IEC 62453-1

Edition 1.0 2009-06

INTERNATIONAL STANDARD

Field device tool (FDT) interface specification –
Part 1: Overview and guidance

IEC 62453-1:2009(E)

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

About the IEC

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

About IEC publications

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

- 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 It also gives information on projects, withdrawn and replaced publications.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 62453-1

Edition 1.0 2009-06

INTERNATIONAL STANDARD

Field device tool (FDT) interface specification –
Part 1: Overview and guidance

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE W

ICS 25.040.40; 35.100.05; 35.110

ISBN 2-8318-1050-3

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 1: Overview and guidance

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 62453-1 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/123/FDIS	65E/136/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.

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.

INTRODUCTION

Enterprise automation requires two main data flows: a “vertical” data flow from enterprise level down to the field devices including signals and configuration data, and a “horizontal” communication between field devices operating on the same or different communication technologies.

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.

Several different manufacturer specific tools have to be used. The data in these tools are often invisible data islands from the viewpoint of system life-cycle management and plant-wide automation.

To ensure the consistent management of a plant-wide control and automation technology, it is necessary to fully integrate fieldbusses, devices and sub-systems as a seamless part of a wide range of automation tasks covering the whole automation life-cycle.

IEC 62453 provides an interface specification for developers of FDT (Field Device Tool) components to support function control and data access within a client/server architecture. The availability of this standard interface facilitates development of servers and clients by multiple manufacturers and supports open interoperation.

A device or module-specific software component, called a DTM (Device Type Manager) is supplied by a manufacturer with the related device type or software entity type. Each DTM can be integrated into engineering tools via defined FDT interfaces. This approach to integration is in general open for all fieldbusses and thus supports integration of different devices and software modules into heterogeneous control systems.

The IEC 62453 common application interface supports the interests of application developers, system integrators, and manufacturers of field devices and network components. It also simplifies procurement, reduces system costs and helps manage the lifecycle. Significant savings are available in operating, engineering and maintaining the control systems.

The objectives of IEC 62453 series are to support:

- universal plant-wide tools for life-cycle management of heterogeneous fieldbus environments, multi-manufacturer devices, function blocks and modular sub-systems for all automation domains (e.g. process automation, factory automation and similar monitoring and control applications);
- integrated and consistent life-cycle data exchange within a control system including its fieldbuses, devices, function blocks and modular sub-systems;
- simple and powerful manufacturer-independent integration of different automation devices, function blocks and modular sub-systems into the life-cycle management tools of a control system.

The FDT concept supports planning and integration of monitoring and control applications, it does not provide a solution for other engineering tasks such as "electrical wiring planning”, “mechanical planning”. Plant management subjects such as "maintenance planning”, “control optimization”, “data archiving”, are not part of this FDT standard. Some of these aspects may be included in future editions of FDT publications.

FIELD DEVICE TOOL (FDT) INTERFACE SPECIFICATION –

Part 1: Overview and guidance

1 Scope

This part of IEC 62453 presents an overview and guidance for the IEC 62453 series. It

- explains the structure and content of the IEC 62453 series (see Clause 5);
- provides explanations of some aspects of the IEC 62453 series that are common to many of the parts of the series;
- describes the relationship to some other standards.

2 Normative references

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

IEC 61158 (all parts)*, Industrial communication networks – Fieldbus specifications*

IEC 61784 (all parts), *Industrial communication networks – Profiles*

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

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Terms and definitions

3.1.1
actor
coherent set of roles that users of use cases play when interacting with these use cases

[ISO/IEC 19501]

NOTE An actor has one role for each use case with which it communicates.

3.1.2
address
communication protocol specific access identifier

3.1.3
application
software functional unit that is specific to the solution of a problem in industrial-process measurement and control

NOTE An application may be distributed among resources, and may communicate with other applications.

3.1.4
business object
object representing specific behavior (e.g. DTM, BTM and channel)

NOTE The term business object has been defined originally as part of the design pattern 3-tier architecture, where the business object is part of the business layer.

3.1.5
Block Type Manager (BTM)
specialized DTM to manage and handle a block

3.1.6
communication
fieldbus protocol specific data transfer

3.1.7
Communication Channel
access point for communication to field device

3.1.8
configuration
system created by configuring the plant components and the topology

3.1.9
configure
(see also parameterize)
setting parameters at the instance data as well as the logical association of plant components to build up the plant topology (off-line)

3.1.10
connection
established data path for communication with an selected device

3.1.11
data
set of parameter values

3.1.12
data type
set of values together with a set of permitted operations

[ISO 2382 series]

3.1.13
DCS manufacturer / system manufacturer
manufacturer of the engineering system

3.1.14
device
(see also field device)

a) networked independent physical entity of an industrial automation system capable of performing specified functions in a particular context and delimited by its interfaces [IEC 61499-1]

b) entity that performs control, actuating and/or sensing functions and interfaces to other such entities within an automation system

3.1.15
device manufacturer
manufacturer of fieldbus devices

3.1.16
device type
device characterization based on abstract properties such as manufacturer, fieldbus protocol, device type identifier, device classification, version information or other information

NOTE The scope of such characterizations can vary depending on the properties that are used in the definition of such a set and is manufacturer specific for each DTM.

3.1.17
distributed system
FDT objects that jointly are executed on different PCs in a network

NOTE The implementation of such a distributed system is vendor specific (for example: DTM and Presentation are executed on different PCs or DTMs are executed in multi-user system on different PCs)

3.1.18
documentation
human readable information about a device instance

NOTE This may be electronic information in a database.

3.1.19
Device Type Manager (DTM)
a) software component containing device specific application software
b) generic class and means "Type Manager"

NOTE The D is kept because the Acronym is well-known in the market.

3.1.20
DTM device type
software module for a particular device type within the DTM

NOTE A DTM may contain one or more DTM device types

3.1.21
entity
particular thing, such as a person, place, process, object, concept, association, or event

[IEC 61499-1]

3.1.22
field device
(see also device)

3.1.23
Frame Application
FDT runtime environment

3.1.24
FDT model
interface specification for objects and object behavior in a monitoring and control system

3.1.25
function
specific purpose of an entity or its characteristic action

[IEC 61499-1]

3.1.26
hardware
physical equipment, as opposed to programs, procedures, rules and associated documentation
[ISO/AFNOR Dictionary of computer science]

3.1.27
implementation
development phase in which the hardware and software of a system become operational

[IEC 61499-1]

3.1.28
instantiation
creation of an instance of a specified type

[IEC 61499-1]

3.1.29
interface
shared boundary between two *functional units,* defined by functional characteristics, signal characteristics, or other characteristics as appropriate

[IEC 60050-351]

3.1.30
mapping
set of values having defined correspondence with the quantities or values of another set
[ISO/AFNOR Dictionary of computer science]

3.1.31
multi-user environment
environment which allows operation by more than one user

3.1.32
network
all of the media, connectors, repeaters, routers, gateways and associated node communication elements by which a given set of communicating devices are interconnected

[IEC 61158-5-X]

NOTE In this document network is used to express that one or more interconnected fieldbus systems with different protocols can be applied.

3.1.33
nested communication
communication using a hierarchy of communication systems

3.1.34
operation
well-defined action that, when applied to any permissible combination of known entities, produces a new entity
[ISO/AFNOR Dictionary of computer science]

3.1.35
parameter
variable that is given a constant value for a specified application and that may denote the application
[ISO/AFNOR Dictionary of computer science]

3.1.36
parameterize
(see also configure)
setting parameters in a device or a block or an object

3.1.37
persistent data
stored data that is preserved through shut down/restart and maintenance activities

3.1.38
Process Channel
representation of process value and its properties

3.1.39
service
functional capability of a resource, which can be modeled by a sequence of service primitives

[IEC 61499-1]

3.1.40
session
instance of user interactions within the FDT model

3.1.41
synchronization
synchronization of data depending on the context where used

NOTE For example, synchronization can occur between the DTM and device or between several DTM instances having a reference to the same instance data.

3.1.42
system
set of interrelated elements considered in a defined context as a whole and separated from its environment

[IEC 60050-351]

NOTE 1 Such elements may be both material objects and concepts as well as the results thereof (e.g. forms of organization, mathematical methods, and programming languages).

NOTE 2 The system is considered to be separated from the environment and other external systems by an imaginary surface, which can cut the links between them and the considered system.

3.1.43
transient data
temporary data which have not been stored (while configuring or parameterizing)

3.1.44
type
software element, which specifies the common attributes shared by all instances of the type

[IEC 61499-1]

3.1.45
variable
software entity that may take different values, one at a time

[IEC 61499-1]

NOTE 1 The values of a variable are usually restricted to a certain data type.

NOTE 2 Variables are described as input variables, output variables, and internal variables.

3.1.46
use case
specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system

[ISO/IEC 19501]

3.2 Abbreviations

BTM	Block Type Manager
COM	Component Object Model
CP	Communication profile
CPF	Communication profile family
DCS	Distributed control system
DD	Device description
DTM	Device Type Manager
ERP	Enterprise resource planning
FA	Frame Application
FB	Function block
FDT	Field device tool
GUI	Graphical user interface
ID	Identifier
IDL	Interface definition language
I/O	Input/output
IT	Information technology
MES	Manufacturing execution systems
OEM	Original equipment manufacturer
OLE	Object Linking and Embedding
OPC	Open connectivity via open standards (originally: OLE for Process Control)
PC	Personal computer
PLC	Programmable logic controller
SCADA	Supervisory, control and data acquisition
UML	Unified modeling language
UUID	Universal unique identifier
XML	Extensible markup language

3.3 Conventions

The conventions for UML notation used in the IEC 62453 series is defined in Annex A.

