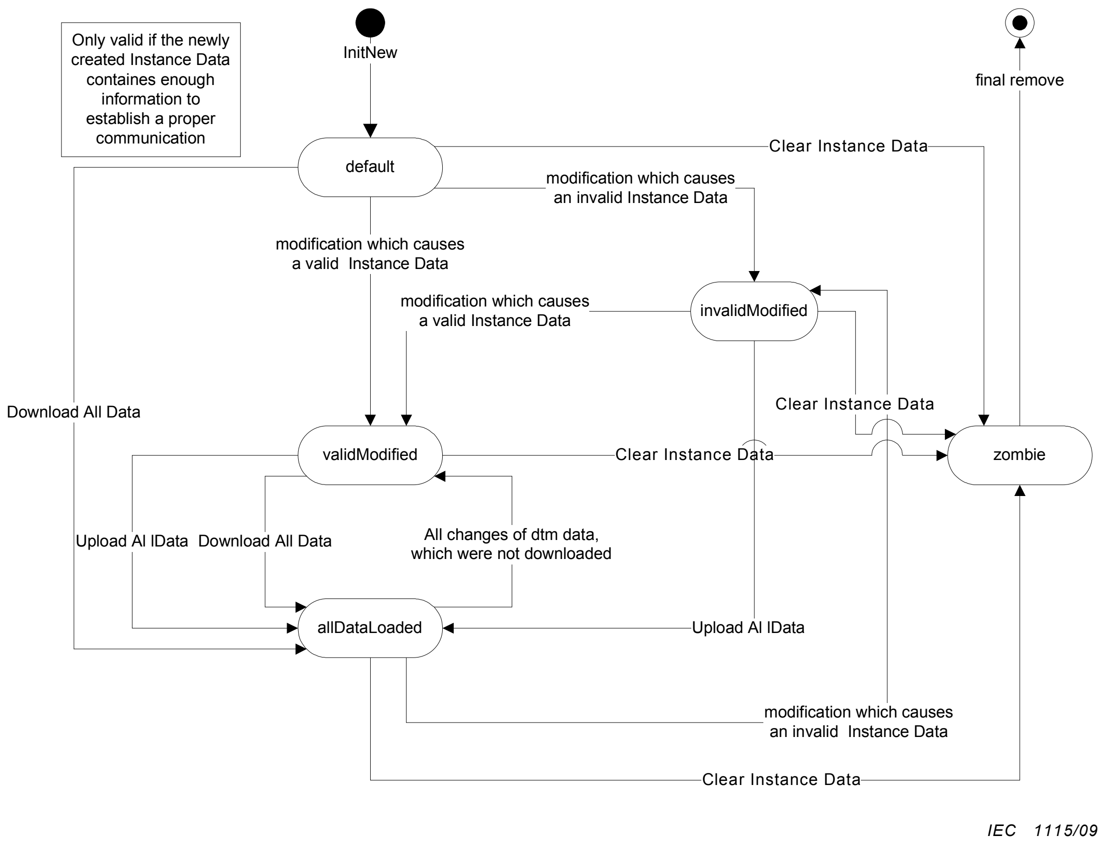

Only valid if the newly created Instance Data containes enough information to establish a proper communication
InitNew
final remove
default
Clear Instance Data
modification which causes an invalid Instance Data
modification which causes a valid Instance Data
invalidModified
modification which causes a valid Instance Data
Download All Data
Clear Instance Data
validModified
Clear Instance Data
zombie
Upload Al lData
Download All Data
All changes of dtm data, which were not downloaded
allDataLoaded
Upload Al lData
modification which causes an invalid Instance Data
Clear Instance Data
IEC 1115/09

Figure 46 – Modifications state machine of instance data

The states of the modifications state machine are explained in Table 100.

Table 100 – Modifications state machine of instance data

State	Meaning
Default	The contents of the instance data are the default data. This state will typically appear after creation of new instance data
validModified	The instance data was modified in a consistent manner
invalidModified	The data set was modified. The data are not in a consistent state
allDataLoaded	The data was loaded from or into the related device. NOTE This state means not, that the data within the device are equal to the data found within the related instance data. Due to the fact that a user can use tools out of the scope of FDT, FDT- cannot guarantee such a state
zombie	The instance data is prepared to delete, no access to this data is allowed

The data set state can be used to verify which devices are in sync with their DTMs and which devices need to be reloaded. For example, ‘allDataLoaded’ indicates that a device is in sync with the DTM and shall not be loaded again. This state shall be changed only if data is really changed and not, for example if a lock is requested or a Presentation Object is active without changing the data set, because reloading a device is a time consuming action.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

8.7.3 Persistence state machine

This state machine reflects the possible states of instance data concerning the persistence of data (see Figure 47).

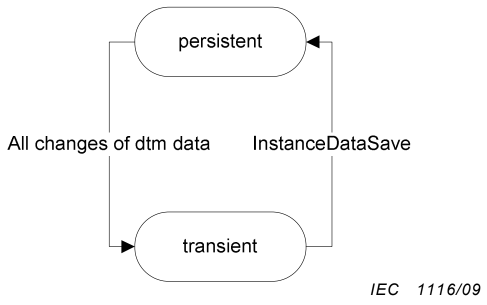


Figure 47 – Persistence state machine of instance data

The states of the persistence state machine are explained in Table 101.

Table 101 – Persistence state machine of instance data

State	Meaning
persistent	The content of the instance related data is persistent. This state will typically appear after the data was saved by the DTM using the persistence service SaveInstanceData (see 7.7.5.1) of the Frame Application.
transient	The instance data was modified. These changes are not saved in a persistent way. All modified data within the DTM is temporary and not synchronized with other DTM instances or with the Frame Application.

A DTM shall request to store the instance data set at minimum if one of the following conditions occurs:

- if the last Presentation Object, which is able to change parameter values has been terminated;
- after parameters have been read from or written to the device (services ReadDataFromDevice / WriteDataToDevice);
- after parameter values are changed by another component (InstanceDataWrite) and no Presentation Object which is able to change parameter values is active.

8.7.4 Modification in device

It is useful to keep the device data set and the DTM instance data set in sync. When invoked in online mode, the DTM should ask the user whether the data in the DTM and in the device should be synchronized. This behavior is mandatory for Presentation Objects with applicationId “fdtOfflineParameterize” and “fdtOnlineParameterize”.

The mandatory flag <fdt:modifiedInDevice> will present the synchronization state.

8.7.4.1 Presentation fdtOnlineParameterize

After the user closes the DTM user interface, the DTM shall ask the user if the instance data set and the current device configuration should be synchronized. This may overwrite already existing parameter modifications within the instance data set (status not equal to ‘allDataLoaded’, refer to 8.7.3). If this is the case, the DTM shall include this information in the message mentioned above.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

If the user confirms the synchronization, the DTM shall load the complete device data set into the instance data set. If the DTM can guarantee consistency of device data and instance data set, it could upload only the modified device parameters under consideration of device specific business rules.

8.7.4.2 Presentation fdtOfflineParameterize

If the user closes the DTM user interface and if the DTM can connect to the device, (meaning the DTM is in a state of 'communication allowed'), the DTM shall ask the user if a complete write should be initiated to synchronize the instance data set and the current device configuration.

If the user confirms the download, the DTM shall download the complete instance data set into the device. If the DTM can guarantee consistency of device data and instance data set, it could write only the modified parameters under consideration of device specific business rules.

8.7.5 Storage life cycle

The following table (Table 102) describes the states of the instance data set during a 'life' cycle of a DTM. The states of instance data set are defined by dataSetState (see Figure 46) and storabeState (see Figure 47).

Table 102 – Example life cycle of a DTM

Scenario	storageState	dataSetState	modifiedInDevice	Remarks
DTM is initialized	transient	default	False	
After DTM is initialized and storing the data	persistent	default	False	
After failed read of device data	persistent	default	False	no change of the states, because due to the unsuccessful read the data set is not changed
After successful read device data	transient	allDataLoaded	False	
After reading from device and storing the data set	persistent	allDataLoaded	False	
After modification of device data set via e.g. the GUI Online Parameterize	transient, because flag <modifiedInDevice> has changed	validModified	True	The device accepted new data, therefore a read from device is necessary in order to synchronize the data sets
After storing the data set	persistent	validModified	True	
After read data from device	transient, because dataSetState is changed	allDataLoaded	False	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Scenario	storageState	dataSetState	modifiedInDevice	Remarks
After modification of instance data set	transient	If changes comply to business rules of device: validModified Otherwise: invalidModified	False	
After storing the data set	persistent	validModified	False	
After release of the DTM	-/-	-/-		
After DTM started with persistent data	persistent	validModified	False	

8.8 Parent component handling redundant slave

A parent component, for example a Communication DTM, handling a redundant fieldbus is able to detect a Device DTM able to handle a redundant slave within the execution of service ChildAdded by examining the information returned by the service NetworkManagementInfoRead of the child DTM. The Communication DTM selects the redundant communication paths (either automatically or by means of a dialog) and sets redundancy information to the Device DTM by calling NetworkManagementInfoWrite.

The Device DTM provides this redundancy information within the call to service Connect to the Communication Channel. For each such opened connection the Communication Channel is able to use one of the available communication paths without need for further interaction with the Device DTM.

A Frame Application implementing services related to redundancy (see 7.7.4) is able to manage topology information about redundant DTMs. In such a Frame Application, the topology view may show this redundancy information. On the other hand Communication DTM and Device DTM of a redundant slave can be used in a Frame Application without knowledge about redundancy, because all redundancy functionality is handled by the Communication DTM and its child DTMs.

The sequence related to management of a redundant topology is shown in Figure 48.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

