TECHNICAL SPECIFICATION

IEC TS 61804-1

First edition
2003-10

Function blocks (FB) for process control –

Part 1: Overview of system aspects

Reference number
IEC/TS 61804-1:2003(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

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

TECHNICAL SPECIFICATION

IEC TS 61804-1

First edition
2003-10

Function blocks (FB) for process control –

Part 1: Overview of system aspects

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XF

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FUNCTION BLOCKS (FB) FOR PROCESS CONTROL –

Part 1: Overview of system aspects

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.

The main task of IEC technical committees is to prepare International Standards. In exceptional circumstances, a technical committee may propose the publication of a technical specification when

- the required support cannot be obtained for the publication of an International Standard, despite repeated efforts, or
- the subject is still under technical development or where, for any other reason, there is the future but no immediate possibility of an agreement on an International Standard.

Technical specifications are subject to review within three years of publication to decide whether they can be transformed into International Standards. IEC 61804-1, which is a Technical Specification, has been prepared by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This standard cancels and replaces IEC/PAS 61804-1 published in 2002.

The text of this technical specification is based on the following documents:

Enquiry draft	Report on voting
65C/296/DTS	65C/310A/RVC

Full information on the voting for the approval of this technical specification 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.

IEC 61804 consists of the following parts under the general title *Function blocks (FB) for process control*

Part 1: *Overview of system aspects*

Part 2: *Specification of FB concept and electronic device description language (EDDL)*

The committee has decided that the contents of this publication will remain unchanged until 2007. At this date, the publication will be

- transformed into an International standard;
- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

INTRODUCTION

This Technical Specification is an end-user driven specification of the requirements of distributed process control systems based on Function Blocks (FB). This Technical Specification and its associated FB standard (IEC 61804-2) originate from the power-plant industrial sector. It is validated by applications in oil and gas, petrochemicals, pharmaceuticals and fine chemicals, pulp and paper, food and beverage, waste water treatment plants, steel milling and others. There will be other general requirement standards and associated specifications for other industrial sectors.

Present and future digital process control systems need to fulfil the following requirements:

- increase security and safety;
- reduce time to market;
- be supportable with available tools;
- reduce costs of development and support;
- minimize training costs;
- support integration of distributed control applications
- support integrated methodology for implementation;
- have increased maintainability, modifiability, agility, upgradeability, flexibility, ability to validate, accessibility, availability, compatibility of support tools, multi-vendor device/ application compatibility, re-usability of knowledge and designs, re-usability of software components;
- be made up of digital devices that are compatible, interworkable, interconnectable interoperable and interchangeable with each other.

Process control systems are required to fulfil these requirements in terms of their architecture and their operation during all the phases of the life cycle. The accepted basic concept for the design process control system is to describe all necessary implementation-specific functions with FB. A FB is an encapsulation of data and algorithms to provide a specific function, which can be self-standing. Process control systems can involve many instances of many different FBs operating in an environment providing common services (for example, communications) and interfaces to other applications. See Figure 1.