4 FDT overview

4.1 State of the art

In industrial automation, a control system often comprises many binary and analog input/output signals transmitted via a communication network. Numerous field devices provided by different manufacturers have to be included in the network by direct connection or I/O multiplex units. Many applications use more than 100 different field device types from various device manufacturers.

Each device has specific configuration and parameterization functions to support its designed task. These device-specific properties and settings have to be taken into consideration when configuring a fieldbus coupler and bus communication for the device. The device presence and its capability have to be made known to the control system. Device input and output signals and function block services need to be effectively integrated into the planning of the control system.

In the absence of a common interface standard, the large number of different device types and suppliers within a control system project makes the configuration task difficult and time-consuming. Various different tools have to be used (see Figure 1). The user requirement for consistency of data, documentation and application configurations can only be guaranteed by intensive and costly system testing.

A common location for service and diagnostic tasks in the control system does not fully cover the functional capabilities of available fieldbus devices nor does it guarantee that different device or module-specific tools can be integrated into other system software tools. Typically, device-specific tools can only be connected directly to a specific fieldbus or directly to a specific field device type.

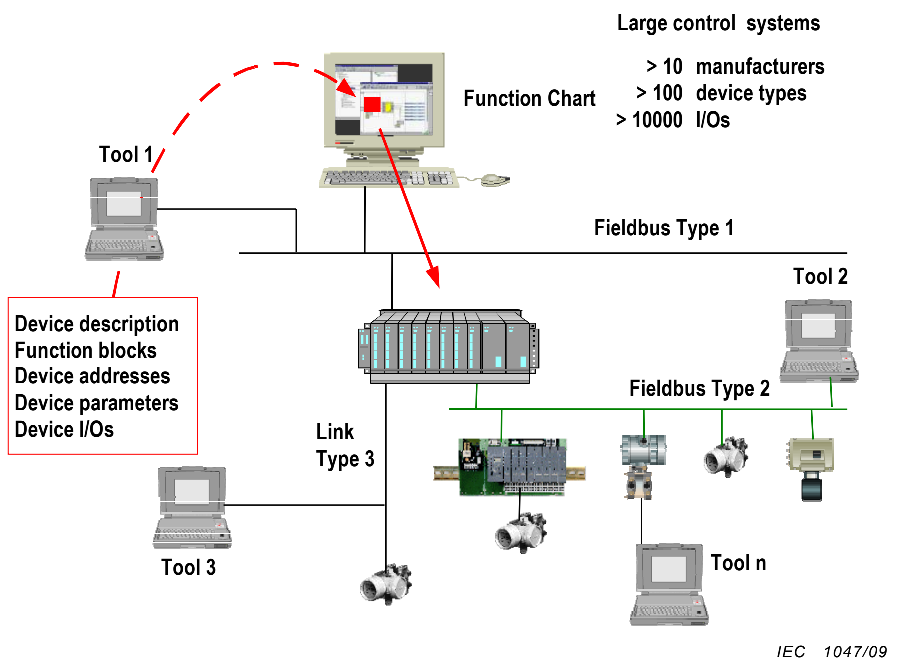


Figure 1 – Different tools and fieldbusses result in limited integration

4.2 Objectives of FDT

4.2.1 General features

Full integration of fieldbus devices or modules into automation systems requires a communication path from central engineering or operator terminals via the system and fieldbusses to the individual field devices.

FDT supports:

- central facilities for planning, diagnostics and service with direct access to all devices;
- integrated, consistent configuration and documentation of the automation system, its fieldbusses and devices;
- organization of common data for the automation system and the field devices;

- central data management and data security;
- simple, fast integration of different device and module types into the automation system.

Integration of field devices into the engineering systems of automation technology can cover a small set of configuration, service and diagnostic functions as well as large set of functions.

4.2.2 Device and module manufacturer benefits

FDT technology allows integration of individual device and module properties, including specific characteristics and special features for different device and module types. Planning and service tools provided by the manufacturer can be integrated as device or module-specific software components into the engineering system. The manufacturer is able to define the configuration, service and diagnostic functions and also to design the appearance of devices and modules in the engineering environment of the automation system.

This reduces the costs for the manufacturer, as one standardized software component is able to support configuration, service and diagnostic functions for the device in any automation system. It also eliminates frequent project-specific or control system-specific adaptations, which have to be developed and maintained for multiple device and module types in the absence of a standard.

4.2.3 System manufacturer and integrator benefits

The control system manufacturer or integrator has to implement the defined interfaces for the integration of all fieldbus devices and modules only once. Manufacturer-specific and/or device-specific implementations and their maintenance are eliminated.

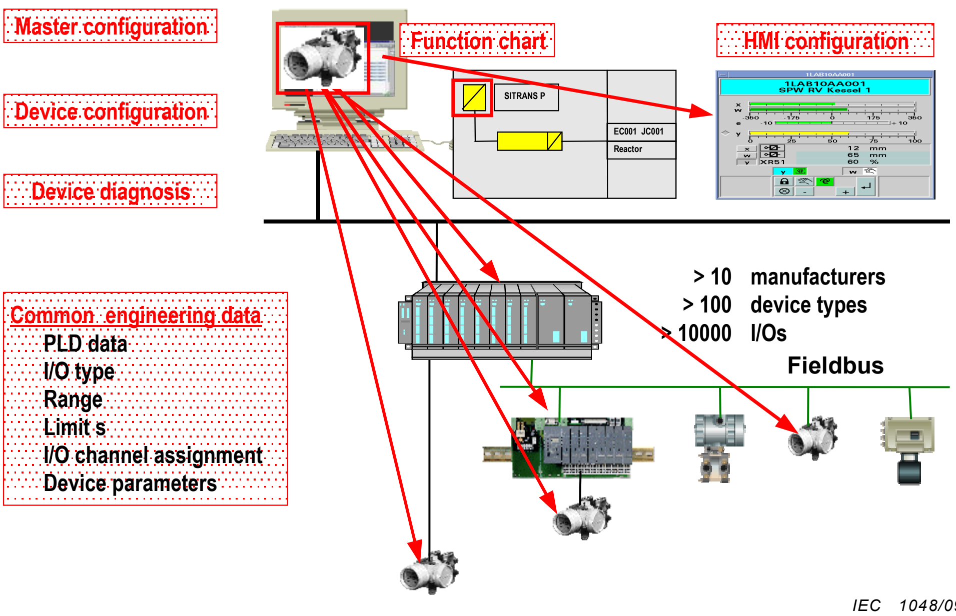


Figure 2 – Full integration of all devices and modules into a homogeneous system

4.2.4 Other applications

Although FDT is primarily designed to control device functionality and for accessing data to configure parts of the control system, FDT interfaces can be used in many places within an application. At the lowest level, they can get raw data from the devices and modules in a SCADA, DCS or PLC to configure the bus master. At a higher level the Frame Application can start device specific diagnosis applications via the DTM. The architecture and design makes it possible to build and to integrate scalable DTMs, where the functionality depends on the capabilities of the device.

4.3 FDT model

4.3.1 General

FDT facilitates the interaction between device-specific software components, fieldbus interface-specific software components and host systems (see Figure 3).

- The device-specific software components are called Device Type Managers (DTMs).
- The fieldbus interface-specific software components are called Communication Channels
- The host systems are called Frame Applications.

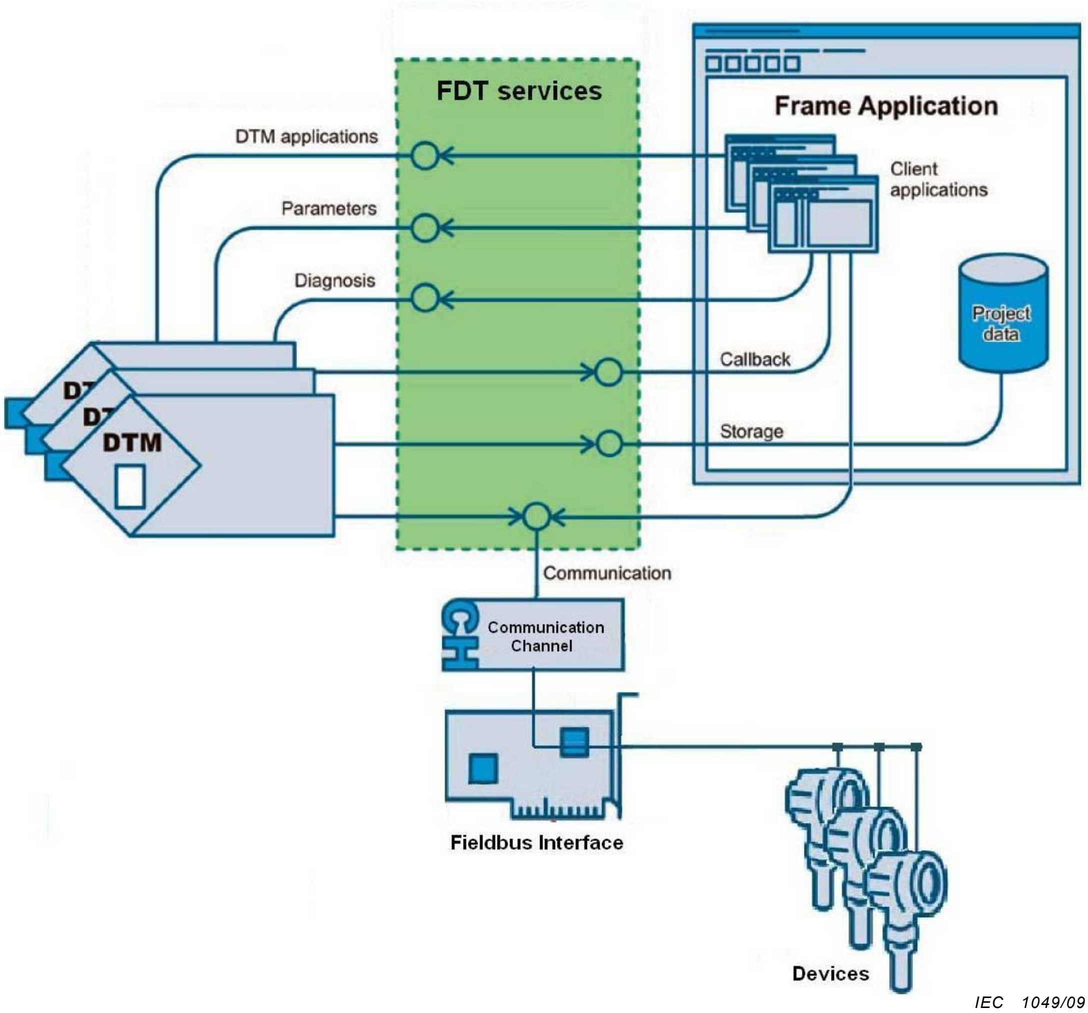


Figure 3 – General architecture and components

A Frame Application provides the runtime environment for the DTMs. Typically, the Frame Application comprises client applications that use DTMs, a database for persistent storage of device data, and a Communication Channel to the field devices. Client applications are single applications focusing on specific aspects such as configuration, observation, channel assignment, and using the service provided by the DTMs.