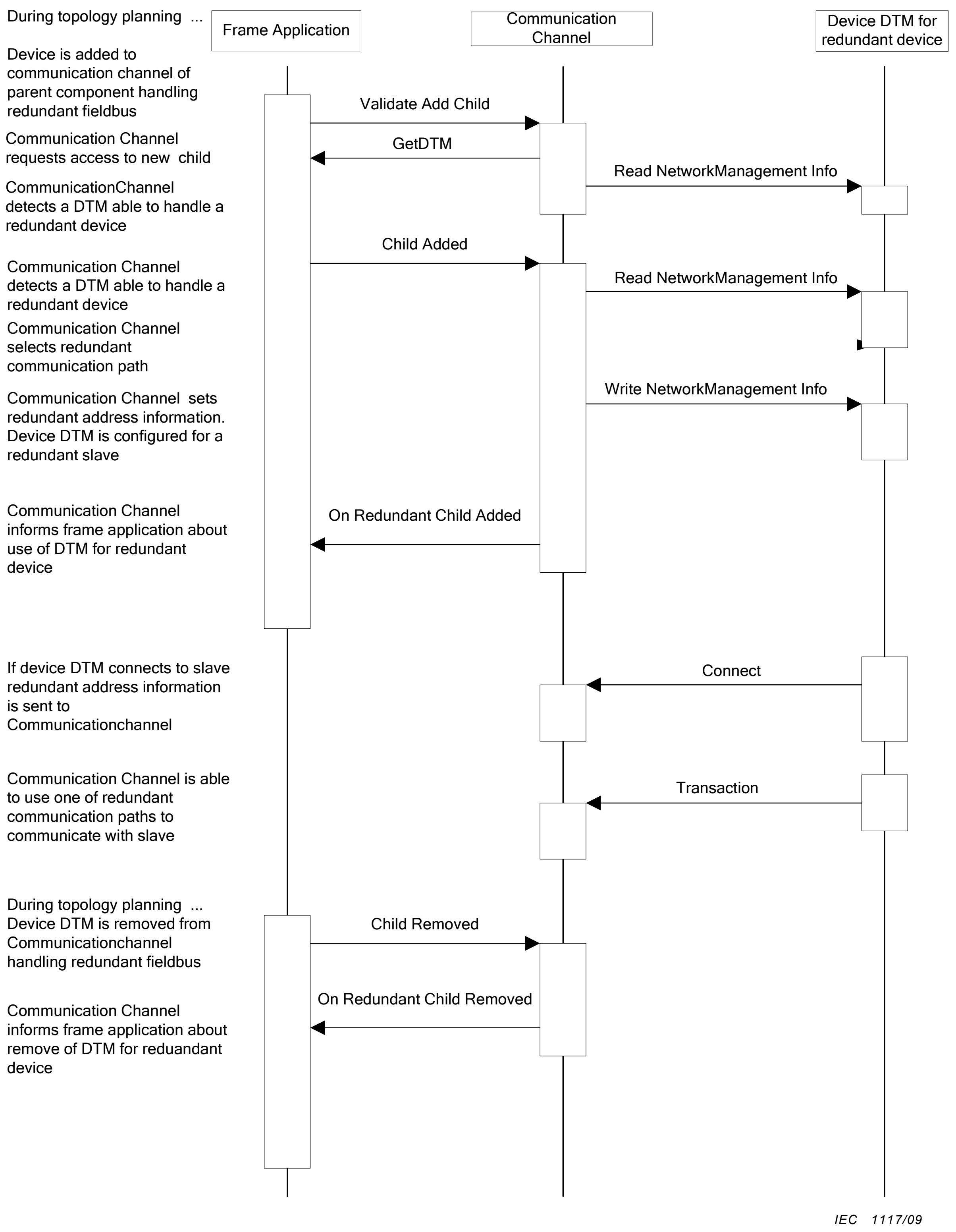

During topology planning ...
Frame Application
Communication Channel
Device DTM for redundant device
Device is added to communication channel of parent component handling redundant fieldbus
Validate Add Child
Communication Channel requests access to new child
GetDTM
Read NetworkManagement Info
CommunicationChannel detects a DTM able to handle a redundant device
Child Added
Communication Channel detects a DTM able to handle a redundant device
Read NetworkManagement Info
Communication Channel selects redundant communication path
Write NetworkManagement Info
Communication Channel sets redundant address information. Device DTM is configured for a redundant slave
Communication Channel informs frame application about use of DTM for redundant device
On Redundant Child Added
If device DTM connects to slave redundant address information is sent to Communicationchannel
Connect
Communication Channel is able to use one of redundant communication paths to communicate with slave
Transaction
During topology planning ... Device DTM is removed from Communicationchannel handling redundant fieldbus
Child Removed
Communication Channel informs frame application about remove of DTM for reduandant device
On Redundant Child Removed
IEC 1117/09

Figure 48 – Management of redundant topology

8.9 DTM upgrade

8.9.1 General rules

The first step to be resolved during upgrade is to verify if the saved data set can be associated with the new DTM.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Each DTM should expose a unique identifier (dataSetId) specifying its data set format. A DTM can provide a list of compatible data set formats it can load. These dataSetIds are returned as part of GetTypeInformation service.

If the ClassIdentifier of the DTM is not changed, it is up to the DTM to handle version upgrade. The list of supported data set formats may not be considered by the Frame Application when data is loading in this case. Additional check for data compatibility must be done by the DTM when the data is loaded.

8.9.2 Saving data from a DTM to be upgraded

The first step is to store the information from the DTM that is for upgrade.

The Frame Application needs to store additional information about the DTM:

- ClassIdentifier of a DTM;
- dataSetId of the stored data format,
- storage type;
- additional information about the device.

This information is associated to the stored data and can be used later by the Frame Application to identify and manage data set.

The following diagram provides an illustration of that sequence (Figure 49):

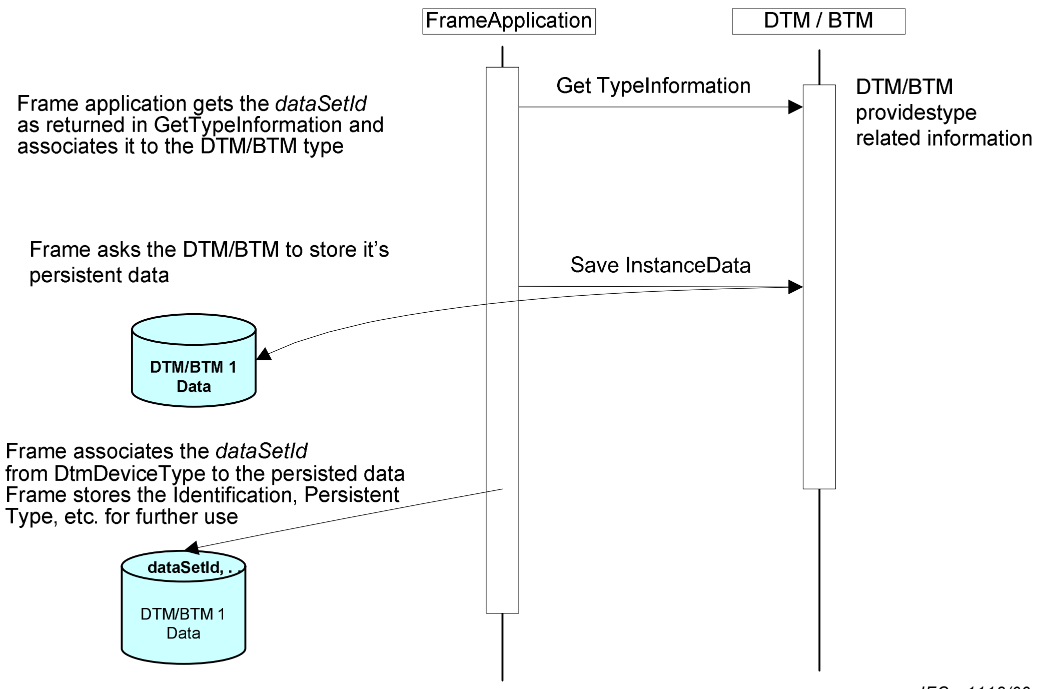

FrameApplication
DTM / BTM
Get TypeInformation
DTM/BTM providestype related information
Frame application gets the *dataSetId* as returned in GetTypeInformation and associates it to the DTM/BTM type
Frame asks the DTM/BTM to store it's persistent data
Save InstanceData
DTM/BTM 1 Data
Frame associates the *dataSetId* from DtmDeviceType to the persisted data Frame stores the Identification, Persistent Type, etc. for further use
dataSetId, ..
DTM/BTM 1 Data
IEC 1118/09

Figure 49 – Associating data to a dataSetId

8.9.3 Loading data in the replacement DTM

Since the stored data is associated to the storage information, it can be used later by the Frame Application to select a different DTM which is able to handle the stored data.

The following diagram provides an illustration of that sequence (Figure 50).

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

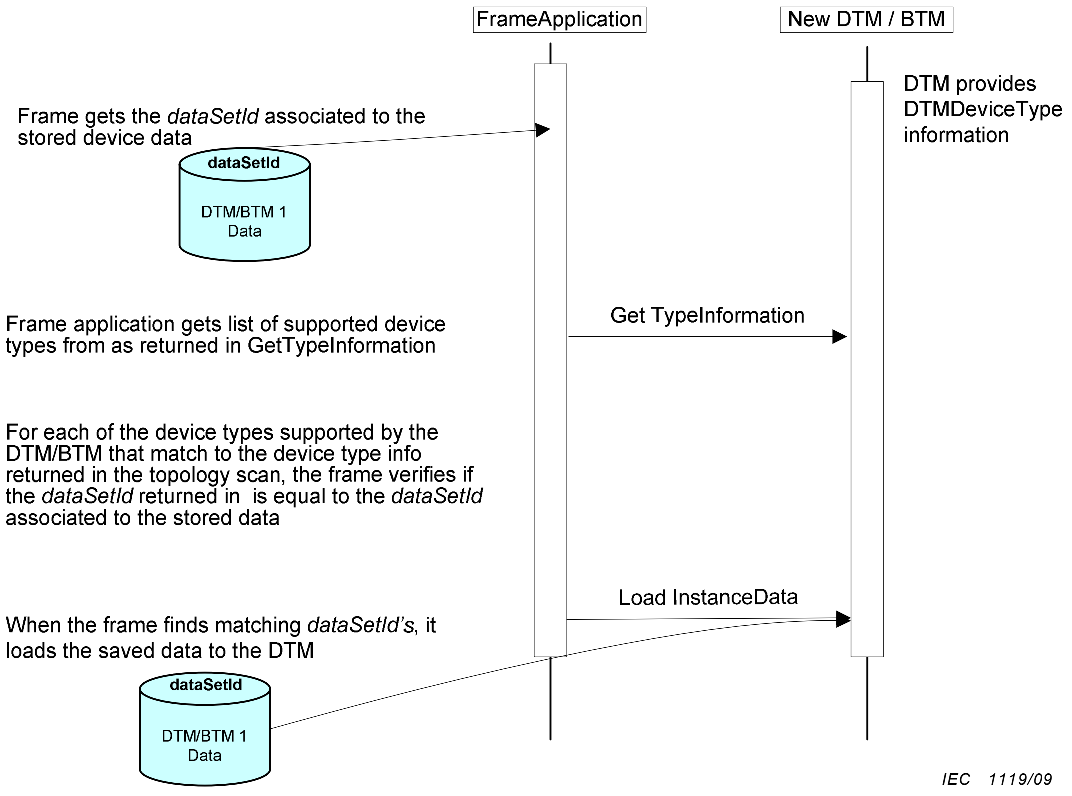

FrameApplication
New DTM / BTM
DTM provides DTMDeviceType information
Frame gets the *dataSetId* associated to the stored device data
dataSetId
DTM/BTM 1 Data
Frame application gets list of supported device types from as returned in GetTypeInformation
Get TypeInformation
For each of the device types supported by the DTM/BTM that match to the device type info returned in the topology scan, the frame verifies if the *dataSetId* returned in is equal to the *dataSetId* associated to the stored data
Load InstanceData
When the frame finds matching *dataSetId's*, it loads the saved data to the DTM
dataSetId
DTM/BTM 1 Data
IEC 1119/09

Figure 50 – Loading data for a supported dataSetId

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Annex A
(normative)

FDT data types definition

A.1 General

This clause defines all data types used for FDT. The data types are grouped according the functionality. Data types belonging together from a logical point of view are listed within one table. For each of these groups, a namespace is defined. The namespace is identified by a prefix. When a datatype is referenced outside of the scope of its namespace, the prefix is added in front of the data type reference.

The names for data types can be composed from different words and are written in „Camel Case“, where an upper case letter indicates the beginning of a new word. Names of simple data types start with a lower case letter. Names of structured data types start with an upper case letter.