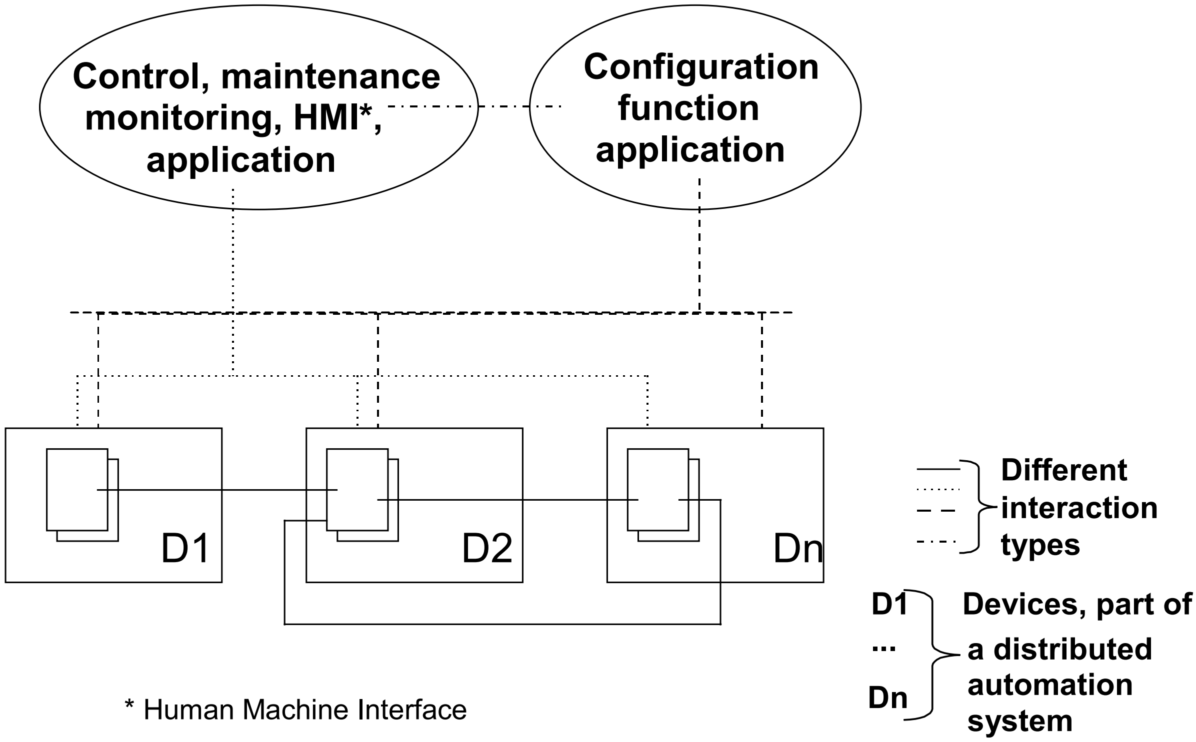


Figure 1 – Interactions of applications

FUNCTION BLOCKS (FB) FOR PROCESS CONTROL –

Part 1: Overview of system aspects

1 Scope

This part of IEC 61804 is a Technical Specification which provides *guidelines* for suppliers to meet evolving requirements for digital process control systems through which users can be assured of the compatibility, the interworkability, the interconnectability, the interoperability and the interchangeability of the devices they choose. This part gives the overall requirements. For better understanding, this part gives background information and examples in annexes.

This Technical Specification defines the requirements for FBs to provide control and to facilitate maintenance and technical management as applications which interact with actuators and measurement devices:

- control covers functions necessary to bring and hold the process at the desired behaviour;
- maintenance covers functions to acquire information about the state of the process equipment and the state of automation devices including their adjustments, for example, calibrate a sensor that has drifted;
- technical management deals with information for the optimization of the process.

  It is concerned primarily with the economics of the process and plant equipment. This relates in particular to the evaluation of performance and reliability of specific items of plant or equipment for all the installed period and comparison of performance and reliability of items from different suppliers performing an identical function in the same operating environment. An example of performance is the number of cycles achieved before failure of two valves from different suppliers. This allows for the production of detailed and valid statistical analysis to support management decisions and plant equipment modifications.

A prerequisite for designing, implementing and operating a FB-based process control system is that the tools, the devices and other components follow the same architecture based on a common specification. The architecture is required to define the components of the systems, for example FB, device, data, data connections and more as well as relations between these components. The IEC 61499 series generic FB model on which this Technical Specification is related is able to provide these basic components for FBs for process control. One add-on to the IEC 61499 series is the specification of parameters and functions of FBs that are implementable in devices.

The architecture and the range of FBs that have to be specified are described in 7.4 contains a minimum set of FBs that will be required for the process industries. These are presented in TWO different clauses. One deals with “rich” FBs covering complex but common functions such as control loop (for example proportional, integral, differential – PID) required by the majority of the process industries. Another covers a set of elementary FBs (EFB) such as Boolean functions required to compose very specific and unique functionality.

FBs are used during the complete life cycle of process control systems but viewed from different aspects. This is covered in detail in Annex A. The process design starts with the Piping and Instrumentation Diagram (P&ID) which gives the requirements of the process and instrumentation from a purely functional point of view. From the P&ID, the desired behaviour of the process control system is extracted into a functional requirements diagram (FRD) without considering the detailed behaviour of the underlying devices. The bricks making up the FRD are application blocks (AB), the representation of the data and algorithms in the design phase. After discussion between the process and automation engineers (end-user and system integrator), the FRDs are turned into detailed designs for the application via several design using devices available on the market together with interconnections and configurations of these devices. In this way a PID loop shown in via bubbles on a P&ID will be transformed into implementable FBs in specific field and/or control-room devices. It should be noted that many parts of the process industries, in particular those with many similar and relatively simple processes (for example, the water industry), do not use the concept or term FRD. They go directly from P&IDs to the implementable FBs and will use a variety of names to describe the process and the resulting design documents. The FRD approach is used here since it represents the most formal view of the design cycle and illustrates the use of FBs at the earliest of phases in the life cycle. Clause 4 summarizes the requirements from this life-cycle point of view.