A DTM encapsulates device-specific software applications and protocol specific definitions. Thus a Frame Application is able to handle any type of device by integrating corresponding DTMs without the need for device or fieldbus specific knowledge.

Device and module-specific software components are of several types to cover integration of products commonly used in automation systems. DTM objects can, for example, represent normal measurement and control devices. There are specializations of DTMs which represent:

- software entities or function blocks that are movable and may be hosted by different modules in a network, also known as Block DTM (BTM) objects;
- modular equipment combinations, such as I/O stations with plug in boards to provide combinations of I/O and control functions also known as Module DTM objects.

A Communication Channel represents the entry point to a fieldbus or point-to-point communication. It provides fieldbus interface independent services. In general, the protocol specific services are 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.).

FDT defines the services each of the components has to provide and the data which is exchanged by those. The services are defined in a fieldbus independent way, but some exchange fieldbus specific data. The content and data format for manufacturer independent fieldbuses is defined in the IEC 62453-3xy specifications defining the protocol profile integration in FDT. FDT also enables manufacturers to define their own content and data formats, for example for a manufacturer specific fieldbus or point-to-point communication.

The main features of the FDT concept are:

- Frame Application is the representation of the host tool where the DTMs are interacting with the control system, maintenance or engineering application;
- DTM is the main concept and can be applied for simple devices but also for modular devices, software components (function block);
- nested communication is provided to meet the requirement of heterogeneous and hierarchical networks where the intelligent field devices are connected. This is the background for various communication parts in the standard;
- graphical interfaces are provided to provide interactive access to the functionality of the intelligent field devices and its DTM to the human beings. These aspects are represented by so called presentation objects.

4.3.2 Frame Applications

Frame Applications provide the runtime environment for the FDT system. Depending on the intended use Frame Applications may have different appearances (e.g. standalone configuration and engineering system). The following general requirements apply to all Frame Applications:

- device and module-specific knowledge is not necessary;
- ability to manage all DTM instances and store instance data;
- ability to manage and create DTM communications and connections (including any necessary message routing);
- guarantee of system-wide consistent configuration;
- enables multi-user and server/client operation (optional);
- takes care of data versions and consistency.

FDT is a specification to facilitate the interaction between device-specific objects (i.e. presentation and DTM) and the Frame Application. This is shown in Figure 4.

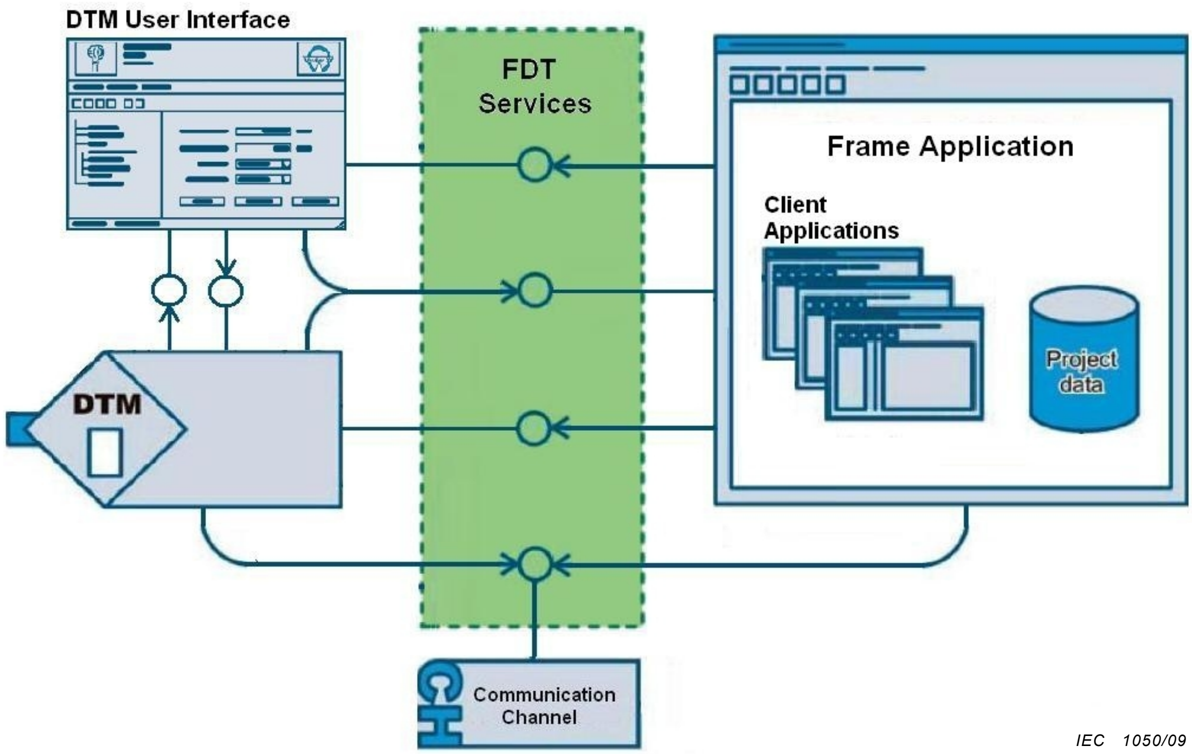


Figure 4 – FDT software architecture

A Frame Application provides the runtime environment for device specific objects. Typically, the Frame Application comprises client applications that use DTMs, a database for persistent storage of device data, and a communication link to the field devices.

Client applications are single applications focusing on specific aspects such as configuration, observation, channel assignment, and using the functionality provided by the DTM as a server.

NOTE The IEC 62453 series distinguishes between the specification of the interactions between objects (e.g. Frame Applications and DTMs) and the implementation technology for the implementation of those objects. It specifies only the behavior that the objects are expected to provide to client applications that use them.

4.3.3 Device Type Manager

DTM objects are supplied by the device manufacturer together with the device. The following properties are characteristic for the DTM:

- generally not a standalone tool;
- graphical user interfaces as defined by this specification;
- all rules of the device are known;
- all user dialogs are contained;
- user interface (multilingual including help system);
- parameter validity check (also depending on other device-specific parameters);
- automatic generation of dependent parameters;
- processing sequences are defined for complex calibration, and setup procedures where needed;
- reading and writing of parameters from/to the field device;

- diagnostic functions customized for the device;
- provision of the type-specific data for establishment of communication;
- provision of device/instance-specific data, for example to be used in function planning;
- device or instance specific documentation;
- no direct connection to any other device;
- no information on the engineering environment;
- support for one or more device types.

Each device manufacturer chooses the number and range of functions for a DTM, as required by the functional capabilities of the device(s) it supports. A DTM covers at least one field device. DTMs can, however, also cover device families (for example, pressure transmitters) or the entire product range of a manufacturer. Communication (via the various bus systems of a control system) and data management are handled via the interfaces of the engineering tool.

Within the framework of overall system planning or plant management, a DTM must always be integrated into the appropriate engineering tool. To avoid data consistency problems, parallel access by standalone tools and the system's engineering tool to access the same devices are not recommended. Parallel standalone operation may be implemented in special cases for example when migrating from a standalone tool to a DTM.

A DTM is installed as a component of an engineering tool or any other application that manages the device instances, and provides the communication mechanisms and supports the component with device-specific tasks. In the following, those applications are referred to as 'Frame Applications'.

4.3.4 Communication Channel concept

The IEC 62453 series specifies the FDT objects and their interfaces which support interaction between a Frame Application and a device or module-specific application called a DTM. Frame applications can be engineering tools, operator stations or standalone tools. The Frame Applications act as clients and the DTMs act as servers.

DTMs can be connected to monolithic Frame Applications or to distributed Frame Applications based on different components provided by one or more manufacturers. Frame Applications can manage and integrate multiple DTMs provided by one or more manufacturers (see Figure 5).