EXAMPLES

Simple data type ‘address’ defined in namespace ‘fdt’:	fdt:address
Simple data type ‘semanticId” defined in namespace ‘fdt’:	fdt:semanticId
Structured data type “SemanticInformation” defined in namespace ‘fdt’:	fdt: SemanticInformation

For each data type the basic data type is defined as well as a description.

The name and the semantics of the data types as defined in the columns “Data type” and “Description” in the data type tables of this annex are normative. The column “Definition” provided in the data type tables contains informative content.

For structured data types, the column “Definition” consists of “Elementary data types”, “Usage” and “Multiplicity”. “Elementary data types” describes the structure of the data type and the data type of the sub-elements.

“Usage” describes how a sub-element is used within the structured data type.

Possible values for usage are:

O - optional (this data element mus not be provided);

M - mandatory (this data element shall be provided);

C - conditional (it depends on a condition, whether this data element is provided, the condition shall be described in the right column);

S - selective (this element and an other element exclude each other, this may occur with a „choice of“ structure or otherwise. If there is a condition other than „choice of“ the condition shall be described in the Description column).

“Multiplicity” defines how often a sub-element may occur within the structured data type. The used syntax is defined by UML notation. Most of the time the given information may co-relate with the Usage information (e.g. optional element may have a multiplicity of [0..1], mandatory element may have multiplicity of [1..1]). However, there are cases where Multiplicity may provide additional information (e.g. in cases of conditional and selective usage).

A.2 Basic data types

The data type definition in the IEC 62453 series is based on data types defined in IEC 61131. The following table (Table A.1) defines additional basic data types.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.1 – Basic data types

Data type	Description
ClassIdentifier	Technology specific identifier for a specific implementation class
LanguageID	Unique identifier for user interface localization. The ID is a standard international numeric abbreviation (e.g. German – Standard: 0x0407, English - United States: 0x0409)
ObjectReference	Technology specific reference to an object instance
UUID	A worldwide unique ID
URI	A compact string of characters used to identify or name a resource. The terms URL and URN are context-dependent aspects of URIs, and rarely need to be distinguished. Examples for URI are: The URI syntax is essentially a URI scheme name like “http”, “ftp”, “mailto”, “urn”, “file”, etc., followed by a colon character, and then a scheme-specific part

A.3 General data types

Namespace: fdt

The simple data types (see Table A.2) and structured data types (see Table A.4) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.2 – Simple general data types

Data type	Definition	Description
address	STRING	Parameter Addressing information. The format of the value for this parameter is described for each communication protocol in the corresponding section of the specification. The protocol portion of the specification provides the guidelines for address attribute. Most of the protocols specify that the DTM shall expose the addressing information to the Frame Application
alarmType	enumeration (highHighAlarm | highAlarm | lowLowAlarm | lowAlarm)	Identifier for the alarm type to show the association between high- and low alarm and high-high- and low-low-alarm
applicationDomain	STRING	The application domain, that applies to provided semanticId. This may be a protocol-specific ID or a different FDT-defined application domain
applicationId	enumeration (fdtAdjustSetValue | fdtAssetManagement | fdtAuditTrail | fdtConfiguration | fdtDiagnosis | fdtForce | fdtManagement | fdtObserve | fdtOfflineCompare | fdtOfflineParameterize | fdtOnlineCompare | fdtOnlineParameterize | fdtIdentify | fdtCalibration | fdtMainOperation | fdtNetworkManagement)	Identification of the current application context
binData	ARRAY OF USINT	Variable containing binary data
bitLength	UDINT	Length of a binary variable as bit count

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
bitPosition	UDINT	Position of a bit within a enumeration (0 based position)
boolValue	BOOL	Variable for configured static boolean data like alarm value
build	INT	build part of the FDT-Specification version on which the implementation of a DTM is based on. For example, for FDT-Specification 1.2.1.0 it shall be set to '0'
busAddress	STRING	
busRedundancy	UDINT	Number of redundant fieldbus interfaces
byteArray	ARRAY OF USINT	Data type used to transfer binary data
byteLength	UDINT	Number of bytes to describe data types like string
channelId	STRING	Id of a channel
channelMode	enumeration (communication | moduleSlot | processValue)	Type information enumeration for a channel
channelPath	STRING	Returns the path that identifies the channel within the device. The string shall not be empty. It always starts with the systemTag of the device instance followed by the channel id. The DTM has to guarantee that the path is unique for a device instance. The channel path is the base information to handle the system structure. <systemTag>/<id> Furthermore, the channelPath contains the information where to find the channel within the information available via [GetParameters] and so at least in case of a monolithic DTM the information whether the channel belongs to a device or module. In the case of Communication Channels there are some special rules for building the channelPath. How to generate the channelPath of a Communication Channel, which also acts as a Process Channel for data that it receives from a Process Channel of a child device, depends on the functionality of the channel. If the channel is passive, that means that it receives its data from a child device and provides this data without any changes within its own [ReadChannelData] information, the channelPath shall be built out of system tag and channel id of the own device instance and the system tag of the child device and the id of the marshalled channel. <systemTag>/<id>//<systemTag>/<id> If the channel is active, that means that it receives its data from a child device for processing and provides the result within its own [ReadChannelData] information, the channelPath shall be built out of the system tag and channel id of the own device instance. <systemTag>/<id> This allows navigation through the internal channel assignment of a device with gateway functionality
classificationDomainId	enumeration (powerDistribution | motionControl | measurement | operatorInterface | modulesAndControllers)	Device classification domain group according to IEC/TR 62390:2005, AnnexG. See Table A.3 below

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
classificationId	enumeration (flow | level | pressure | temperature | valve | positioner | actuator | nc_rc | encoder | speedDrive | hmi | analogInput | analogOutput | digitalInput | digitalOutput | electrochemicalAnalyser | dtmSpecific | universal | analyser | remoteIO | analogCombinedIO | digitalCombinedIO | recorder | controller | angle | limitSwitch | converter | motor | | switchboard | circuitBreaker | powerMonitoring | distributionPanel | contactor | protectionStarter | softStarter | drive | axisControl | motorControlCenter | controlValve | electrical | density | quality | speedOrRotaryFrequency | radiation | weightMass | distanceOrPositionPresence | pushButton | joystick | keypad | pilotLight | stackLight | display | combinedButtonsAndLights | operatorStation | generalInput | generalOutput | combinedInputOutput | relay | timer | scanner | programmableController)	Unique identifier for classification of a channel or device according to its primary function.'classificationId' should be selected from the table ' Device classification according to IEC/TR 62390:2005, Annex G' and use corresponding 'classificationDomainId' attribute for it. If suitable classification ID does not exist in that table, it should be selected from 'FDT classification ID table' and use it without 'classificationDomainId' attribute. See Table A.3 below
communicationError	enumeration (abort | busy | invalidCommunicationReference | noConnection | noParallelServices | noPendingRequest | unknownError | timeout | dtmSpecific | notSupportedFeature | sequenceTimeExpired)	Fieldbus protocol independent error occurred during communication. This kind of error is used especially if an error occurred during nested communication. The fieldbus specific communication error is part of the fieldbus specific parts of this specification. Communication errors detected by a communication component, e.g. a Communication-DTM, shall be handled within the data returned to the child component. All errors returned by a device as result of a fieldbus transaction should be returned by protocol specific response read or write response elements, typically by using fieldbus specific status and errors. All errors detected by the Communication-DTM shall be mapped to one of the fdt:communicationError enumeration values (e.g. abort busy invalidCommunicationReference noConnection noParallelServices noPendingRequest unknownError timeout dtmSpecific notSupportedFeature) and returned as a fdt:CommunciationError element to the child DTM. A DTM shall be able to handle both types of error responses for a transaction request sent to the parent component
communicationReferenceID	UUID	Identifier for a communication link to a device. This identifier is allocated by the communication component during the connect. The communication reference has to be used for all following communication calls

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
communicationType	enumeration (supported | required)	Indicates the type of the protocol support: • ‘supported’: bus protocol which can be provided by the DTM at a channel. • ‘required’: bus protocol which will be expected by the DTM from the parent DTM NOTE The actual supported bus protocols depend on the configuration of the DTM. This data type shows the possible supported, not the actual available protocols
dataSetID	UUID	Unique identifier of the data set version
dataSetState	enumeration (default | validModified | invalidModified | allDataLoaded)	State of an instance data set concerning modifications (refer to 4.8)
dataType	enumeration (byte | float | double | int | unsigned | enumerator | bitEnumerator | index | ascii | packedAscii | password | bitString | hexString | date | time | dateAndTime | duration | binary | structured | dtmSpecific)	Identifier for the data type of a transferred variable
dataTypeDescriptor	STRING	Description of data type
date	DATE	Date variable within an element
description	STRING	A human readable description of the supported and current data set format. Can be used to provide additional information to the user in case of partial level of support
descriptor	STRING	Human readable description within the context of an element
deviceGraphicState	enumeration (device | diagnosis | oem)	List of possible device states used in the DeviceIcon and DeviceBitmap element. Possible states are: - device: symbolic representation of the device - diagnostic: symbolic representation of device if there is online diagnosis available - oem: symbol representation of device in special operating modes (e.g. OEM service) The Frame Application should display the correct picture according to protocol specific rules
deviceTypeId	UDINT	Unique identifier for a device type within the name space of the fieldbus specification
deviceTypeInformation	STRING	Additional device type information supplied with a device. For example, a PROFIBUS device has to provide its GSD information as human readable string at this attribute. The information shall be based on the current locale according to the usage of DTM service SetLanguage NOTE The GSD information is accessible via: services InstanceDataInformation and GetTypeInformation
deviceTypeInformationPath	URI	Path to the file containing the information which is provided via the attribute ‘deviceTypeInformation’. It is recommended to use this attribute for providing additional protocol specific descriptions of the device type. The use of this attribute is specified in the protocol specific documents (IEC 62453-3xy).
deviceTypeVariant	STRING	Manufactuer specific device type variant definition string.
deviceTypeVariantInfo	STRING	Contains additional information for manufactuer specific device type variant definition.
display	STRING	Carries a human readable display string for tasks like documentation
displayFormat	STRING	Describes the display format for a display attribute (e.g. “%3.2f “ for a float value)