This Technical Specification specifies a system (an industrial-process measurement and control system based on distributed FB application). A system is described stepwise in terms of architecture, models and the life cycle. The architecture is the "road map" which names the components and presents the structure of the system. The models describe the details of the components, i.e. their functions in the system. The life cycle makes visible how the components work together during their use in different phases of the lifetime, i.e. it makes the operation visible.

Figure 2 shows the different influences, basic specifications and technology support on IEC 61804 from the top-down and bottom-up point of view.

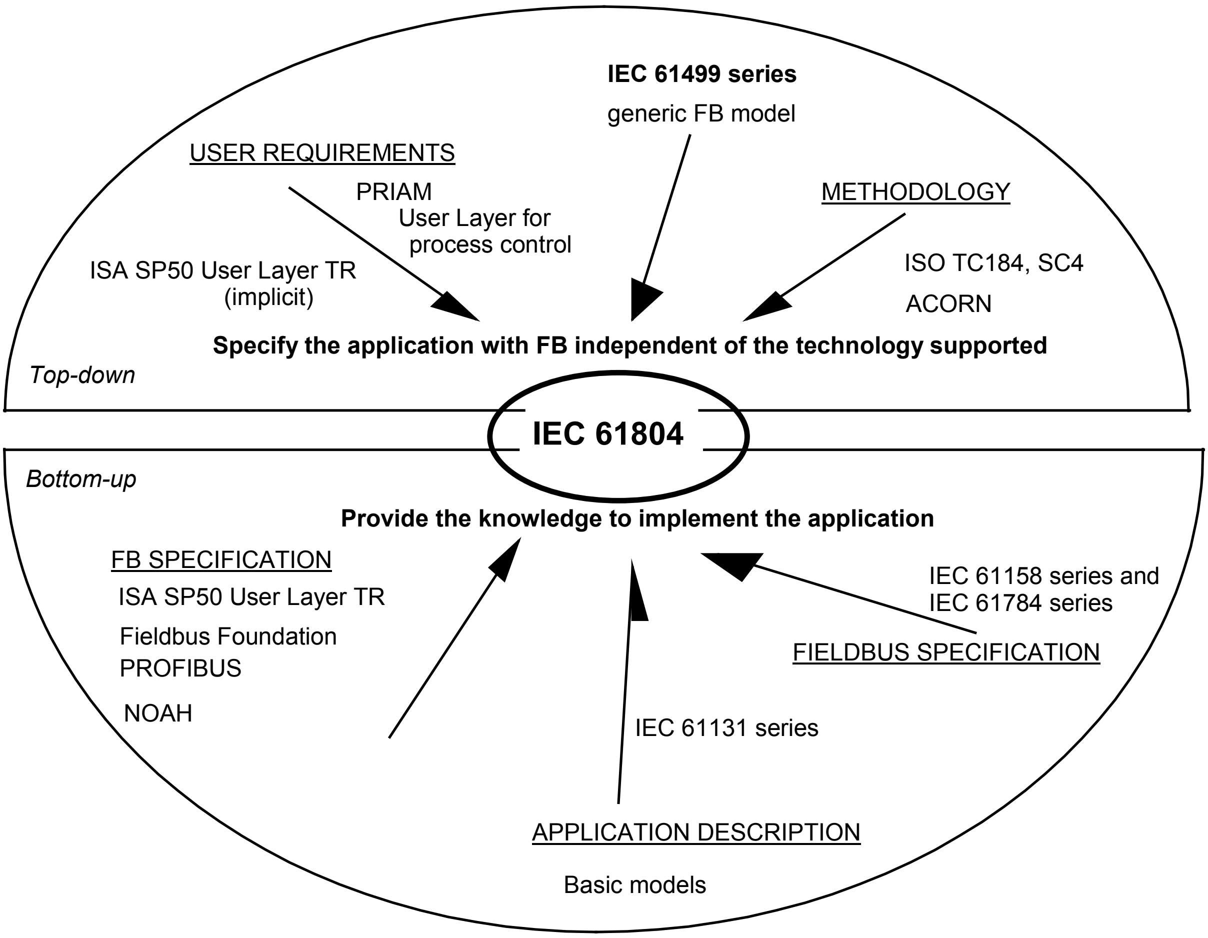


Figure 2 – Influences on IEC 61804

The influences are international standards and projects (PROFIBUS and Fieldbus Foundation[1]), which relate to the same area as IEC 61804. These standards are either technology-independent ones supporting the top-down approach or dedicated to a certain technology, for example, programmable controller or fieldbus. Both together will build the basis of the standard specified by IEC 61804.

The main purpose of this part is the harmonization of different views, models and starting points of end-users, system providers and device manufacturers. It will be the reference document leading the discussions during the specification and the guideline for the readers of IEC 61804-2.

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 61131-3:1992, *Programmable controllers – Part 3: Programming languages*

IEC 61499-1:2000, *Function blocks for industrial-process measurement and control systems – Part 1: Architecture*

IEC 61512-1:1997, *Batch control – Part 1:Models and terminology*

IEC 61784:2003, *Digital data communications for measurement and control – Part 1: Profile sets for continuous and discrete manufacturing – Fieldbus relative to use in industrial control systems*

IEC 61158 (all parts), *Digital data communication for measurement and control – Fieldbus for use in industrial control systems*

IEC 61804-2, *Function blocks for process control – Part 2: Specification of FB concept and Electronic Device Description Language (EDDL)* [2]

EN 50170:1996, *General purpose field communication system*

3 Terms, definitions and abbreviated terms

3.1 General definitions

For the purposes of this document, the following definitions apply.

3.1.1
interface
shared boundary between two *functional units,* defined by functional characteristics, signal characteristics, or other characteristics as appropriate

[IEC 60050-351:1998, 11-19]

[1] PROFIBUS 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 Technical Specification and does not constitute an endorsement by CENELEC of the trademark holder or of any of its products. Compliance to this profile does not require use of the trade name PROFIBUS. Use of the trade name PROFIBUS requires permission of the trade-name holder.