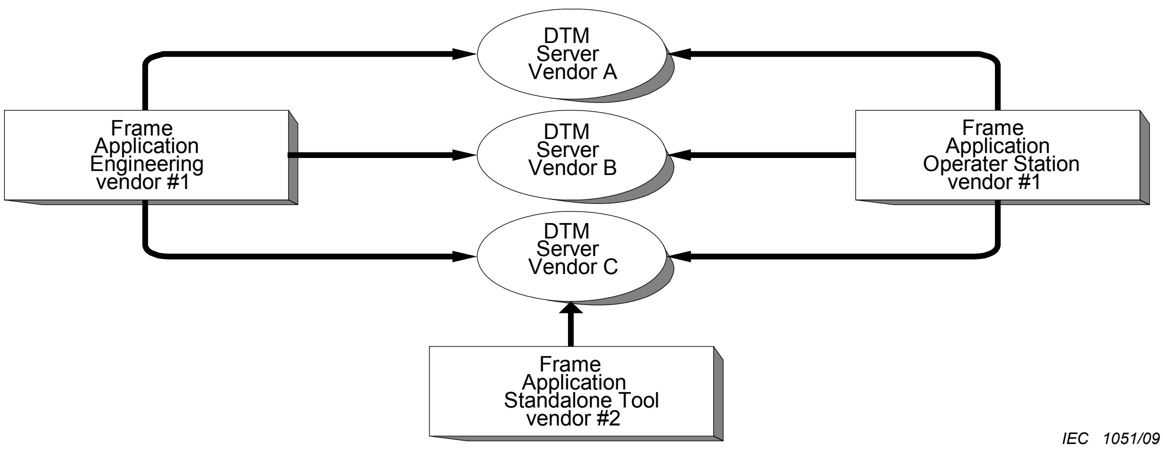


Figure 5 – General FDT client/server relationship

DTMs are composed of a server object and accompanying objects, this represents the device and block as viewed from an external application. The server object represents the accessible set of device and block functionality.

There are two different types of channels, the Communication Channel and the Process Channel.

The Communication Channel object provides access to the fieldbus and is communication technology dependent.

The process channel provides information about the I/O values of a device or module (e.g. access, units, range and label). This information is needed for the client application to provide the correct context of the parameter for visualization.

The DTM representing the device, block or module behavior uses the Communication Channel for data transaction (i.e. DTM are communication clients). Assuming that a device is directly connected to a fieldbus the FDT object hierarchy is as shown in Figure 6 at the right side. The DTM communicates to the device using the Communication Channel. If the device is plugged to a communication hierarchy (Figure 6 left side) this hierarchy is represented at the software side using according FDT objects. Each fieldbus is represented by a Communication Channel and the crossover between the fieldbusses by according objects, in this example a gateway DTM. The device DTM does not recognize the underlying communication hierarchy. According to this model, the device DTM maintains the functionality of the device and the used Communication Channel object supports external communication.

The DTM provides the Process Channel for semantic information about the I/O parameters (i.e. DTM are information servers). The DTM provides the semantic information to the application using the Process Channel.

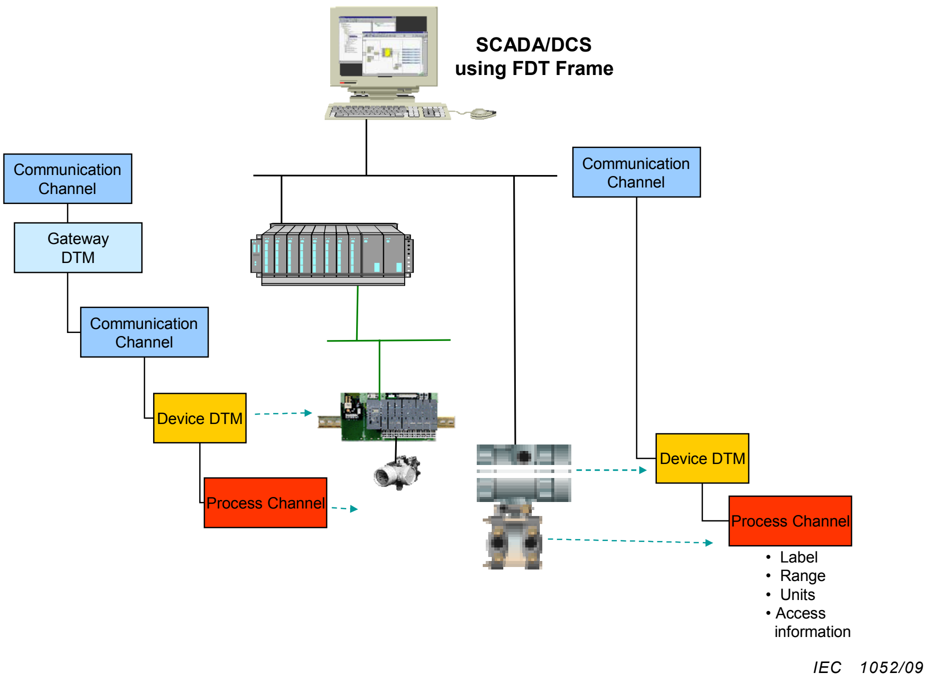


Figure 6 – Typical FDT channel architecture

Each Process Channel provides information to access and interpret about device specific I/O values (e.g. data types, ranges, alarms, etc), (see Figure 7).

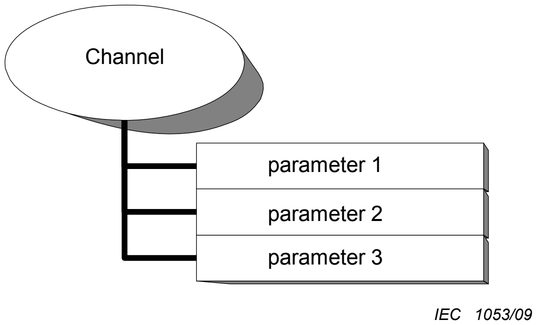


Figure 7 – Channel/parameter relationship

These I/O value related parameters contain all information a Frame Application needs for the integration of the I/O data of a device. In general, Process Channels carrying the parameters to describe the I/O data are not the data sources - they are just representations for them. These parameters should be thought of as simply specifying the address of the data, not as the actual source of the data that the address references.

4.3.5 Presentation object

The presentation object provides optional a graphical user interface for the DTM, BTM and Communication Channel. The presentation object can be integrated into the GUI of the Frame Application or it can be an object that runs outside of the GUI of the Frame Application (e.g. a standalone executable).

5 Structure of the IEC 62453 series

5.1 Structure overview

In addition to this overview, the FDT interface specification has several parts as shown in Figure 8.

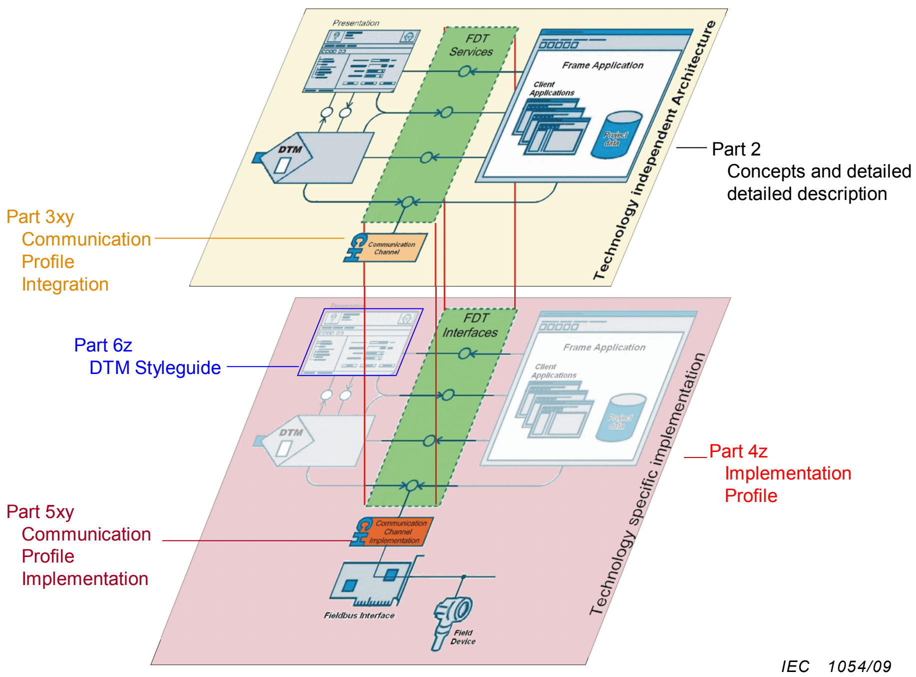


Figure 8 – Structure of the IEC 62453 series

The FDT interface specification describes interfaces between device specific objects (DTM and presentation object), client applications (within the Frame Application) and the FDT channel (communication functions). The interfaces are based on services, which carry the commands and information required by the client server interactions.

The IEC 62453 series is structured to separate the objects, client applications and channel service specifications from their mapping to any specific implementation technology.

Separate parts of the IEC 62453 series are used as follows:

- concepts and details of objects and services are specified in Part 2;
- communication profile integration for different communication services are specified in parts 3xy;
- object model integration profiles implementation technologies are specified in parts 4x;
- communication implementation for common object model for different communication services are specified in parts 5xy.

The IEC 62453 series includes the following parts:

Part 1:	*Overview and guidance*
Part 2:	*Concepts and detailed description*
Part 301:	*Communication profile integration – IEC 61784 CPF 1*
Part 302:	*Communication profile integration – IEC 61784 CPF 2*
Part 303-1:	*Communication profile integration – IEC 61784 CP 3/1 and CP 3/2*
Part 303-2:	*Communication profile integration – IEC 61784 CP 3/4, CP 3/5 and CP 3/6*
Part 306:	*Communication profile integration – IEC 61784 CPF 6*
Part 309:	*Communication profile integration – IEC 61784 CPF 9*
Part 315:	*Communication profile integration – IEC 61784 CPF 15*
Part 41:	*Object model integration profile – Common object model*
Part 501:	*Communication implementation for common object model – IEC 61784 CPF 1*
Part 502:	*Communication implementation for common object model – IEC 61784 CPF 2*
Part 503-1:	*Communication implementation for common object model – IEC 61784 CP 3/1 and CP 3/2*
Part 503-2:	*Communication implementation for common object model – IEC 61784 CP 3/4, CP 3/5 and CP 3/6*
Part 506:	*Communication implementation for common object model – IEC 61784 CPF 6*
Part 509:	*Communication implementation for common object model – IEC 61784 CPF 9*
Part 515:	*Communication implementation for common object model – IEC 61784 CPF 15*
Part 61:	*Device Type Manager (DTM) Styleguide for common object model*

The remainder of Clause 5 describes the principle content of the parts.