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
documentClassification	enumeration (help | technicalDocumentation | orderingInformation | miscellaneous)	Specifies the subject of the document
documentLanguageId	UDINT	Identifier for the language of the document. The language-id is defined as a Windows locale identifier (LCID)
dtmDevTypeID	UUID	dtmDevTypeID is a DTM specific unique identifier of the software related to the device type. For the same device type, the value remains unchanged although some identification information in DtmDeviceType element is updated. This can be a result of DTM update. The same dtmDevTypeID value shall always be related to the same device as in the previous DTM version (e.g. to provide backward compatibility)
dtmDevTypeIDVersion	UDINT	Version number of the dtmDevTypeID. It is also used to indicate that the information related to DtmDeviceType is changed. Frame Application can use this information, e.g. to update DTM related information in DTM library. An example: When a DTM is DD based, an upgrade of DD will cause the change of dtmDevTypeIDVersion without changing the dtmDevTypeID
errorCode	ARRAY OF USINT	Status information according to fieldbus specification
errorMessage	STRING	Human readable error message
filePath	URI	Path to a file .(e.g. document, executable)
frameApplicationTag	STRING	Frame Application specific tag used for identification and navigation. The DTM should display this tag at channel specific user interfaces
functionId	DINT	Identifier of a function provided by a DTM
help	STRING	Human readable help string for a document
id	STRING	Unique identifier for an element. This identifier is used within collections for the direct access of elements. This id shall be unique within the namespace of a device instance
idref	STRING	Reference to an element specified by the attribute ‘id’. The identifier is used within collections for the direct access of elements
index	UDINT	Index within an enumerator
invokeId	UUID	Identifier of a context of a service call, if the service provider handles several contexts in parallel. The identifier is used to reference former context calls
label	STRING	Human readable label for a document
languageId	UDINT	Identifier for a supported language or the language a DTM should interact. See also DTM service Setlanguage
levelOfSupport	enumeration (full | partial)	Provides an indication about the level of support of the supported data set version. - full – the data set is fully supported - partial – some of the information in the data set cannot be used in the upgrade procedure. Additional information should be provided in description attribute. Can be used to warn the user that some values can be lost.
major	INT	Major part of the FDT-specification version on which the implementation of a DTM is based on. For example, for FDT-specification 1.2.1.0 it shall be set to ‘1’
manufacturerId	UDINT	Unique identifier for a manufacturer within the name space of the fieldbus specification
minor	INT	Minor part of the FDT-specification version on which the implementation of a DTM is based on. For example, for FDT-specification 1.2.1.0 it shall be set to ‘2’

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
modifiedInDevice	BOOL	Flag to provide more information about the current state of the instance data set. TRUE, indicates that parameters have been changed in the device but not in instance data set (e.g.: see use case Online parameterization, DTM services to access to device data) 'modifiedInDevice' flag shall be set only once in case the data in the device has been changed. In the case of successful Upload or Download of complete data set, 'modifiedInDevice' flag shall be reset (set to FALSE). Data in the device may also be modified directly by a tool out of the scope of the FDT. In this case, the flag 'modifiedInDevice' should not be set. Data set shall be locked before an application is started, which may need to change the flag "modifiedInDevice" (the flag 'modifiedInDevice is part of the instance data set and could not be changed if the data set is not locked)
name	STRING	Human readable name within the context of an element
number	INT	Number variable like float, integer or other numeric data types
onlineStatus	enumeration (communicationSet | goingOnline | goingOffline | online)	Information about online state of a DTM describing the 'communication allowed' substates
operationPhase	enumeration (notSupported | engineering | commissioning | runtime | service)	Information about the current operation phase
orderCode	STRING	Order code of the device variant
parameters	STRING	Contains the parameters to be passed to the application, if the file attribute specifies an executable file
path	URI	Path to the icon for a device
percent	USINT	State of progress 0..100%
physicalLayer	UUID	CATID for description of a physical layer of a fieldbus
physicalLayerName	STRING	Human readable description for the physical layer
protectedByChannelAssignment	BOOL	Defines if the channel is set to read only by the Frame Application. This is set to TRUE if a channel assignment exists
protocolId	UUID	Identification of a protocol. Standard values for this data type are defined in the protocol specific documents (IEC 62453-3xy)
protocolIdName	STRING	Human readable name of a protocol, associated to a protocolId. Standard values for this data type are defined in the protocol specific documents (IEC 62453-3xy)
readAccess	BOOL	Specifies whether the value can be read from a device
reference	STRING	Reference to a variable of a structure
release	INT	Release part of the FDT-specification version on which the implementation of a DTM is based. For FDT-specification 1.2.1.0 it shall be set to '1'.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
semanticId	STRING	Semantic identifier for an element. This identifier shall be unique at least within the context of an element. By using this attribute, e.g. a Frame Application is able to get the information regarding the meaning and usage of a single data structure. The definition regarding the identifier is protocol specific and described in protocol specific annexes. Furthermore, the following protocol independent semanticIds are defined to cover the network information: - fdt::DeviceAddress - fdt:busMasterConfigurationPart A parameter with one of these semanticIds shall be the same parameter as the corresponding information (e.g. bus address) provided by service NetworkManagementInfoRead (see 7.2.11.1)
serviceSucceeded	BOOL	TRUE indicates succeeded service, FALSE indicates failed service
showProgress	BOOL	Set to TRUE, if the progress should be displayed, otherwise the progress would not be shown and an open progress bar shall be closed within the Frame Application
signalType	enumeration (input | output)	Specifies a signal as input or output
staticValue	INT	Variable for configured static Data like an alarm value
statusFlag	enumeration (ok | warning | error | invalid)	Identifier for the current status of a device or module
storageState	enumeration (persistent | transient)	State of an instance data set concerning the persistent state (refer to 4.8)
string	STRING	String variable within an element
subDeviceType	STRING	Manufacturer specific unique identifier for a device type within the name space of the device type id. This parameter shall be passed to the DTM during initialization via the Initialize service to advise a pre-configuration for the requested subtype. For example, the same transmitter can be pre-configured for level or flow measuring
substituteType	enumeration (lastValue | lastValidValue | upperRange | lowerRange)	Type of an substitute value
systemTag	STRING	Unique identification of DTM within Frame Application
tag	STRING	Unique identifier for a device, module, or channel
time	DATE_AND_TIME	Time variable within an element
url	URI	Contains a URL to a document in the Web
vendor	STRING	Human readable description of the vendor of component
version	STRING	Human readable description of the version of component like “1.0”
windowTitle	STRING	Window title required by the Frame Application
writeAccess	BOOL	Specifies whether the value can be written into a device

Table A.3 defines the value range for classification of device types.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.3 – Definition of classificationId enumeration values

Domain ()		**Subdomain**
classificationDomainId	**IEC domain group name**	**classificationId**
General FDT	<none>	valve
		actuator
		nc_rc
		encoder
		speedDrive
		hmi
		analogInput
		analogOuput
		digitalInput
		digitalOutput
		electrochemicalAnalyser
(..continuing General FDT)		dtmSpecific
		universal
		analyser
		remoteIO
		analogCombinedIO
		digitalCombinedIO
		recorder
		controller
		angle
		limitSwitch
		converter
		motor
powerDistribution	Power distribution	switchboard
		circuitBreaker
		powerMonitoring
		distributionPanel
motionControl	Motion control	contactor
		protectionStarter
		softStarter
		drive
		axisControl
		motorControlCenter
		motorMonitoring
		positioner
		controlValve
measurement	Detection, measurement	electrical
		density
		flow
		level
		quality

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Domain ()		Subdomain
classificationDomainId	IEC domain group name	classificationId
		pressure
		speedOrRotaryFrequency
		radiation
		temperature
		weightMass
		distanceOrPositionOrPresence
operatorInterface	Dialogue / operator interfaces	pushButton
		joystick
		keypad
		pilotLight
		stackLight
		display
		combinedButtonsAndLights
		operatorStation
modulesAndControllers	Logic / universal I/O modules and controllers	generalInput
		generalOuput
		combinedInputOuput
		relay
		timer
		scanner
		programmableController

Table A.4 – General structured data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Alarm	STRUCT			Description of an alarm
	alarmType	M	[1..1]	
	Unit	O	[0..1]	
	choice of	O	[0..1]	
	StaticValue	S	[1..1]	
	NumberData	S	[1..1]	
	AlarmEnumerationEntries	S	[1..1]	
	ChannelReferences	S	[1..1]	
AlarmEnumerationEntries	STRUCT			Collection of alarm enumeration entries
	AlarmEnumerationEntry	M	[1..*]	
AlarmEnumerationEntry	STRUCT			Alarm enumeration entry
	bitPosition	M	[1..1]	
	name	M	[1..1]	
	boolValue	M	[1..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	descriptor	O	[0..1]	
Alarms	STRUCT			Collection of alarms
	Alarm	M	[1..*]	
ApplicationId	STRUCT			FDT global application id coded as an enumeration
	applicationId	M	[1..1]	
FDTApplicationIds	STRUCT			Collection of application id
	ApplicationId	O	[0..*]	
BinaryVariable	STRUCT			Element containing binary data
	binData	M	[1..1]	
BitEnumeratorEntries	STRUCT			Collection of EnumerationEntry
	EnumeratorEntry	M	[1..*]	
BitEnumeratorVariable	STRUCT			Current EnumeratorEntry and the collection of possible EnumeratorEntries
	BitVariable	M	[1..1]	
	BitEnumeratorEntries	O	[0..1]	
BitVariable	STRUCT			Selected element of an enumeration
	EnumeratorEntry	O	[0..*]	
BoolValue	STRUCT			Variable for configured static boolean data
	boolValue	O	[0..1]	
BusCategories	STRUCT			Collection of BusCategory
	BusCategory	M	[1..*]	
BusCategory	STRUCT			Description of a Fieldbus
	protocolId	M	[1..1]	
	protocolIdName	O	[0..1]	
	CommunicationTypeEntry	O	[0..*]	
	PhysicalLayer	O	[0..*]	
BusRedundancy	STRUCT			Number of redundant fieldbus interfaces
	busRedundancy	M	[1..1]	
ChannelInformation	STRUCT			Type information for a FDT channel. This element is used within a document returned by InstanceDataInformation service
	BusCategories	M	[1..1]	The BusCategories element should contain the list of supported fieldbus protocols of this channel
	ChannelModes	M	[1..1]	
ChannelMode	STRUCT			Type information element for a channel
	channelMode	M	[1..1]	
ChannelModes	STRUCT			List of ChannelMode elements
	ChannelMode	M	[1..*]	
ChannelObjectReference	ObjectReference			Reference to a Channel object.
ChannelReference	STRUCT			Reference to an object identified by its

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	idref	M	[1..1]	id
	ChannelInformation	O	[0..1]	
ChannelReferences	STRUCT			Collection of ChannelObjectReferences
	ChannelReference	M	[1..*]	
ChannelType	STRUCT			Description of the channel component
	VersionInformation	M	[1..1]	
	BusCategory	O	[0..1]	
ClassificationId	STRUCT			Classification Id. See tables below
	classificationId	M	[1..1]	
ClassificationIds	STRUCT			Collection of classificationId elements. See tables below
	classificationDomainId	O	[0..1]	
	ClassificationId	M	[1..*]	
CommunicationClientObjectReference	ObjectReference			Reference to a communication client object
CommunicationData	STRUCT			Variable used to transfer binary communication data
	byteArray	M	[1..1]	
CommunicationError	STRUCT			Description of a fieldbus protocol independent error occurred during nested communication with:
	communicationError	M	[1..1]	protocol independent error
	tag	M	[1..1]	the tag of the Communication Channel’s device
	errorCode	O	[0..1]	fieldbus specific error code
	descriptor	O	[0..1]	error description
CommunicationObjectReference	Technology specific ObjectReference			Reference to communication object. This data type depends on the used implementation technology (see documents IEC 62453-4z)
CommunicationTypeEntry	STRUCT			Enumeration element for the communication type
	communicationType	M	[1..1]	
CurrentDatasetFormat	STRUCT			A current version of the data set used for saving the data
	dataSetID	M	[1..1]	
	description	O	[0..1]	
DataSetFormats	STRUCT			Data formats of the persisted data used and supported by the DTM
	CurrentDatasetFormat	M	[1..1]	
	SupportedDatasetFormat	O	[0..*]	
Deadband	STRUCT			Deadband is the amount of value changes that triggers for example new trend values
	number	M	[1..1]	
DeviceBitmap	STRUCT			Bitmap for a device in BMP format (70*40 pixels (width*height), 16 colors)
	deviceGraphicState	M	[1..1]	
	path	M	[1..1]	
DeviceIcon	STRUCT			Icon for a device