Foundation Fieldbus is the trade name of the consortium Fieldbus Foundation. This information is given for the convenience of users of this Technical Specification and does not constitute an endorsement by CENELEC of the product named. Equivalent products may be used if they can be shown to lead to the same results.

[2] To be published.

3.1.2
system
set of interrelated elements considered in a defined context as a whole and separated from its environment

[IEC 60050-351:1998, 11-01]

NOTE 1 Such element may be material objects and concepts as well as the results thereof (for example 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 system considered.

3.1.3
data type
set of values together with a set of permitted *operations*

[ISO 2382 series]

3.1.4
data connection
association between two *function blocks* for the conveyance of *data*

[IEC/PAS 61499-1:2000, 1.3.2.22]

3.1.5
data
representation of facts, concepts or instructions in a formalized manner suitable for communication, interpretation or processing by human beings or by automatic means

[ISO modified[3]]

3.1.6
functional unit
entity of *hardware* or *software,* or both, capable of accomplishing a specified purpose

[ISO 2382 series]

3.1.7
hardware
physical equipment, as opposed to programs, procedures, rules and associated documentation

[ISO modified[3]]

3.1.8
mapping
set of values having defined correspondence with the quantities or values of another set

[ISO 2382 series]

3.1.9
parameter
variable that is given a constant value for a specified *application* and that may denote the application

[ISO 2382 series]

---

3 The notation [ISO modified] following a definition indicates that the definition is taken from the "ISO/AFNOR Dictionary of Computer Science" and has been modified.

3.1.10
algorithm
finite set of well-defined rules for the solution of a problem in a finite number of *operations*

[IEC/PAS 61499-1:2000, 1.3.2.5]

3.1.11
application
software functional unit that is specific to the solution of a problem in industrial-process measurement and control

[IEC/PAS 61499-1:2000, 1.3.2.6]

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

3.1.12
application block
design pattern which is used in an FRD to represent one or multiple (rich) functions blocks

3.1.13
attribute
property or characteristic of an *entity*, for instance, the version identifier of a FB type specification

[IEC/PAS 61499-1:2000, 1.3.2.7]

NOTE The formal description of attributes is to specify interoperability. IEC 61499-1 does not specify certain attributes like FB type information. IEC 61499-1 gives the general rules to define the attributes and IEC 61804-2 specifies the attributes for process control as other groups may specify their own. Rules should be able to prevent non-unique attribute names.