5.2 Part 2 – Concepts and detailed description

Part 2 is the FDT interface specification. It contains abstract specifications of all objects, their behavior, their interface services and interactions. IEC 62453-2 is normative.

5.3 Parts 3xy – Communication profile integration

5.3.1 General

IEC 62453-3xy specifies the communication service for an FDT application. Implementation of these features requires suitable profiles from appropriate technologies. These communication implementation profiles are specified in the related parts of IEC/TR 62453-5xy.

The IEC 62453-3xy parts have normative content and at least one communication technology profile is required for a working implementation.

NOTE The initial specification includes several communication technology profiles and additional profiles can be added in future editions of this Part.

5.3.2 Communication profile integration – IEC 61784 CPF 1

IEC 62453-301 provides information for integrating FOUNDATION Fieldbus®[1] technology into the FDT interface specification (IEC 62453-301).

5.3.3 Communication profile integration – IEC 61784 CPF 2

IEC 62453-302 provides information for integrating CIP™[2] technology into the FDT interface specification (IEC 62453-302).

5.3.4 Communication profile integration – IEC 61784 CP 3/1 and 3/2

IEC 62453-303-1 provides information for integrating PROFIBUS[3] technology into the FDT interface specification (IEC 62453-303-1).

5.3.5 Communication profile integration – IEC 61784 CP 3/4, CP 3/5 and 3/6

IEC 62453-303-2 provides information for integrating PROFINET[4] technology into the FDT interface specification (IEC 62453-303-2).

5.3.6 Communication profile integration – IEC 61784 CPF 6

IEC 62453-306 provides information for integrating INTERBUS®[5] technology into the FDT interface specification (IEC 62453-306).

---

1 FOUNDATION® Fieldbus is the trade name of the consortium Fieldbus Foundation (non-profit organization). This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

2 CIP™ is a trade name of Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name CIP™. Use of the trade name CIP™ requires permission of Open DeviceNet Vendor Association, Inc.

3 PROFIBUS logo is the registered trade mark of PROFIBUS International (PI). PI is a non-profit trade organization to support the fieldbus PROFIBUS. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the tradename PROFIBUS. Use of the tradename PROFIBUS requires permission of the tradename holder.

4 PROFINET logo is the registered trade mark of PROFIBUS International (PI). PI is a non-profit trade organization to support the fieldbus PROFIBUS. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the tradename PROFIBUS. Use of the tradename PROFIBUS requires permission of the tradename holder.

5 "INTERBUS is the trade name of Phoenix Contact GmbH & Co. KG., control of trade name use is given to the non-profit organisation INTERBUS Club (http://www.interbusclub.com/). This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name INTERBUS. Use of the trade name INTERBUS requires permission of the INTERBUS Club.

5.3.7 Communication profile integration – IEC 61784 CPF 9

IEC 62453-309 provides information for integrating HART®[6] technology into the FDT interface specification (IEC 62453-309).

5.3.8 Communication profile integration – IEC 61784 CPF 15

IEC 62453-315 provides information for integrating Modbus®[7] technology into the FDT interface specification (IEC 62453-315).

5.4 Parts 4x – Object model integration profiles

5.4.1 General

IEC 62453-2 specifies the object behavior requirements and client/server functions for an FDT application. Implementation of these features requires suitable profiles from appropriate technologies. These object implementation profiles are specified in various parts of IEC 62453-4z.

The IEC 62453-4z parts have informative content and explain the additions to IEC 62453, Part 2 which are necessary to implement FDT based on a specific technology. Each Part 4x defines a specific technology, Part 41 specifies the implementation profile for a common object model technology.

NOTE The initial specification includes one object model profile and additional profiles will be added in future editions of IEC 62453.

5.4.2 Object model integration profile – Common object model

IEC/TR 62453-41 contains all details of the services provided by implementation technology depended interfaces, for example COM. IEC/TR 62453-41 is informative and therefore a TR.

5.5 Parts 5xy – Communication profile implementation

5.5.1 General

IEC/TR 62453-5xy specifies the implementation syntax of the communication service for an FDT application. These communication implementation profiles are based on the communication services specified in various parts of IEC 62453-3xy.

The IEC/TR 62453-5xy parts have informative content and at least one implementation of a communication technology profile is required for a working implementation.

NOTE The initial specification includes several communication technology profiles and additional profiles can be added in future editions of this Part.

5.5.2 Communication profile integration – IEC 61784 CPF 1

IEC/TR 62453-501 provides the syntax for integrating FOUNDATION Fieldbus®[8] technology into the FDT interface specification (IEC/TR 62453-501).

---

6 HART ® is the trade name of the product supplied by HART Communication Foundation. This information is given for convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

7 Modbus is the trademark of Schneider Automation Inc. It is registered in the United States of America. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trademark Modbus. Use of the trademark Modbus requires permission from Schneider Automation Inc.

8 FOUNDATION® Fieldbus is the trade name of the consortium Fieldbus Foundation (non-profit organization). This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

5.5.3 Communication profile integration – IEC 61784 CPF 2

IEC/TR 62453-502 provides the syntax for integrating CIP™[9] technology into the FDT interface specification (IEC/TR 62453-502).

5.5.4 Communication profile integration – IEC 61784 CP 3/1 and 3/2

IEC/TR 62453-503-1 provides the syntax for integrating PROFIBUS[10] technology into the FDT interface specification (IEC/TR 62453-503-1).

5.5.5 Communication profile integration – IEC 61784 CP 3/4, CP 3/5 and 3/6

IEC/TR 62453-503-2 provides the syntax for integrating PROFINET[11] technology into the FDT interface specification (IEC/TR 62453-503-2).

5.5.6 Communication profile integration – IEC 61784 CPF 6

IEC/TR 62453-506 provides the syntax for integrating INTERBUS®[12] technology into the FDT interface specification (IEC/TR 62453-506).

5.5.7 Communication profile integration – IEC 61784 CPF 9

IEC/TR 62453-509 provides the syntax for integrating HART®[13] technology into the FDT interface specification (IEC/TR 62453-509).

5.5.8 Communication profile integration – IEC 61784 CPF 15

IEC/TR 62453-515 provides the syntax for integrating Modbus®14 technology into the FDT interface specification (IEC/TR 62453-515).

---

9 CIP™ is a trade name of Open DeviceNet Vendor Association, Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name CIP™. Use of the trade name CIP™ requires permission of Open DeviceNet Vendor Association, Inc.

10 PROFIBUS logo is the registered trade mark of PROFIBUS International (PI). PI is a non-profit trade organization to support the fieldbus PROFIBUS. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the tradename PROFIBUS. Use of the tradename PROFIBUS requires permission of the tradename holder.

11 PROFINET logo is the registered trade mark of PROFIBUS International (PI). PI is a non-profit trade organization to support the fieldbus PROFIBUS. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the tradename PROFIBUS. Use of the tradename PROFIBUS requires permission of the tradename holder.

12 INTERBUS is the trade name of Phoenix Contact GmbH & Co. KG., control of trade name use is given to the non-profit organisation INTERBUS Club (http://www.interbusclub.com/). This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trade name INTERBUS. Use of the trade name INTERBUS requires permission of the INTERBUS Club.

13 HART ® is the trade name of the product supplied by HART Communication Foundation. This information is given for convenience of users of this document and does not constitute an endorsement by IEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

14 Modbus is the trademark of Schneider Automation Inc. It is registered in the United States of America. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance to this profile does not require use of the trademark Modbus. Use of the trademark Modbus requires permission from Schneider Automation Inc.

5.6 Parts 6x – DTM styleguides

5.6.1 General

Besides the technical specifications provided in IEC 62453-2 and IEC/TR 62453-41 guidelines are provided that supports a consistent look and feels of DTMs.

The IEC 62453-6z parts have informative content and explain the additions to IEC 62453 which are necessary to implement a consistent look and feel based on a specific technology. Each Part 6x defines support for a specific technology, Part 61 specifies the style guide for a common object model technology.

5.6.2 Device Type Manager (DTM) styleguide for common object model

IEC/TR 62453-61 contains all details of the GUI design bases on a specific implementation technology. IEC/TR 62453-61 is informative and a Technical Report.

6 Relation of the IEC 62453 series to other standardization activities

Field devices and modules are typically integrated as components into an industrial automation system to perform the required application. The structure of an industrial automation system may be represented as a set of multiple hierarchical levels connected by communication systems as illustrated in Figure 9.

A communication system (e.g. a fieldbus) connects the field devices to the next level controllers, which are typically programmable logic controllers (PLC) or distributed control systems (DCS) or manufacturing execution systems (MES). Since the engineering tools and the commissioning tools can access both the field devices and the controllers, these tools are also located at the controller level.

In larger automation systems the next level may be connected via a communication system such as Ethernet. This level contains complex visualization systems (HMI), central engineering tools and SCADA host systems. Multiple clusters of field devices (with or without a controller, as described above) can be connected with each other or to higher-level systems through communication networks such as fieldbus.

Additional levels may also be provided to support manufacturing execution system (MES), enterprise resource planning (ERP) and other information technology (IT) based systems with access to field devices indirectly via LAN's and the controllers or directly via routers.