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	deviceGraphicState	O	[0..1]	
	path	M	[1..1]	
DeviceTypeVariant	STRUCT			Contains manufacturer specific device type variant information. A device type variant is a property of the physical device that has no influence to the software (i.e. flange material, Ex certificate ...). However some Frame Applications will use this information for documentation purposes
	deviceTypeVariant	M	[1..1]	
	deviceTypeVariantInfo	O	[0..1]	
	orderCode	O	[0..1]	
DeviceTypeVariants	STRUCT			Collection of DeviceVariant elements. The DeviceVariants element should only be used within context of DTM information data
	DeviceTypeVariant	M	[1..*]	
Display	STRUCT			Display variable
	string	M	[1..1]	
DisplayContext	Technology specific display context			Contex information for display. This data type depends on the used implementation technology (see documents IEC 62453-4z)
DocumentExe	STRUCT			Defines a executable, which is used to show DTM documents
	file	M	[1..1]	
	parameters	O	[0..1]	
DocumentFile	STRUCT			Defines a file document
	filePath	M	[1..1]	
DocumentUrl	STRUCT			Defines a URL to a document in the Web Implementation hint: The file, url and executable documents can be implemented by using the ShellExecute() function.
	url	M	[1..1]	
DtmDeviceType	STRUCT			Description of a device type
	readAccess	O	[0..1]	
	writeAccess	O	[0..1]	
	manufacturerId	O	[0..1]	
	deviceTypeId	O	[0..1]	
	subDeviceType	O	[0..1]	
	deviceTypeInformation	O	[0..1]	
	deviceTypeInformationPath	O	[0..1]	
	classificationId	O	[0..1]	
	dtmDevTypeID	M	[1..1]	
	dtmDevTypeIDVersion	M	[1..1]	
	VersionInformation	M	[1..1]	
	SupportedLanguages	M	[1..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	BusCategories	O	[0..1]	
	DeviceIcon	O	[0..*]	
	DtmDocuments	O	[0..1]	
	DeviceBitmap	O	[0..*]	
	ClassificationIds	O	[0..1]	
	DataSetFormats	O	[0..1]	
	choice of	O	[0..1]	
	DeviceTypeVariants	S	[1..1]	
	DeviceTypeVariant	S	[1..1]	
DtmDeviceTypes	STRUCT			Collection of device types
	DtmDeviceType	M	[1..*]	
DtmDocument	STRUCT			Definition of a document, which is provided by a DTM and displayed by the Frame Application
	label	M	[1..1]	
	help	O	[0..1]	
	documentClassification	M	[1..1]	
	documentLanguageId	O	[0..1]	
	choice of	M	[1..1]	
	DocumentFile	S	[0..1]	
	DocumentUrl	S	[0..1]	
	DocumentExe	S	[0..1]	
DtmDocuments	STRUCT			List of DTM documents. This tag allows a DTM to publish documents
	collection of	M	[1..1]	
	DtmDocument		[1..*]	
DtmObjectReference	ObjectReference			Reference to DTM
DtmVariable	STRUCT			Variable description with name, value, range, etc.
	SemanticInformation	M	[1..*]	The DTM shall provide a SemanticInformation element for all supported fieldbus protocol of the DTM instance
	name	M	[1..1]	
	descriptor	O	[0..1]	
	Value	M	[1..1]	
	Unit	O	[0..1]	
	Ranges	O	[0..1]	
	statusFlag	O	[0..1]	
	StatusInformation	O	[0..1]	
DtmVariableReference	STRUCT			Reference to a DTM variable
	reference	M	[1..1]	
DtmVariables	STRUCT			Collection of DTM variables
	SemanticInformation	O	[0..*]	
	name	M	[1..1]	
	descriptor	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	collection of	M	[1..1]	
	DtmVariables		[0..*]	
	DtmVariable		[0..*]	
EnumeratorEntries	STRUCT			Enumeration element
	EnumeratorEntry	M	[1..*]	
EnumeratorEntry	STRUCT			Element of an enumeration
	index	M	[1..1]	
	name	M	[1..1]	
	descriptor	O	[0..1]	
EnumeratorVariable	STRUCT			Current EnumeratorEntry and the collection of possible EnumeratorEntries
	Variable	M	[1..1]	
	EnumeratorEntries	O	[0..1]	
FDTVersion	STRUCT			Definition of the element which specifies the version information on which the implementation of a DTM is based on
	major	M	[1..1]	
	minor	M	[1..1]	
	release	M	[1..1]	
	build	M	[1..1]	
FrameObjectReference	ObjectReference			Reference to Frame Application
FrameVersion	STRUCT			Describes the version of the Frame Application
	FDTVersion	M	[1..1]	
LanguageId	STRUCT			Contains the languageId (refer to languageId)
	languageId	M	[1..1]	
LowerRange	STRUCT			Defintion of a lower range element
	choice of	O	[0..1]	
	NumberData	S	[1..1]	
	StringData	S	[1..1]	
	TimeData	S	[1..1]	
	ChannelReference	S	[1..1]	
LowerRawValue	STRUCT			A numeric entry which reflects the real via, e.g. PROFIBUS transferred value. The value is mapped to the related range (LowerRangeValue). For example, if a PROFIBIUS device maps a range from 10 to 100 mbar to an integer of value 1024-4096 the 'LowerRawValue' contains 1024, the 'UpperRawValue' contains 4096.
	number	M	[1..1]	
NumberData	STRUCT			Number variable like float, integer or other numeric data types
	number	M	[1..1]	
PhysicalLayer	STRUCT			Unique identifier for a physical layer of a fieldbus like Profibus PA
	physicalLayer	M	[1..1]	
	physicalLayerName	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
PresentationObject-Reference	ObjectReference			Reference to a Presentation Object
ProgressInformation	STRUCT			Progress information with:
	description			Description of the running process
	percent			
	showProgress			
Range	STRUCT			Describes the valid range of a process variable
	LowerRange	M	[1..1]	
	UpperRange	M	[1..1]	
	Unit	O	[0..1]	
	LowerRawValue	O	[0..1]	
	UpperRawValue	O	[0..1]	
Ranges	STRUCT			Collection of ranges
	readAccess	O	[0..1]	
	writeAccess	O	[0..1]	
	Range	M	[1..*]	
SemanticInformation	STRUCT			The element provides semantic information for a data object. For each object at least one SemanticInformation element with an FDT-defined protocol-specific semanticId shall be provided. The DTM shall provide a SemanticInformation element for all supported fieldbus protocol of the DTM instance
	applicationDomain	M	[1..1]	
	semanticId	M	[1..1]	
	address	O	[0..1]	
StaticValue	STRUCT			Variable for configured static data like an alarm value
	staticValue	M	[1..1]	
StatusInformation	STRUCT			Current status information of a device or module
	readAccess	O	[0..1]	
	writeAccess	O	[0..1]	
	EnumeratorEntry	M	[1..*]	
StringData	STRUCT			String variable
	string	M	[1..1]	
StorageObject	ObjectReference			Technology specific object used in data save and load services
StructuredElement	STRUCT			Variable as display value or as reference to a variable with data length information
	bitLength	M	[1..1]	
	choice of	M	[1..1]	
	Display	S	[1..1]	
	DtmVariableReference	S	[1..1]	
StructuredElements	STRUCT			Collection of structured elements
	StructuredElement	M	[1..*]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
StructuredVariable	STRUCT			Describes a binary value and its structure information
	BinaryVariable	M	[1..1]	
	StructuredElements	M	[1..1]	
	dataTypeDescriptor	O	[0..1]	
SubstituteType	STRUCT			Type of a substitute value
	substituteType	M	[1..1]	
SubstituteValue	STRUCT			Describes a substitute value which is used in combination with the behavior of disturbed channel values
	choice of	M	[1..1]	
	SubstituteType	S	[1..1]	
	Variant	S	[1..1]	
SupportedDatasetFormat	STRUCT			A list of the supported data set that can be upgraded by the DTM
	dataSetID	M	[1..1]	
	levelOfSupport	O	[0..1]	
	description	O	[0..1]	
SupportedLanguages	STRUCT			Collection of language ids
	LanguageId	M	[1..*]	
TimeData	STRUCT			Element of a time date value
	time	M	[1..1]	
Unit	STRUCT			Current unit and the collection of possible units of a process variable
	readAccess	O	[0..1]	
	writeAccess	O	[0..1]	
	choice of	O	[0..1]	
	EnumeratorVariable	S	[1..1]	
	ChannelReference	S	[1..1]	
UpperRange	STRUCT			Definition of an upper range element
	choice of	O	[0..1]	
	NumberData	S	[1..1]	
	StringData	S	[1..1]	
	TimeData	S	[1..1]	
	ChannelReference	S	[1..1]	
UpperRawValue	STRUCT			A numeric entry which reflects the real via, e.g. PROFIBUS transferred value. The value is mapped to the related range (UpperRangeValue). For example if a PROFIBIUS device maps a range from 10 to 100 mbar to an integer of value 1024-4096 the 'UpperRawValue' contains 4096, the 'UpperRange' contains 4096.
	number	M	[1..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Value	STRUCT			Contains the display string for a variable or the variable itself
	readAccess	O	[0..1]	
	writeAccess	O	[0..1]	
	choice of	M	[1..1]	
	Display	S	[1..1]	
	Variant	S	[1..1]	
Variable	STRUCT			Selected element of an enumeration
	EnumeratorEntry	M	[1..1]	
Variant	STRUCT			Variable with data type and display format
	dataType	M	[1..1]	
	byteLength	O	[0..1]	
	displayFormat	O	[0..1]	
	choice of	M	[1..1]	
	StringData	S	[1..1]	
	NumberData	S	[1..1]	
	TimeData	S	[1..1]	
	EnumeratorVariable	S	[1..1]	
	BitEnumeratorVariable	S	[1..1]	
	BinaryVariable	S	[1..1]	
	StructuredVariable	S	[1..1]	
VersionInformation	STRUCT			Description of the version of a component where used. For example - version of DTM (if used in DtmInfo) - version of physical device (if used in DtmDeviceType)
	readAccess	O	[0..1]	
	writeAccess	O	[0..1]	
	name	M	[1..1]	
	vendor	O	[0..1]	
	version	O	[0..1]	
	date	O	[0..1]	
	descriptor	O	[0..1]	

A.4 User information data types

Namespace: user

The simple data types (see Table A.5) and structured data types (Table A.6) defined in this clause are used as a base for definition of service specific data types or as service arguments.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.5 – Simple user information data types

Data type	Definition	Description
administrator	BOOL	Flag describing if the user has administrative right (default to “FALSE”)
loginLocation	STRING	Description of the location the user logged in
loginTime	DATE_AND_TIME	Time of last login
oemService	BOOL	Flag describing if the user has OEM service rights (default to “FALSE”)
projectName	STRING	Unique identifier for the project within the name space of the Frame Application
sessionDescription	STRING	Description of the user session (may be given by the user)
userLevel	enumeration (observer | operator | maintenance | planningEngineer)	User level specifying users rights
userName	STRING	Name of the current user