3.1.14
configuration (of a *system* or *device*)
step in system design: selecting *functional units*, assigning their locations and defining their interconnections

[IEC/PAS 61499-1:2000, 1.3.2.17]

3.1.15
device
independent physical *entity* capable of performing one or more specified *functions* in a particular context and delimited by its *interfaces*

[IEC/PAS 61499-1:2000, 1.3.2.26]

3.1.16
device management application
application whose primary function is the management of a multiple *resources* within a *device*

[IEC/PAS 61499-1:2000, 1.3.2.27]

3.1.17
elementary FB
FB or *function* which provide logico-mathematical functionality with exception handling

NOTE There can be a difference between an EFB used in the FRD and the FB in the operating application

3.1.18
entity
particular thing, such as a person, place, process, object, concept, association, or *event*

[IEC/PAS 61499-1:2000, 1.3.2.28]

3.1.19
event
instantaneous occurrence that is significant to scheduling the *execution* of an *algorithm*

[IEC/PAS 61499-1:2000, 1.3.2.29]

NOTE The execution of an algorithm may make use of variables associated with an event.

3.1.20
exception
event that causes suspension of normal *execution*

[IEC/PAS 61499-1:2000, 1.3.2.35]

3.1.21
execution
process of carrying out a sequence of *operations* specified by an *algorithm*

NOTE The sequence of operations to be executed may vary from one *invocation* of a *function block* instance to another, depending on the rules specified by the FB *algorithm* and the current values of *variables* in the FB data structure.

[IEC/PAS 61499-1:2000, 1.3.2.36]

3.1.22
function
specific purpose of an *entity* or its characteristic action

[IEC/PAS 61499-1:2000, 1.3.2.42]

3.1.23
FB (FB instance)
software functional unit comprising an individual, named copy of a data structure and associated *operations* specified by a corresponding *FB type*

NOTE Typical operations of a FB include modification of the values of the data in its associated data structure.

[IEC/PAS 61499-1:2000, 1.3.2.43]

3.1.24
implementation
development phase in which the *hardware* and *software* of a *system* become operational

[ISO modified[3]]

3.1.25
input variable
variable whose value is supplied by a *data input*, and which may be used in one or more *operations* of a *FB*

NOTE An input parameter of a FB, as defined in IEC 61131-3, is an input variable.

[IEC/PAS 61499-1:2000, 1.3.2.48]

3.1.26
instance
functional unit comprising an individual, named *entity* with the *attributes* of a defined *type*

[IEC/PAS 61499-1:2000, 1.3.2.49]

3.1.27
instance name
identifier associated with and designating an *instance*

[IEC/PAS 61499-1:2000, 1.3.2.50]

3.1.28
instantiation
creation of an *instance* of a specified *type*

[IEC/PAS 61499-1:2000, 1.3.2.51]

3.1.29
internal operations (of a *FB)*
operations associated with an *algorithm* of a *FB*, with its *execution* control, or with the functional capabilities of the associated *resource*

[IEC/PAS 61499-1:2000, 1.3.2.52]

3.1.30
internal variable
variable whose value is used or modified by one or more *operations* of a *FB* but is not supplied by a *data input* or to a *data output*.

[IEC/PAS 61499-1:2000, 1.3.2.53]

3.1.31
invocation
process of initiating the *execution* of the sequence of *operations* specified in an *algorithm*

[IEC 61131-3, modified]

3.1.32
management FB
FB whose primary *function* is the management of *applications* within a *resource*

[IEC/PAS 61499-1, 1.3.2.56]

3.1.33
management resource
resource whose primary *function* is the management of other *resources*

[IEC/PAS 61499-1:2000, 1.3.2.57]

3.1.34
model
representation of a real world process, *device*, or concept

[IEC/PAS 61499-1:2000, 1.3.2.58]

3.1.35
operation
well-defined action that, when applied to any permissible combination of known *entities*, produces a new *entity*

3.1.36
output variable
variable whose value is established by one or more *operations* of a *FB* and is supplied to a *data output*

NOTE An output parameter of a FB, as defined in IEC 61131-3, is an output variable.