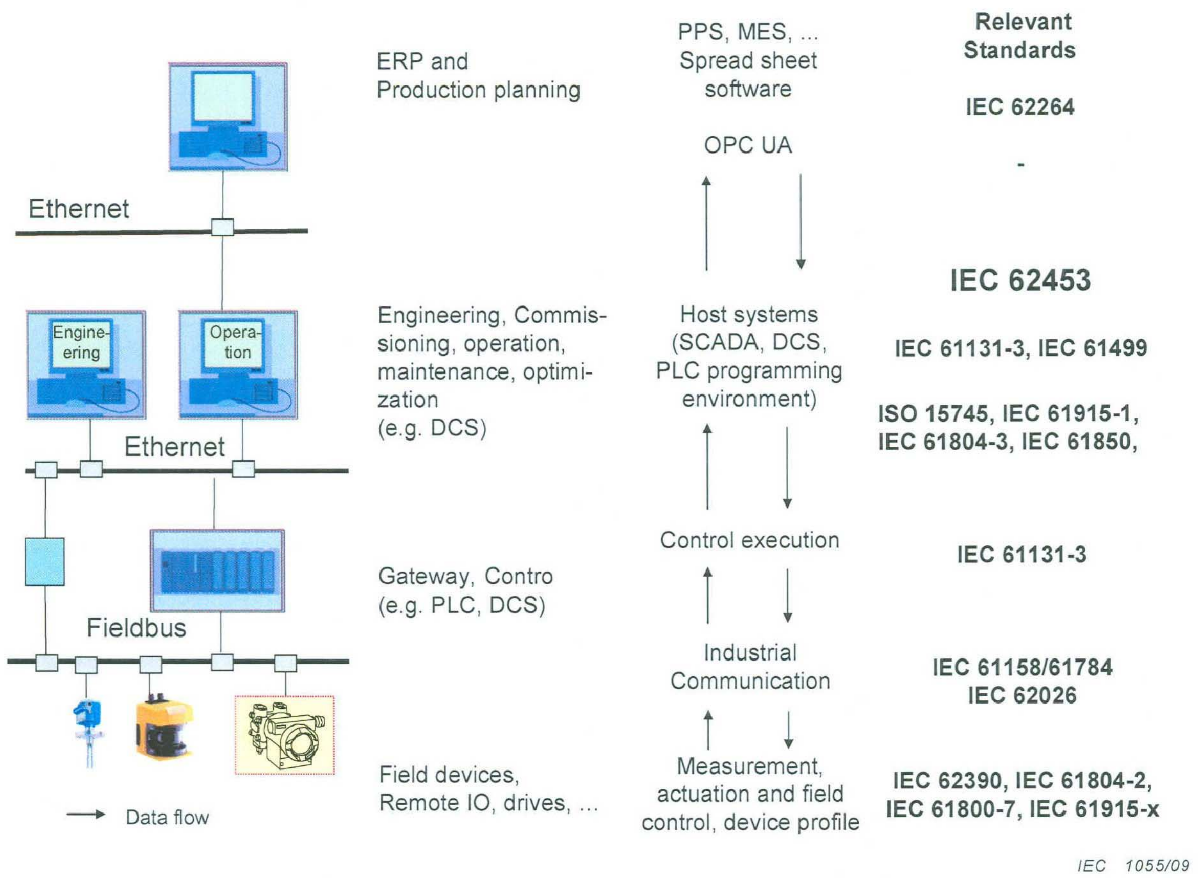


Figure 9 – Standards related to IEC 62453 – in an automation hierarchy

There are a number of standards related to communication and integration of field devices into automation systems. These standards are shown in Figure 9 at the levels which normally implement the functions or elements specified in the standard.

Device profiles in terms of parameter lists, function blocks or objects are usually intended to be implemented in field devices (IEC 61915-1, IEC 61800-7 series, IEC 61804-2 and IEC/TR 62390). Industrial communication standards are covered by IEC 61158 and IEC 61784-1/IEC 61784-2 and in IEC 62026 series. IEC 61131 series covers languages for programmable controllers. The standards ISO 15745, IEC 61804-3, IEC 61519-1 and IEC 61850 series cover description languages to represent field devices. Elements of both IEC 61131 series and IEC 61499 series can be used for design and programming of higher functions in operation, commissioning and maintenance stations.

This standard, the IEC 62453 series provides generic interfaces to various host applications needing to interact with field devices and their related control functions.

Higher level applications such as MES and ERP are reflected in IEC 62264 series.

Table 1 gives a short summary of the above-mentioned standards:

Table 1 – Overview of related standards

IEC 61131-3	This standard describes the textual programming languages Structured Text and Instruction List as well as the graphical languages Ladder Diagram, Function Block Diagram. All languages work together with a common software model of the PLC and common elements such as variables and tasks.
IEC 61158	IEC 61158 is a collection of multiple fieldbus and Ethernet based industrial communication specifications. These communication systems represent the actual state of the art of industrial communication.
IEC 61499 series	Function blocks for industrial measurement and control systems. General function block model using an event driven architecture.
IEC 61784-1 / IEC 61784-2	IEC 61784-1/IEC 61784-2 describe communication profiles of certain fieldbusses based on the specifications of IEC 61158.
IEC 61800-7 series	IEC 61800-7-n is a generic interface and profiles for power drive systems. Part 7-1 is a generic interface definition for power drives, Part 7-2 specifies a number of power drive profiles, Part 7-3 specifies the mapping of the profiles and interfaces in Part 7-1 and Part 7-2 to work with network communication technologies such as IEC 61158.
IEC 61804-2	This standard specifies FB by using the device model which defines the components of an IEC 61804-2 conformant device and conceptual specifications of FBs for measurement, actuation and processing. This includes general rules for the essential features to support control, whilst avoiding details which stop innovation as well as specialization for different industrial sectors. This Part of IEC 61804 provides conceptual Function Block specifications, which can be mapped to specific communication systems and their accompanying definitions by industrial groups.
IEC 61804-3	This Part specifies the Electronic Device Description Language (EDDL) technology, which enables the integration of real product details using the tools of the engineering life cycle. The EDDL fills the gap between the conceptual FB specification of IEC 61804-2 and a product implementation. It is a generic language for describing the properties of automation system components. EDDL is capable of describing: • device parameters and their dependencies; • device functions, for example, simulation mode, calibration; • graphical representations, for example, menus; • interactions with control devices; • enhanced user interface; • graphing system; • persistent data store.
IEC 61850 series	IEC 61850 series " Communication and systems in substation " is the global standard for information models and information exchange for substation automation. IEC 61850 series has become the base standard for modeling of power systems information, for exchanging information between devices, and for the configuration of systems and devices applied in the whole electrical energy supply chain. Information produced and consumed by substations are modeled as logical nodes, data objects and data attributes.
IEC 61915-1	“Low-voltage switchgear and control gear device profiles for networked industrial devices – Part 1 General rules for the development of device profiles“ defines a frame work for common representation of networked industrial devices including a profile template for documentation. Main focus is drawn to the above-mentioned device classes. This representation follows the principles given in the IEC/TR 62390 “Common automation device – Profile guideline“, also refer to ISO 15745.
IEC 62026 series	“Low-voltage switchgear and control gear – Controller-device interfaces (CDIs)" specifies communication interfaces between low-voltage switchgear, control gear, and controllers (e.g. programmable controllers, personal computers), for use in industrial automation applications. Part 1 specifies common requirements for CDIs, while subsequent parts specify several CDIs technologies, using a common document structure as defined in Part 1.
IEC/TR 62390	This technical report provides guidance for the development of device profiles for industrial field devices and control devices, independent of their complexity. This guideline focuses on the functional aspects of the device. It is a recommended outline for use by standardization product committees, fieldbus consortia and product manufacturers to develop and provide profiles for networked devices. Some aspects of this guideline may also be applicable to stand-alone devices. It is the intention of this guideline to provide a common and more generic way to publish device information and behavior.
ISO 9506-1	Manufacturing Message Specification (MMS). MMS specifies structure for messages required to control and monitor intelligent electronic devices. It is an OSI Reference model layer 7 specification.

ISO 15745	”Open system application integration framework”, the first Part defines the generic elements and rules for describing integration models and application interoperability profiles, together with their component profiles - process profiles, information exchange profiles, and resource profiles. The others parts define the technology specific elements and rules for describing both communication network profiles and the communication related aspects of device profiles based upon particular fieldbus technologies. This standard specifies a device model, which is supported by an XML schema allowing the production of an XML device description file.
ISO/IEC 19501	“Information technology – Open Distributed Processing – Unified Modeling Language (UML)”. UML provides means for the specification of objects, their relations and behavior. UML is frequently used in international standards.

The standards have various relationships to each other as shown in Figure 10 and Table 1. In Figure 10 solid lines are used to represent normative references, and dashed lines represent informative references (contained in the Bibliography of the standards). The standards are grouped under three headings:

- industrial communication;
- measurement, actuation, field control and device profile;
- device description.

The connecting lines show which industrial communication standard supports which field device standard. IEC/TR 62390 is shown as a bridging function between field device related standards and various devices description specifications.

As shown, this IEC 62453 series specification can use other standards for field device functions, devices description and controller related standards in DTM. Similarly it can use all communication profile families of IEC 61784-1/IEC 61784-2.

ISO/IEC 19501:2005, i.e. UML provides abstract description models and languages used for the formalized descriptions of the specification content of the standards as described in Table 1.