Table A.6 – Structured user information data type

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
FDTUserInformation	STRUCT			Description of the user role including information about permissions, current session, etc.
	projectName	M	[1..1]	
	userName	M	[1..1]	
	userLevel	M	[1..1]	
	oemService	O	[0..1]	
	administrator	O	[0..1]	
	loginTime	O	[0..1]	
	loginLocation	O	[0..1]	
	sessionDescription	O	[0..1]	

A.5 DTM information data type

Namespace: dtmInfo

The data type (see Table A.7) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.7 – Structured DTM information data type

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DtmInfo	STRUCT			Describes the DTM itself
	fdt:FDTVersion	M	[1..1]	
	fdt:VersionInformation	M	[1..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	fdt:DtmDeviceTypes	M	[1..*]	

A.6 BTM data types

Namespace: btm

The simple data types (see Table A.8) and structured data types (see Table A.9) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.8 – Simple BTM data types

Data type	Definition	Description
blockCategory	UUID	Unique identifier for a block type like Analog Input, Resource Block
blockCategoryName	STRING	Human readable block type name
blockTag	STRING	Block Tag is a unique identifier for the block
index	UDINT	An integer value that provides the offset from the beginning of the block or from the beginning of the structure
label	STRING	Readable name. If the optional attribute 'label' exists, the attribute 'name' contains a language independent identifier
profile	UINT	Number assigned by the Fieldbus Foundation that uniquely identifies the profile on which the block is based
profileRevision	UINT	The revision number of the profile on which the block definition is based
slot	UDINT	An integer provides parameter-grouping number. In Profibus, it can provide the slot number (if the device support slots). In Foundation fieldbus, it can provide the Application VFD number if the instrument supports more than one application VFD
usage	enumeration (contained | input | output | eventSource | eventSink)	An indication of whether this object may be set by another function block (I), provides a value to another function block (O), or only provides communications support (C) to devices other than function blocks - of type Visible String, 1 Octet

Table A.9 – Structured BTM data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
BlockIcon	STRUCT			Icon for a Block
	fdt:path	M	[1..1]	
BlockTypeCategory	STRUCT			Type of the Block
	blockCategory	M	[1..1]	
	blockCategoryName	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
BtmBlockType	STRUCT			Description of a block type
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	
	fdt:manufacturerId	O	[0..1]	
	profile	O	[0..1]	
	profileRevision	O	[0..1]	
	fdt:VersionInformation	M	[1..1]	
	fdt:SupportedLanguages	O	[0..1]	
	BlockTypeCategory	O	[0..1]	
	fdt:BusCategories	M	[1..1]	
	BlockIcon	O	[0..1]	
BtmVariable	STRUCT			Block variable description with name, index, value, range, etc.
	fdt:name	M	[1..1]	
	label	M	[1..1]	
	index	M	[1..1]	
	slot	O	[0..1]	
	fdt:descriptor	O	[0..1]	
	usage	M	[1..1]	
	fdt:Value	M	[1..1]	
	fdt:Unit	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:statusFlag	O	[0..1]	
	fdt:StatusInformation	O	[0..1]	
BtmVariables	STRUCT			Collection of BTM variables
	fdt:name	M	[1..1]	
	label	M	[1..1]	
	index	M	[1..1]	
	collection of	M	[1..1]	
	BtmVariables		[0..*]	
	BtmVariable		[0..*]	

A.7 Device and Scan identification data types

Namespace: ident

The data types defined in this clause (see Table A.10 and Table A.11) are used as a base for definition of service specific data types or as service arguments.

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.10 – Simple device identification data types

Data type	Definition	Description
configuredState	enumeration (configuredAndPhysicallyAvailable | configuredAndNotPhysicallyAvailable | availableButNotConfigured | notApplicable)	The attribute is only used for protocols, which have a master configuration like Profibus. For all other protocols “notApplicable” is to be used. This attribute defines, if a scanned module connected to this bus address is in configured state or. the attribute is optionally used by Frame Applications to display the information to the user. Following values are valid in the enumeration: - configuredAndPhysicallyAvailable - configuredAndNotPhysicallyAvailable - availableButNotConfigured - notApplicable
idDTMSupportLevel	enumeration (genericSupport | profileSupport | blockspecificProfileSupport | specificSupport | identSupport)	Defines the support level of a DTMDeviceType for a physical device. This attribute can be used by a Frame Application to display the support level of a DTM to the user. Users may use this information for an assignment decision. genericSupport: DTMDeviceType applies to all kinds of physical devices of the corresponding fieldbus protocol profileSupport: DTMDeviceType applies to physical devices of a certain profile of the fieldbus protocol. blockspecificProfileSupport DTMDeviceType applies to blocks included in physical device types (e.g. Pressure TransducerBlock in Profibus PA) specificSupport (DTMDeviceType is developed for this physical device type. identSupport: The DTMDeviceType is capable of identifying the physical device in a vendor specific manner and to propose a better DTMDeviceType
match	STRING	Attribute contains a regular expression string, which shall match with an attribute provided by scan result
name	STRING	Contains the name of one single identification information
nomatch	STRING	Attribute contains a regular expression string, which shall not match with an attribute provided by scan result
protocolSpecificName	STRING	Fieldbus protocol specific name. This attribute provides a reference to the fieldbus specification
resultState	enumeration (provisional | final | error)	Marks the XML document as first provisional result provided by ScanResponse(). DTM will send further provisional XMLs (resultState = “provisional”) and one result document at the end of the scan operation (resultState = “final”). In case of an error, the result indicates this by setting resultState = error
value	STRING	Attribute contains a regular expression string, which shall match with an attribute provided by scan result

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.11 – Structured device identification data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DeviceTypeIdentification	STRUCT			Contains all identification elements of a device type for a physical device type or group
	idDTMSupportLevel	M	[1..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdValues	O	[0..1]	
DeviceIdentificationList	STRUCT			List of identifications of several device types for physical device types. The DeviceTypeIdentifications element should not be used in context of data returned by service GetIdentificationInformation. The element should only be used in context of data which could contain identification information for several device types (i.e. services Scan, HardwareInformation)
	DeviceIdentification	M	[1..*]	
IdAddress	STRUCT			Contains the field bus address of a device or a block address. This information is available in scan response only. It can not be used for matching
	ident:value	M	[1..1]	
	ident:protocolSpecificName	M	[1..1]	
IdBusProtocol	STRUCT			Provides the protocol variant, e.g. Profibus PA
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdBusProtocolVersion	STRUCT			Contains the version of the protocol, e.g. 5 or 6 in case of HART, or 3.0 in case of Profibus PA
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdDeviceTag	STRUCT			Contains the tag of the scanned device or block instance. This attribute is only available in scan result and for information only. It cannot be used for matching
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
IdManufacturer	STRUCT			Identifies the manufacturer of the device or block
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
IdSerialNumber	STRUCT			Contains the serial number of a scanned device or block. This attribute is only available in scan online process and for information only. It cannot be used for matching. NOTE 1 This serialnumber might by only unique for one manufacturer and one devicetype. NOTE 2 In order to present a worldwide unique device identification, a Frame Application has to combine IdManufacturer, IdTypeID and IdSerialNumber
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
IdTypeID	STRUCT			Identifies the device or block type
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdSoftwareRevision	STRUCT			Contains the software version of the device or block
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdHardwareRevision	STRUCT			Contains the hardware version of the device or block
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdDTMSupportLevel	STRUCT			This element is used only in DTMDeviceTypeIdentSchema and not in DTMScanIdentSchema. It cannot be used for matching. It can be displayed by Frame Applications applications for an assignment decision by user
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdValue	STRUCT			Contains name value pair for one identification parameter without semantic information for the Frame Application
	name	M	[1..1]	
	value	O	[0..1]	
	protocolSpecificName	M	[1..1]	
	RegExpr	O	[0..*]	This element is not available in a Response of service Scan
IdValues	STRUCT			List of identification elements, which are not specifically defined by Idxxx elements listed above. No further protocol independent semantic information is available for those identification elements
	IdValue	O	[0..*]	
RegExpr	STRUCT			Contains one or more regular expression attributes (refer to Regular Expression Specification clause) of type match or nomatch
	match	O	[0..1]	
	nomatch	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
ScanIdentification	STRUCT			Contains all identification elements for one single scanned device or block. This includes elements with well defined semantics as well as IdValues
	configuredState	O	[0..1]	
	fdt:CommunicationError	O	[0..1]	
	IdBusProtocol	M	[1..1]	
	IdBusProtocolVersion	M	[1..1]	
	IdAddress	M	[1..1]	
	IdManufacturer	M	[1..1]	
	IdTypeID	M	[1..1]	
	IdSoftwareRevision	M	[1..1]	
	IdHardwareRevision	M	[1..1]	
	IdDeviceTag	M	[1..1]	
	IdSerialNumber	M	[1..1]	
	IdValues	O	[0..1]	
ScanIdentificationList	STRUCT			List of identifications for several scanned physical devices or blocks
	fdt:protocolId	M	[1..1]	
	resultState	M	[1..1]	
	ScanIdentification	O	[0..*]	

A.8 Function data types

Namespace: func

The simple data types (see Table A.12) and structured data types (see Table A.13) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.12 – Simple function data types

Data type	Definition	Description
checked	BOOL	Status of a menu entry
defaultFunctionId	DINT	Contains a function ID which can be used to open a default presentation object or to raise a default function
enabled	BOOL	Status of a menu entry
hasGUI	BOOL	Specifies whether the DTM supports a user interface for a extended function
help	STRING	Human readable help string for a function
hidden	BOOL	Status of a menu entry
label	STRING	Human readable label of a function
mime-type	STRING	Mime-type of a document provided by a DTM
moduleId	UDINT	Unique identifier for a module within the name space of the device instance. The same ids as defined in the DTMParameter document shall be used
path	URI	Path to an object that has to be embedded for a function like bitmaps or other graphical elements

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
printable	BOOL	Specifies whether the DTM supports a printable document for a function (access via service GetDocumentation)
printableStandardFunction	BOOL	Specifies whether the standard function is printable
programName	STRING	Specifies the name of a document viewer program
resizable	BOOL	Specifies whether the GUI is resizable
resizableStandardFunction	BOOL	Specifies whether the GUI of a standard function is resizable
separator	BOOL	Special menu entry
toggle	BOOL	Status of a menu entry whether it is active or not