[IEC/PAS 61499-1:2000, 1.3.2.60]

3.1.37
resource
functional unit which has independent control of its operation and which provides various *services* to *applications,* including the scheduling and *execution* of *algorithms*

NOTE 1 The resource defined in IEC 61131-3 is a programming language element corresponding to the resource defined above.

NOTE 2 A device contains one or more resources.

[IEC/PAS 61499-1:2000, modified]

3.1.38
resource management application
application whose primary function is the management of a single *resource*

[IEC/PAS 61499-1:2000, 1.3.2.66]

3.1.39
scheduling function
function which selects *algorithms* or *operations* for *execution,* and initiates and terminates such execution

[IEC/PAS 61499-1:2000, 1.3.2.70]

3.1.40
service
functional capability of a resource which can be modeled by a sequence of service primitives

[ISO/IEC 7498-1, modified]

3.1.41
software
intellectual creation comprising the programs, procedures, rules and any associated documentation pertaining to the operation of a *system*

[ISO modified[3]]

3.1.42
transaction
unit of *service* in which a request and possibly *data* are conveyed from a requester to a responder, and in which a response and possibly data may also be conveyed from the responder back to the requester

[IEC/PAS 61499-1:2000, 1.3.2.79]

3.1.43
type
software element which specifies the common *attributes* shared by all *instances* of the type

[IEC/PAS 61499-1:2000, 1.3.2.80]

3.1.44
type name
identifier associated with and designating a *type*

[IEC/PAS 61499-1:2000, 1.3.2.81]

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

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

NOTE 2 Variables may be classified as input variables, output variables, and internal variables.

[ISO modified[3]]

3.2 Definitions based on IA/IM-channel

3.2.1
actuation (measurement) channel
sum of all the items necessary to perform each actuation (measurement) as users need it. The physical composition extends from the attachment to the process, to the valve, motor, actuator (sensor, transmitter), the network, and the complementary processing in the computers

NOTE The expression IA/IM-channel means intelligent actuation/measurement solution of all the requirements for each actuation/measurement necessary. Intelligent here means provided with all the functionalities as users need them.

3.2.2
system (or channel or device) status
actual health (or condition) of the related item (system or channel or device). In other words it is defined at the several levels of system distribution: the system as a whole, each IA/IM-channel of the system, each device composing the channel

NOTE A detailed explanation is given in 6.2.

3.2.3
validity index (VI)
qualifier of the information to which it is added. It can be seen as a quality index

NOTE A detailed explanation is given in 6.3.

3.2.4
measurement uncertainty
parameter associated with the actual result of a measurement, which characterizes the dispersion of the values that could reasonably be attributed to the measured

NOTE 1 The word "uncertainty" means "doubt", and thus, in its broadest sense, "uncertainty of measurement" means the extent of doubt about the exactness or accuracy of the result of a measurement.

NOTE 2 The uncertainty may be, for example, a standard deviation or the width of a confidence interval.

NOTE 3 The uncertainty can be expressed with data which can be treated mathematically, so that the uncertainty of an indirect measurement can be calculated if the uncertainty of the several component direct measurements are known.

3.2.5
profile
set of one or more base standards and/or ISPs, and, where applicable, the identification of chosen classes, conforming subsets, options and parameters of those base standards, or ISPs necessary to accomplish a particular function

[ISO/IEC 10000-1]

NOTE ISPs may contain normative references to specifications other than International Standards.

3.3 Abbreviated terms

AB	Application Block
AME	Application Management Entity
CHD	Control Hierarchy Diagram
CMM	Control, Maintenance and Technical Management
DCS	Distributed Control System
DFBAP	Distributed FB Application Process
EFB	Elementary FB
FB	Function Block
FRD	Functional Requirement Diagram
HMI	Human Machine Interface
I&C	Instrumentation and Control
IA/IM-channel	Intelligent Actuation/Intelligent Measurement-channel
ISP	International Standard Profile
MCC	Motor Control Centre
MIB	Management Information Base
MGT	Management
PFD	Process Flow Diagram
PID	Proportional, Integral, Derivative
PRIAM	Prenormative Requirements for Intelligent Actuation and Measurement
P&ID	Piping & Instrumentation Diagram
SCADA	Supervision Configuration And Data Acquisition
SM	System Management
ST	Structured Text
VI	Validity Index

4 Engineering requirements

4.1 General

This clause expects from the reader certain knowledge of engineering of a distributed FB system. See Annex A for background information. It is designed to give the reader an overview of the life of a FB-oriented control system from conception through design and engineering and onto operation support and maintenance. Each of these phases has different environments for the actual FB entities and their own specific requirements.