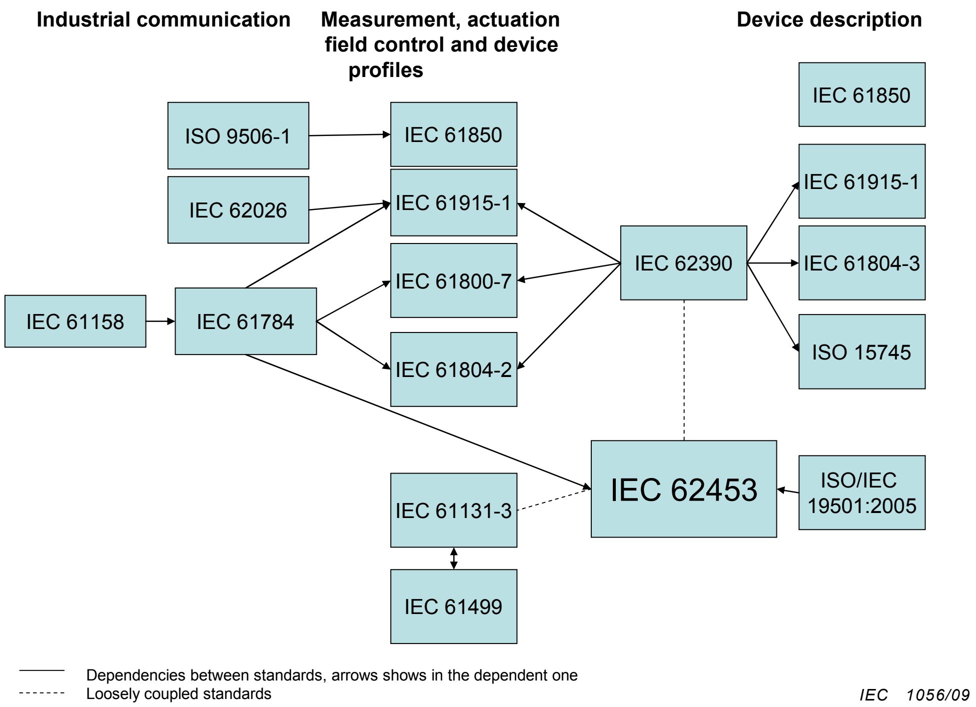


Figure 10 – Standards related to IEC 62453 – grouped by purpose

7 Migration to DTM

Industrial systems use many different field devices and modules ranging from simple I/O sensors to complex, modular remote-I/Os and drives. These devices and modules may be grouped into four categories of complexity:

a) simple devices that communicate only cyclically, for example a light barrier;

b) adjustable devices with fixed hardware and software, for example a pressure transducer;

c) adjustable devices with modular hardware but fixed software blocks, for example remote I/O;

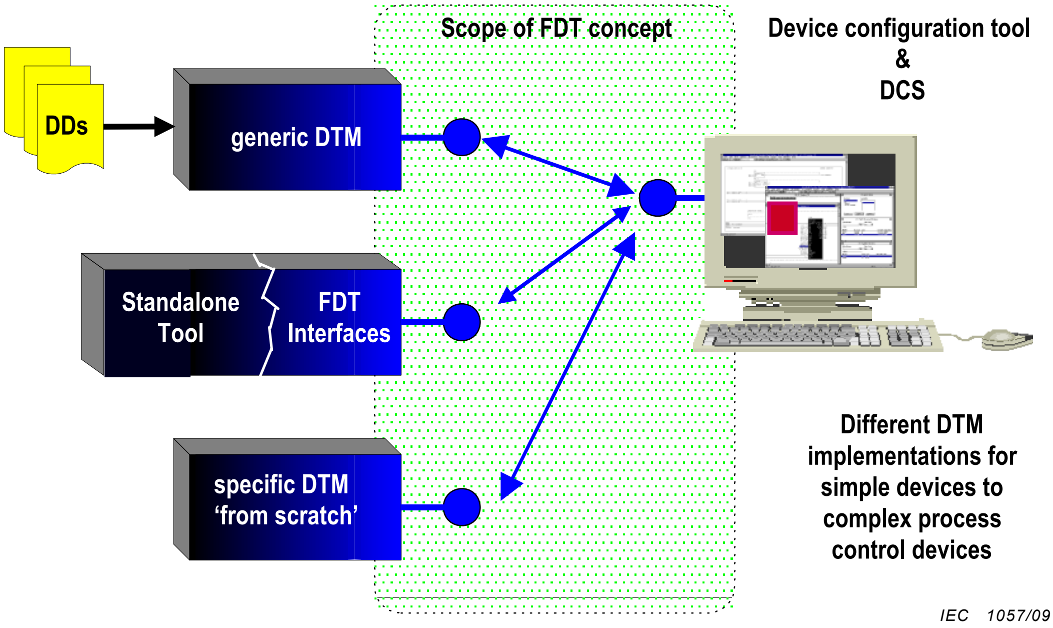
d) adjustable devices with modular hardware and programmable software blocks, for example a complex servo-drive.

These different devices come with different kinds of descriptions of their capabilities (device description) or even with their own configuration tools depending on the functionality the devices provide. FDT enables all these devices to be integrated into Frame Applications via DTMs in a unified way. For instance, simple devices of categories a) and b) may be sufficiently described by existing device descriptions (files containing information about the device capabilities). It is possible to develop ‘generic DTMs’ that can interpret these device descriptions and make the contained information and functions available for the system and its user. Once a generic DTM for a specific device or module description is developed, all modules supporting this description can be integrated using the same DTM.

Devices of categories c) and d) may also have existing standalone tools. FDT provides the interfaces and features to equip these tools with the FDT-interfaces and to build DTMs using existing standalone tools. DTMs that are equipped with external tools can use this approach to migrate towards FDT and protect existing investments. The final goal of such a migration should be a DTM, which provides as well an integrated GUI.

Each device and module manufacturer has the freedom to choose from the following options for existing or new items (see Figure 11):

- retain generic solutions based on existing descriptions;
- offer a DTM that is maintained as a standalone tool;
- build a specific DTM to fully describe all available features and methods for handling the device or module.



Key

DD – Device description

Figure 11 – DTM – implementations

8 How to read IEC 62453

8.1 Architecture

The FDT functionality and the chosen architecture are described in an abstract way in Part 2 of this standard. The description includes:

- main conceptual elements;
- their relation to each other (e.g. aggregation or specialization);
- their interactions;
- description of services and service primitives of all FDT elements together with their mandatory and optional arguments. The arguments are structured data types.

Each implementation technology dependent detail of the services is described in Part 41. This means that the syntax of the service primitives in detail as well as the interface definition for FDT is specified.

8.2 Dynamic behavior

Dynamic behavior of an FDT system is described in Part 2 using sequence diagrams to show the interactions using abstract message names.

The details of these interactions are described in Part 41 using the syntax defined in the FDT IDL.

Dynamic behavior of a DTM as well as the management of data is described in terms of state machines in Part 2. While Part 2 describes the abstract aspects of the DTM behavior as state machine, Part 41 describes the details of the states and transitions using the syntax of the service mapping to the interface methods.

Part 41 describes additional implementation specific dynamic behavior.

8.3 Structured data types

The service primitive arguments are structured data types. They are described in Part 2 as abstract data types. Part 41 describes them in terms of XML schemas.

8.4 Fieldbus communication

The Communication Profile Family (CPF) specific service primitives are transmitted using Communication Channel mechanism which is described in principle and with the abstract arguments in Part 2. Part 41 contain the implementation dependent specification of the syntax of the Communication Channel service primitives.

All CPF specific services and service primitives are described in parts 3xy together with their mandatory and optional arguments. Parts 5xy contain the implementation dependent schema necessary to transmit the communication service primitives described in the related Part 3xy.

Annex A
(informative)

UML notation

A.1 General

Note
contains information to people reading the UML diagram (or model). Notes provide additional context to help explain details that are not apparent in the diagram (see Figure A.1).

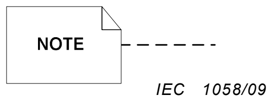


Figure A.1 – Note

A.2 Class diagram

The class diagram is one of the UML specification methods. The UML elements, which are used in the class diagrams of IEC62453 are explained in the following.

Class
is a description of a set of objects that share the same attributes, operations, methods, relationships, and semantics (see Figure A.2).

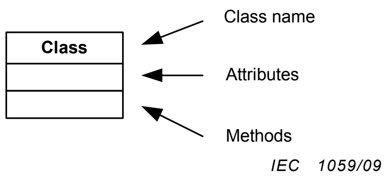


Figure A.2 – Class

Abstract class
is a class that cannot be directly instantiated and is used only for specification purposes. A class is abstract if it has no instances. An abstract class is only used to inherit from. Abstract classes are represented by an italicized class name.

Association
is the semantic relationship (between two or more classifiers) that specifies connections among their instances (see Figure A.3).

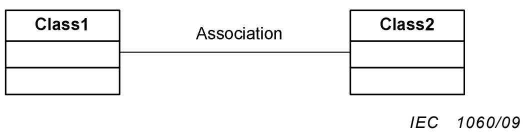


Figure A.3 – Association

Composition
is a form of symmetric association that specifies a whole-part relationship between the composition (whole) class and a subordinate (part) class in which removing the whole also removes the parts (see Figure A.4).

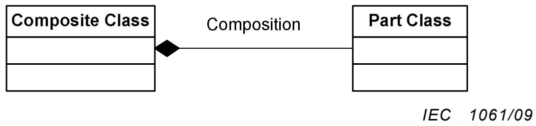


Figure A.4 – Composition

Aggregation
is a form of asymmetric association that specifies a whole-part relationship between the aggregate (whole) class and a subordinate (part) class (see Figure A.5).

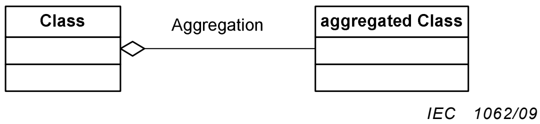


Figure A.5 – Aggregation

Dependency
is a form of association that specifies a dependency relationship between two classes. An arrowhead can be used to indicate an asymmetric dependency (see Figure A.6).