Table A.13 – Structured function data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Document	STRUCT			Definition of a document, which is provided by a DTM and displayed by the Frame Application, if a user selects the entry
	label	M	[1..1]	
	help	M	[1..1]	
	path	M	[1..1]	
	choice of	M	[1..1]	
	MimeType	S	[1..1]	
	OpenWith	S	[1..*]	
	Icon	O	[0..1]	
FDTFunctionCall	STRUCT			Definition of the element which specifies a specific function call. Contains fdt:FunctionID
	fdt:functionId	M	[1..1]	
	fdt:ApplicationId	O	[0..1]	
	ops:operationPhase	O	[0..1]	
Function	STRUCT			Definition of a single function entry, which is not a standard function (concerning the ApplicationIds): Contains a fdt:FunctionID
	label	M	[1..1]	
	fdt:name	M	[1..1]	
	help	M	[1..1]	
	hasGUI	O	[0..1]	
	printable	O	[0..1]	
	resizable	O	[0..1]	
	functionId	M	[1..1]	
	Icon	O	[0..1]	
	Status	O	[0..1]	
	fdt:FDTApplicationIds	O	[0..1]	
Functions	STRUCT			Definition of a group of function entries
	label	M	[1..1]	
	fdt:name	M	[1..1]	
	help	M	[1..1]	
	moduleId	O	[0..1]	
	Status	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	**Usage**	**Multiplicity**	
	collection of	O	[0..*]	
	Function		[0..*]	
	Document		[0..*]	
	StandardFunction		[0..*]	
	Functions		[0..*]	
	StandardFunctions		[0..*]	
GUIInformation	<technology specific>			Information about how to start a user interface of a function
Icon	STRUCT			Icon for a function
	path	M	[1..1]	
MimeType	STRUCT			Mime-type of a document provided by a DTM. The mime-type is used by the Frame Application to determine an appropriated viewer for a document
	mime-type	O	[0..1]	
OpenWith	STRUCT			Name of the program, which should be used to open a document
	programName	O	[0..1]	
StandardFunction	STRUCT			Definition of a single function entry, which is a standard function (concerning the applicationIds).This entry has not an own label. The label shall be supplied by the Frame Application
	fdt:name	M	[1..1]	
	help	M	[1..1]	
	printableStandardFunction	O	[0..1]	
	resizableStandardFunction	O	[0..1]	
	functionId	M	[1..1]	
	Icon	O	[0..1]	
	Status	O	[0..1]	
	fdt:ApplicationId	M	[1..1]	
StandardFunctions	STRUCT			Definition of a group of standard function entries. This entry has not an own label. The label shall be supplied by the Frame Application. It can not contain standard functions
	fdt:ApplicationId	M	[1..1]	
	fdt:name	M	[1..1]	
	help	M	[1..1]	
	moduleId	O	[0..1]	
	Status	O	[0..1]	
	collection of	O	[0..*]	
	Function		[0..*]	
	Functions		[0..*]	
Status	STRUCT			Status of a function
	toggle	M	[1..1]	
	checked	M	[1..1]	
	enabled	M	[1..1]	
	hidden	M	[1..1]	
	separator	M	[1..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

A.9 AuditTrail data types

Namespace: auditTrail

The simple data types (see Table A.14) and structured data types (see Table A.15) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.14 – Simple auditTrail data types

Data type	Definition	Description
path	URI	Path to an object that has to be embedded within the document like bitmaps or other graphical elements
title	STRING	Human readable title for the documentation

Table A.15 – Structured auditTrail data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
AuditTrailDeviceStatusEvent	STRUCT			Description of the status of a device
	fdt:statusFlag	M	[1..1]	
	fdt:StatusInformation	O	[0..1]	
LogEntry	STRUCT			Notification about changed variables, executed functions, or device status events
	fdt:descriptor	O	[0..1]	
	choice of	M	[1..1]	
	AuditTrailFunctionEvent	S	[1..1]	
	AuditTrailVariableEvent	S	[1..1]	
	AuditTrailDeviceStatus-Event	S	[1..1]	
AuditTrailFunction-Event	STRUCT			Description about an executed function
	fdt:display	M	[1..1]	
AuditTrailVariable	STRUCT			Description of a changed variable
	fdt:display	M	[1..1]	
	fdt:Unit	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:statusFlag	O	[0..1]	
	fdt:StatusInformation	O	[0..1]	
AuditTrailVariable-Event	STRUCT			Notification about a changed variable
	fdt:name	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	ChangedFrom	M	[1..1]	
	ChangedTo	M	[1..1]	
ChangedFrom	STRUCT			Last value of an audit trail variable
	AuditTrailVariable	M	[1..1]	
ChangedTo	STRUCT			New value of an audit trail variable
	AuditTrailVariable	M	[1..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
TransactionInfo	STRUCT			Used to request start and stop audit trail for the following actions (e.g. configuration or simulation)
	systemTag	M	[1..1]	

A.10 Documentation data types

Namespace: doc

The simple data types (see Table A.16) and structured data types (see Table A.17) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.16 – Simple documentation data types

Data type	Definition	Description
path	URI	Path to an object that has to be embedded within the document like bitmaps or other graphical elements
title	STRING	Human readable title for the documentation

Table A.17 – Structured documentation data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
Documentation	STRUCT			Documentation of a DTM for a specific FDTFunctionCall
	title	M	[1..1]	
	fdt:classificationId	O	[0..1]	
	fdt:manufacturerId	O	[0..1]	
	fdt:deviceTypeId	O	[0..1]	
	dtmi:deviceTypeInformation	O	[0..1]	
	fdt:descriptor	O	[0..1]	
	fdt:date	O	[0..1]	
	fdt:windowTitle	O	[0..1]	
	fdt:VersionInformation	M	[1..1]	
	fdt:ClassificationIds	O	[0..1]	
	DocumentVariables	M	[1..1]	
	choice	O	[0..1]	
	DTMStyleForComplete-Document	M	[1..1]	
	DTMSpecificXMLData	M	[1..1]	
DocumentVariable	STRUCT			Human readable variable description with name, value, range, etc.
	fdt:name	M	[1..1]	
	fdt:descriptor	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
	fdt:display	M	[1..1]	
	fdt:Unit	O	[0..1]	
	fdt:Ranges	O	[0..1]	
	fdt:statusFlag	O	[0..1]	
	fdt:StatusInformation	O	[0..1]	
DocumentVariables	STRUCT			Collection of document variables
	fdt:name	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	collection of	M	[1..1]	
	DocumentVariables		[0..*]	
	DocumentVariable		[0..*]	
	GraphicReference		[0..*]	
DTMSpecificXMLData	STRUCT			Optional additional information which shall be described by a private style
DTMStyleForCompleteDocument	STRUCT			Optional style information which has to be provided by a DTM if it returns documents which cannot be described by the FDT standard style
GraphicReference	STRUCT			Reference to an object that has to be embedded within the document like bitmaps or other graphical elements
	title	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	path	M	[1..1]	

A.11 DeviceList data type

Namespace: devList

The simple data types (see Table A.18) and structured data types (see Table A.19) defined in this clause are used as a base for definition of service specific data types or as service arguments.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.18 – Simple deviceList data type

Data type	Definition	Description
errorDescription	STRING	Detailed error information in case of dtmSpecificError
errorInfo	enumeration (ok | failedToSet | failedDuplicateAddress | cancelled | dtmSpecificError)	To be used when DTMDeviceListSchema is returned to indicate error summary if used as an attribute of DeviceList element and error information for a specific address when used in Device element. Enumeration: • "ok" Address was set successfully – used in DeviceList as well as in Device • "failedToSet" – used in Device to indicate failed NetworkManagementInfoWrite. • "failedDuplicateAddress" - used to indicate that the address is already available and could not be set. • "cancelled" – used to indicate that the setting was cancelled by user. • "dtmSpecificError" – indicates a DTM specific error. In this case, error description text is to be used to give more detailed error information
showUserInterface	enumeration (openUserInterface | noUserInterface | setNextValidAddress)	Indicates if the Communication Channel should open a user interface in order to get a protocol specific address selection by decision of the user. Enumeration: • openUserInterface – user interface should be opened to request the address from user • oUserInterface – no user interface should be opened • setNextValidAddress - Communication Channel has to set the next valid address without using a user interface. In this case, the busAddress is not used by Communication Channel. If this attribute is not set, NoUserInterface is assumed

Table A.19 – Structured deviceList data type

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
BusAddress	STRUCT			Contains a single busAddress
	fdt:busAddress	M	[1..1]	
BusAddressRange	STRUCT			Bus address range. The structure data type defines a start and an end bus address (for example used in ScanRequest)
	BusAddress	M	[1..*]	
Device	STRUCT			Contains device identification information and system tag of corresponding DTM
	fdt:systemTag	M	[1..1]	
	errorInfo	O	[0..1]	
	errorDescription	O	[0..1]	
	BusAddressRange	M	[1..1]	
DeviceList	STRUCT			List of DTM system tags and corresponding device addresses to set
	errorInfo	O	[0..1]	
	errorDescription	O	[0..1]	
	showUserInterface	O	[0..1]	
	Device	M	[1..*]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

A.12 Network management data types

Namespace: net

The simple data types (see Table A.20) and structured data types (see Table A.21) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.20 – Simple network management data types

Data type	Definition	Description
busAddress	String	Protocol specific address of device
configurationData	ARRAY OF USINT	Protocol specific configuration data as binary stream according to the fieldbus-specification
moduleId	UDINT	Unique identifier for a module within the name space of the device instance
moduleTypeId	UDINT	Unique identifier for a module type within the name space of the device type
redundant	BOOL	Specifies whether a device or module is redundant
slot	UDINT	Unique identifier for the slot of a module within the name space of the device instance

Table A.21 – Structured network management data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DeviceAddress	STRUCT			Protocol specific address of device
	fdt:busAddress	M	[1..1]	
	configurationData	O	[0..1]	
UserDefinedBus	STRUCT			Protocol specific part of NetworkInfo, is defined within the IEC 62453-3xy documents
	<protocol specific>			
NetworkInfo	STRUCT			Description of network configuration of a device instance
	fdt:protocolId	M	[1..1]	
	fdt:BusRedundancy	O	[0..1]	
	choice of	O	[0..*]	
	DeviceAddress	S	[1..1]	
	UserDefinedBus	S	[1..1]	
DtmDeviceInstanceTopology	STRUCT			Description of internal topology of a DtmDeviceType
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	
	NetworkInfo	O	[0..1]	
	fdt:ChannelReferences	O	[0..1]	
	InternalChannel	M	[1..*]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
InternalChannel	STRUCT			An internal channel is the connection point for an internal module within the internal topology
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	
	Module	O	[0..*]	
Module	STRUCT			Description of a hardware or software module of a device
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	
	moduleId	M	[1..1]	
	moduleTypeId	O	[0..1]	
	slot	O	[0..1]	
	redundant	O	[0..1]	
	configurationData	O	[0..1]	
	fdt:VersionInformation	M	[1..1]	
	fdt:ChannelReferences	O	[0..1]	

A.13 Instance data types

Namespace: param

The simple data types (see Table A.22) and structured data types (see Table A.23) defined in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.22 – Simple instance data types

Data type	Definition	Description
itemErrorDescription	enumeration (dtmSpecific | noLock | notLongerValid | outOfResources | invalidValue)	Enumeration describing an error: • dtmSpecific: DTM specific error; • noLock: instance data set could not be locked; • notLongerValid: the item is not longer valid. This may happen due to configuration changes; • outOfResources: The DTM has no resources to perform the request. This may happen if the DTM can not queue the request; • invalidValue: The requested value is invalid
itemId	STRING	Unique id of an item