4.2 Requirements for design phase

a) To identify a FB as part of a particular functional requirement diagram (FRD) or a certain application blocks (AB) in a distributed field device system, a FB is required to be able to carry an identification of a particular FRD block.

NOTE This is required to be a parameter and may be called STRATEGY. On the basis of this parameter an engineering system may be able to identify the distributed FBs which are combined in a FRD as one FB for reverse engineering purposes.

b) To identify a FB type a type name is required. This type name is required to be unique within a library but can have multiple instances. An engineering tool may navigate by this information to an online help file.

c) To identify a device that hosts one or more FBs a device type identifier is required that allows a link of a device description type to this device type. This device type identifier is required to be based on a profile and not on a vendor-specific type to support interchangeability during design phase.

d) Graphical representation of a device type is required to be referenced within device description as an option. That icon is not used for FB chart nor for P&ID, it is only usable within a topological view of field devices. There is no requirement of this representation within this standard.

e) To identify the elements of control hierarchy diagram (CHD), which are comparable with IEC 61512-1 BATCH processes, some parameters are required that are defined according to this batch standard.

   Example:

   FBs carry these parameters. There is no algorithm necessary within a FB. The EFBs do not carry these parameters.

   1. Batch ID (BATCH_ID)
   2. No. of recipe unit procedure or of unit
   3. No. of recipe operation
   4. No. of recipe phase

f) FB invocation is required to be supported by appropriate scheduling mechanisms in cyclic manner.

g) There is no requirement for field devices and their FBs based on ABs.

h) EFBs are required to be defined and gathered within a library. An EFB is a repetitive logic-mathematics treatment, which is embedded into a FB or a function as defined in the IEC 61131 series. An EFB is a processing module restricted to the process control domain. An EFB cannot be split. All FBs (not functions) are required to carry a Library Name. The combination of a FB name and a library name is a unique identifier of a FB. That means to identify this FB type when instantiated in a device mixed with FBs of branch or application-specific libraries it can be identified as an IEC EFB. This is necessary for version control, link to online help and so on. A formal internal description of the behaviour of the EFB is required to be specified in an IEC 61131-3 language, for example, structured text (ST), ladder diagram or instruction list. Missing elements within ST have to be programmed in several statements and may lead to an additional parameter, for instance, exception handling results.

i) For human machine interface, it is necessary to select the dynamic parameter of an EFB, which builds the information base to visualize the process, to allow operation by changing a parameter and to store data. For this selection, there is no syntax defined in the IEC 61131 series. As an addition to IEC 61131 for process control, attributes within the comment field of definitions within ST are defined.

j) For maintenance and technical management tools, it is necessary to specify or predefine attributes to the parameter and FBs of EFB. For this selection there is no syntax defined in the IEC 61131 series. As an addition to IEC 61131 for process control, attributes within the comment field of definitions within ST are defined.

5 Compatibility levels

5.1 General

There are certain levels of compatibility and according levels of cooperation between FB-based devices. The levels are dependent on well-defined communication and application features. See Figure 3 and Table 1.