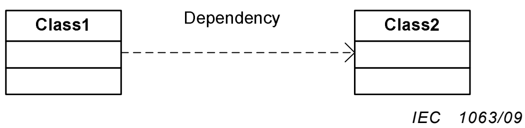


Figure A.6 – Dependency

Generalization
is the taxonomic relationship between a more general element and a more specific element that is fully consistent with the first element and that adds additional information. It is used for classes, packages, use cases, and other elements. The construct is also use to describe Inheritance (see Figure A.7).

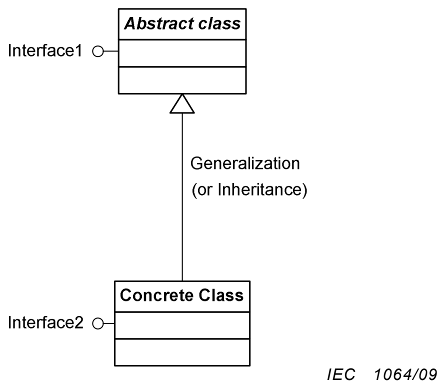


Figure A.7 – Abstract class, generalization and interface

Interface
is a mechanism used to conveniently package and reuse a collection of methods (method signatures) and constants. An interface is an abstract class that only contains method signatures and can also contain constants. There is no underlying implementation for the methods. Essentially, an interface is a promise to implement a standard package of methods and constants. An interface may be inherited by an abstract class as well as by a concrete class may also implement an interface. In Figure A.7, the concrete class implements Interface1 (inherited with abstract class) as well as Interface2.

Multiplicity
multiplicity specification is shown as a text string comprising a comma-separated sequence of integer intervals (see Figure A.8), where an interval represents a (possibly infinite) range of integers, in the format: lower-bound .. upper-bound where lower-bound and upper-bound are literal integer values, specifying the closed (inclusive) range of integers from the lower bound to the upper bound. In addition, the star character (*) may be used for the upper bound, denoting an unlimited upper bound.

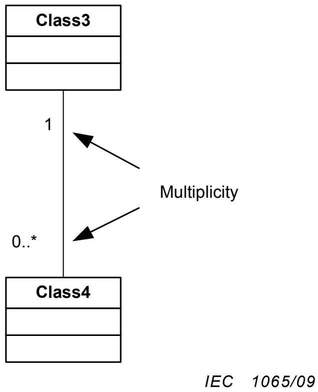


Figure A.8 – Multiplicity

A.3 Statechart diagram

The statechart diagram is a graph that represents a state machine. States and various other types of vertices (pseudostates) in the state machine graph are rendered by appropriate state and pseudostate symbols, while transitions are generally rendered by directed arcs that interconnect them (see Figure A.9).

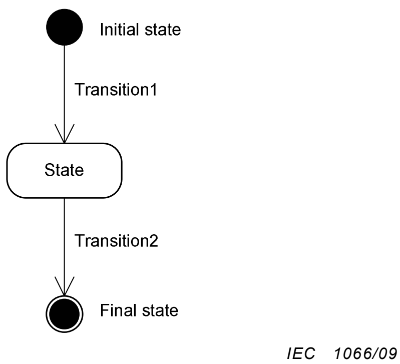


Figure A.9 – Elements of UML statechart diagrams

A **state** is shown as a rectangle with rounded corners. Optionally, it may have an attached name tab.

An **initial state** (pseudostate) is shown as a small solid filled circle.

A **final state** is shown as a circle surrounding a small solid filled circle (a bull’s eye).

A **composite** (**super) state** is shown as a rectangle with rounded corners, containing two small ellipses. Such a state is composed from a set of sub-states, which in turn are connected by transitions.

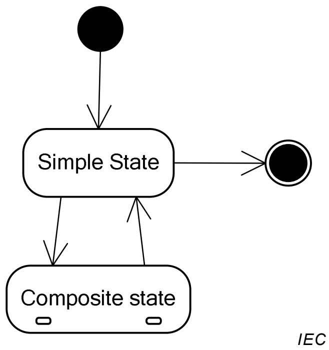


Figure A.10 – Example of UML state chart diagram

A.4 Use case diagram

A **use case diagram** is a class diagram for specifying required usages of a system. It contains actors, use cases and their relations. The following figure shows the UML syntax that is used throughout this specification.

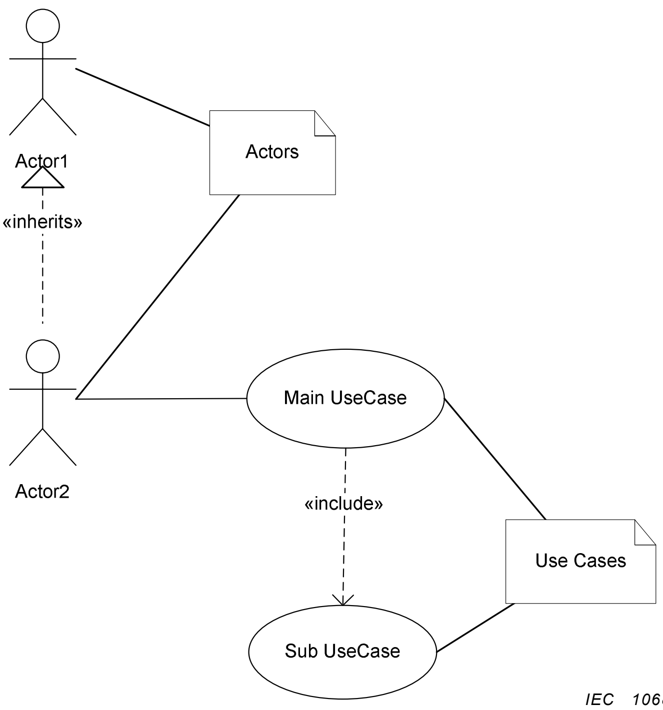


Figure A.11 – UML use case syntax

An **actor** is shown as a person. It can represent a human being or an external other system interaction with the system which is specified.

A **use case** captures a functional requirement by way of describing the interaction between actor and the system. If complex interactions are composed from more simple interactions, this is shown by 'include' relation.

An **inheritance** relation is shown as a dashed line with a triangle at the parent, i.e. more general element. Inheritance between actors is used in this standard to describe that one actor 'inherits' the permissions to execute certain use cases of another actor.

A.5 Sequence diagram

The sequence diagram is a diagram that depicts an interaction by focusing on the sequence of messages between objects on the lifelines.

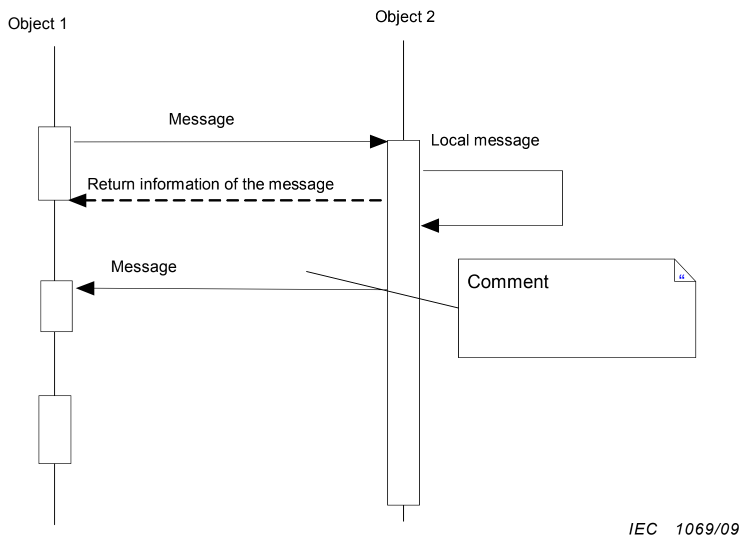


Figure A.12 – UML sequence diagram

Object instances are represented by a vertical line.

A message is represented by an arrow. In this specification, a full arrow is used to depict the general occurrence of a message independently of synchronous or asynchronous handling of the message.

The sequence of messages is defined by the order of messages starting from the top of the diagram.

Annex B
(informative)

Implementation policy

B.1 General

FDT technology is based on component technology with components (software modules) being provided by different manufacturers for integration into a complete system. The suppliers of FDT based software are responsible for the conformance of the software modules. System integrators and/or technology organizations are responsible to provide a sufficient integration framework and test programs or certification process that enables the separate modules to be integrated efficiently and securely.

Bibliography

[1] IEC 60050 (all parts), *International Electrotechnical Vocabulary*

NOTE See also the IEC Multilingual Dictionary – Electricity, Electronics and Telecommunications (available on CD-ROM and at <http://www.electropedia.org/ >).

[2] ISO/IEC 7498 (all parts), *Information processing systems – Open Systems Interconnection – Basic Reference Model*

[3] ISO 2382 (all parts), *Information technology – Vocabulary*

[4] ISO/AFNOR *Dictionary of computer science*

[5] IEC 61131 (all parts)*, Programmable controllers*

[6] IEC 61499 (all parts), *Function blocks*

[7] IEC 61800-7 (all parts), *Adjustable speed electrical power drive systems – Part 7-X: Generic interface and use of profiles for power drive systems*

[8] IEC 61850 (all parts), *Communication networks and systems in substations*

[9] IEC 61915-1*, Low-voltage switchgear and controlgear – Device profiles for networked industrial devices – Part 1: General rules for the development of device profiles*

[10] IEC 62026 (all parts), *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs)*

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

[12] ISO 9506-1*, Industrial automation systems – Manufacturing Message Specification – Part 1: Service definition*

[13] IEC 61804-2, *Function blocks (FB) for process control – Part 2: Specification of FB concept*

[14] IEC 61804-3, *Function blocks (FB) for process control – Part 3: Electronic Device Description Language (EDDL)*

[15] ISO 15745 (all parts), *Industrial automation systems and integration – Open systems application integration framework*

[16] IEC 62453 (all parts), *Field Device Tool (FDT) interface specification*

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