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
itemKind	enumeration (alarm | analogInput | analogOutput | computation | contained | correction | device | diagnostic | digitalInput | digitalOutput | discrete | discreteInput | discreteOutput | dynamic | frequency | frequencyInput | frequencyOutput | hart | input | local | localDisplay | operate | output | sensorCorrection | service | tune |others)	Identification of the item context. This is some kind of classification regarding the type of value. Enumerations. In the following each entry is explained: • alarm - contains alarm limits; • analogInput - is part of an analog input block; • analogOutput - is part of an analog output block; • computation - is part of a computation block; • contained - represents the physical characteristics of the device; • correction - is part of the correction block; • device - represents the physical characteristics of the device; • diagnostic - indicates the device status; • digitalInput - is part of a digital input block; • digitalOutput - is part of a digital output block; • discrete – is part of a discrete block; • discreteInput - is part of a discrete input block; • discreteOutput - is part of a discrete output block; • dynamic - is modified by the device without stimulus from the network; • frequency – is part of frequency block; • frequencyInput - is part of a frequency input block; • frequencyOutput - is part of a frequency output block; • hart – is part of HART block; • input - is part of an input block. An input block is a special kind of computation block which does unit conversions, scaling, and damping. The parameter of the input block parameters can be determined by the output of another block; • local - is locally used by the an application. Local items are not stored in a device, but they can be sent to a device; • localDisplay - is part of the local display block. A local display block contains the items associated with the local interface (keyboard, display, etc.) of the device, • operate - is used to control a block's operation output - is part of the output block. The values of output items may be accessed by another block input; • sensorCorrection – is part of sensor correction block; • service - is used when performing routine maintenance; • tune - is used to tune the algorithm of a block; • others - is used if all other entries do not match
itemType	enumeration (standard | specific)	Information whether the item follows the semantics defined via the general rules defined for a specific protocol (standard) or a DTM/vendor specific semantics (specific)
label	STRING	Human readable name
limitBits	enumeration (none | low | high | constant)	Limit status of the item. NOTE Additional information may be found in the OPC XML-DA Specification Version 1.0
moduleName	STRING	This attribute contains the name of the module

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition	Description
qualityBits	enumeration (bad | badConfigurationError | badNotConnected | badDeviceFailure | badSensorFailure | badLastKnownValue | badCommFailure | badOutOfService | badWaitingForInitialData | uncertain | uncertainLastUsableValue | uncertainSensorNotAccurate | uncertainEUExceeded | uncertainSubNormal | good | goodLocalOverride)	Quality status of the item. NOTE Additional information may be found in the OPC XML-DA Specification Version 1.0

Table A.23 – Structured instance data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DtmItem	STRUCT			Contains the value of a parameter or a process value and some additional optional information like time stamp
	fdt:id	M	[1..1]	
	TimeStamp	M	[1..1]	
	Quality	M	[1..1]	
	choice of	M	[1..1]	
	fdt:Variant	S	[1..1]	
	ItemError	S	[1..1]	
DtmItemInfo	STRUCT			Describes a parameter or a process value that is available via the Services to access the instance data of a DTM. The information contains descriptive attributes like name as well as information how the item is accessible. The relation between item information and the item itself is realized via ItemId
	fdt:id	M	[1..1]	
	fdt:SemanticInformation	M	[1..*]	The DTM shall provide a <SemanticInformation> element for all supported fieldbus protocol of the DTM instance
	fdt:name	M	[1..1]	
	fdt:dataType	M	[1..1]	
	itemType	M	[1..1]	
	fdt:descriptor	O	[0..1]	
	moduleName	O	[0..1]	
	fdt:readAccess	O	[0..1]	
	fdt:writeAccess	O	[0..1]	
	label	O	[0..1]	
	ItemKind	M	[1..*]	
	UnitDescription	O	[0..1]	
	choice of	M	[1..1]	
	RangeDescriptions	S	[0..1]	
	ValueDescription	S	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DtmItemInfoGroup	STRUCT			List of DTM item information
	fdt:name	M	[1..1]	
	fdt:semanticId	M	[1..1]	
	label	O	[0..1]	
	DtmItemInfo	O	[0..*]	
	DtmItemInfoGroup	O	[0..*]	
DtmItemInfoList	STRUCT			List of DTM item information and/or a list of item information groups
	DtmItemInfo	O	[0..*]	
	DtmItemInfoGroup	O	[0..*]	
DtmItemList	STRUCT			List of DTM items
	DtmItem	M	[1..*]	
DtmItemSelection	STRUCT			Items selected for execution of a service (or similar)
	fdt:id	M	[1..1]	
DtmItemSelectionList	STRUCT			List of selected items
	DtmItemSelection	M	[1..*]	
ItemError	STRUCT			Communication error or other error
	choice of	O	[0..1]	
	ItemErrorDescription	S	[1..1]	
	fdt:CommunicationError	S	[1..1]	
ItemErrorDescription	STRUCT			Description of non-communication error
	itemErrorDescription	M	[1..1]	
	fdt:descriptor	O	[0..1]	
ItemKind	STRUCT			Identification of the item context
	itemKind	M	[1..1]	
ItemReference	STRUCT			Reference to an other item within the xml document
	fdt:idref	O	[0..1]	
LowerRange-Description	STRUCT			Description of the lower range
	choice of	O	[0..1]	
	ItemReference	S	[1..1]	
	fdt:StringData	S	[1..1]	
	fdt:NumberData	S	[1..1]	
	fdt:TimeData	S	[1..1]	
PossibleEnumerations	STRUCT			Possible enumerations of an item
	fdt:EnumeratorEntries	M	[1..1]	
Quality	STRUCT			Description of the quality of the item. For write requests: The Frame Application may define a Quality In case the underlying device/parameter supports the quality definitions, the value+quality should be handed over by DTM to the device, otherwise the DTM should only process values with good quality
	qualityBits	M	[1..1]	
	limitBits	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
RangeDescription	STRUCT			Description of an range
	LowerRangeDescription	M	[1..1]	
	UpperRangeDescription	M	[1..1]	
	fdt:LowerRawValue	O	[0..1]	
	fdt:UpperRawValue	O	[0..1]	
RangeDescriptions	STRUCT			Description of the ranges of the item
	RangeDescription	M	[1..*]	
TimeStamp	STRUCT			Description of the time stamp of the item
	fdt:time	M	[1..1]	
UnitDescription	STRUCT			Description of the unit of the item
	choice of	O	[0..1]	
	ItemReference	S	[1..1]	
	fdt:EnumeratorEntry	S	[1..1]	
UpperRangeDescription	STRUCT			Description of the upper range
	choice of	O	[0..1]	
	ItemReference	S	[1..1]	
	fdt:StringData	S	[1..1]	
	fdt:NumberData	S	[1..1]	
	fdt:TimeData	S	[1..1]	
ValueDescription	STRUCT			Description the value of the item
	PossibleEnumerations	M	[1..1]	

A.14 DeviceStatus data types

Namespace: status

The simple data types (see Table A.24) and structured data types (see Table A.25) define in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.24 – Simple device status data types

Data type	Definition	Description
deviceInitiatedCommunication	BOOL	Device is currently using device initiated communication

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.25 – Structured device status data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DtmDeviceStatus	STRUCT			Description of the current status of a device
	fdt:statusFlag	M	[1..1]	
	deviceInitiatedCommunication	O	[0..1]	
	fdt:StatusInformation	O	[0..1]	

A.15 OnlineCompare data types

Namespace: onlineComp

The simple data types (see Table A.26) and structured data types (see Table A.27) define in this clause are used as a base for definition of service specific data types or as service arguments.

Table A.26 – Simple online compare data types

Data type	Definition	Description
statusFlag	enumeration (equal | notEqual | noComparableData)	Describes whether the data are equal or not.

Table A.27 – Structured online compare data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
DTMOnlineCompare	STRUCT			Contains the compare result or a communication error
	statusFlag	M	[1..1]	
	fdt:StatusInformation	O	[0..1]	

A.16 UserInterface data types

Namespace: ui

The simple data types (see Table A.28) and structured data types (see Table A.29) defined in this clause are used as a base for definition of service specific data types or as service arguments.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

Table A.28 – Simple user interface data types

Data type	Definition	Description
helpContext	INT	Help context (e.g. the reference number within the help file)
helpFile	STRING	Definition of the help file
messageButtons	enumeration (buttonsAbortRetryIgnore | buttonsOk | buttonsOkCancel | buttonsRetryCancel | buttonsYesNo | buttonsYesNoCancel)	Definition of the button types which shall appear within the message box
messageDefault	enumeration (buttonAbort | buttonRetry | buttonIgnore | buttonOk | buttonCancel | buttonYes | buttonNo)	Definition of the default button
messageType	enumeration (messageExclamation | messageInformation | messageQuestion | messageStop)	Type of a message like exclamation, information, …
resultMessage	enumeration (nobutton | buttonAbort | buttonRetry | buttonIgnore | buttonOk | buttonCancel | buttonYes | buttonNo)	Definition of the result of the user interaction
resultStatus	enumeration (notSupported | denied | systemResponse | ok)	
runAsModal	BOOL	Userinterface as modal dialog
title	STRING	

Table A.29 – Structured user interface data types

Data type	Definition			Description
	Elementary data types	Usage	Multiplicity	
TextLine	STRUCT			Definition of a single text line within the message
	fdt:string	M	[1..1]	
UserMessage	STRUCT			Definition of the whole message
	runAsModal	O	[0..1]	
	messageType	M	[1..1]	
	messageButtons	M	[1..1]	
	messageDefault	M	[1..1]	
	title	M	[1..1]	
	helpFile	O	[0..1]	
	helpContext	O	[0..1]	
	collection of	M	[1..1]	
	TextLine		[0..*]	
	fdt:DtmVariable		[0..*]	
	resultMessage	O	[0..1]	
	resultStatus	O	[0..1]	

LICENSED TO MECON Limited. - RANCHI/BANGALORE, FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

A.17 Fieldbus specific data types

Table A.30 shows protocol specific data types which shall be defined within an IEC 62453-3xy document describing protocol profile integration in FDT.

This namespace defines abstract data types that will be replaced by specific data types within the IEC 62453-3xy documents.

Namespace: fieldbus

Table A.30 – Fieldbus data types

Data type	Description
ChannelData	Protocol specific identification information for a channel (see 7.6.1.2)
ConnectRequest	Protocol specific information which is provided within a service Connect (see 7.6.1.2)
ConnectResponse	Protocol specific information which is provided within a service Connect (see 7.6.1.2)
DeviceTypeIdentification	Protocol specific identification information which can be transferred into devIdent:DeviceTypeIdentification
DeviceTypeIdentifications	Collection of DeviceTypeIdentification elements
DisconnectRequest	Protocol specific information which is provided within a service Disconnect (see 7.6.1.3)
DisconnectResponse	Protocol specific information which is provided within a service Disconnect (see 7.6.1.3)
ScanIdentification	Protocol specific identification information which can be transferred into devIdent:ScanIdentification
ScanIdentifications	Collection of ScanIdentification elements
SequenceDefine	Protocol specific information which is provided within a service SequenceBegin (see 7.6.1.7). A SequenceDefine data type typically contains multiple TransactionRequests and may contain additional protocolspecific information regarding the sequence execution
TransactionRequest	Protocol specific information which is provided within a service Transaction (see 7.6.1.6)
TransactionResponse	Protocol specific information which is provided within a service Transaction (see 7.6.1.6)

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

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

LICENSED TO MECON Limited. - RANCHI/BANGALORE,
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.