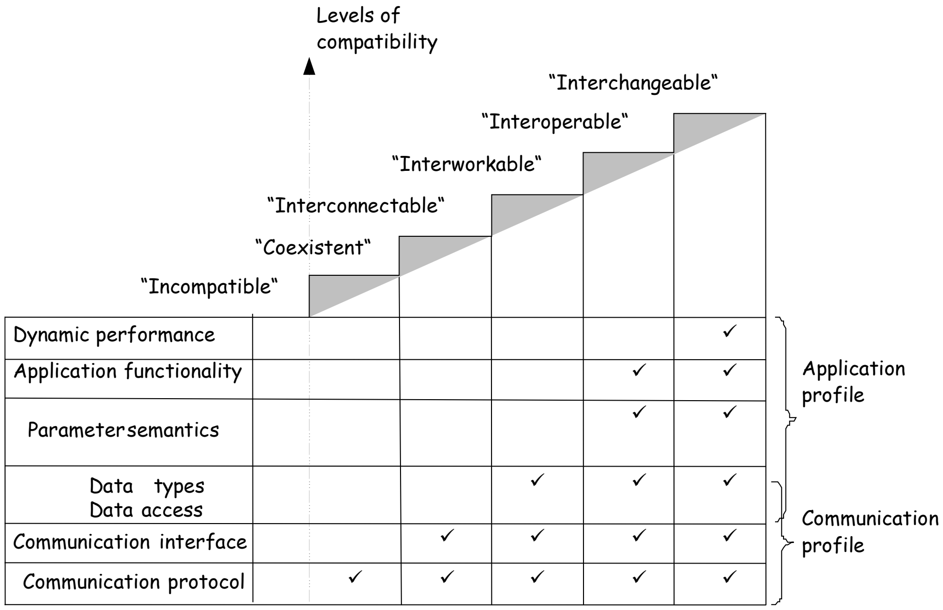


Figure 3 – Levels of functional device compatibility

The following main features are used for the definition of compatibility levels.

Table 1 – Functionality features

Feature	**Description**
Communication profile	
Communication protocol	This feature is defined by all protocols of layer 1 to 7 of the OSI reference model, i.e. from the physical medium access to the application layer protocol
Communication interface	This feature is defined by the communication service definition of application layer including the services and the service parameters. Additional mapping mechanisms can be necessary. The dynamic performance of the communication system is part of this feature
Application profile	
Data types	This feature is defined by the data type of the block data input, data output or parameter)
Parameter semantics	This feature is defined by the characteristic features of the data (for example, this can be data name, data descriptions, the data range,the substitute value of the data, the default value, persistence of the data after power loss and deployment)
Application functionality	This feature is defined by specifying the dependencies and consistency rules between the variables inside the blocks. This is done in the data description part or in a separate behaviour section
Dynamic performance	This feature is defined by time constraints which influence the data or the general device behaviour. For example, the update rate of a process value can influence block algorithms

Regarding these functional features, the following compatibility level names are used for the classification of devices. The IEC 61804 standard for distributed FB applications provides for, but does not require, coexistence, interconnectability, interworkability, interoperability and interchangeability between devices using FBs from different manufacturers. This allows the user to choose a device as part of a new system, or as a replacement, and to understand the consequences of that choice.

5.2 Incompatibility

Inability of two or more devices to work together in the same distributed application.

NOTE Incompatibility can result from differences in application functionality, data semantic, data types, communications interface, or even communications protocols used by the affected devices. Incompatible devices may even interfere with, or prevent, each other's proper communication or functioning (possibly even destructively), if placed in the same distributed application network.

5.3 Coexistence

Ability of two or more devices, regardless of manufacturer, to operate independently of one another in the same communications network, or to operate together using some or all of the same communications protocols, without interfering with the functioning of other devices on the network.

NOTE It is not necessary to have an agreement regarding the communication services. Application- and system-specific programming in one or both devices is generally required in order for coexistent devices to work together in the same distributed application

5.4 Interconnectability

Ability of two or more devices, regardless of manufacturer, to operate with one another using the same communications protocols, communication interface.

NOTE The devices allow data exchange without agreements about the data types. A data type conversion may be necessary. Unique application-specific programming in one or both devices is generally required for interconnectable devices to function together in the same distributed application

5.5 Interworkability

Ability of two or more devices, regardless of manufacturer, to support the transfer of device parameters between devices having the same data types of the data inputs, data outputs and parameters.

NOTE If a device is replaced with a similar one of a different manufacture, it can be necessary to reprogram the application. The distributed application must be designed to accommodate the unique functionality and dynamic responses of the interworkable devices used in the implementation.

5.6 Interoperability

Ability of two or more devices, regardless of manufacturer, to work together in one or more distributed applications. The data input, data output, parameters, their semantics and the application-related functionality of each device is so defined that, should any device be replaced with a similar one of different manufacture, all distributed applications involving the replaced device will continue to operate as before the replacement, but with possible different dynamic responses.

NOTE Interoperability is achieved when both a field device and a system support the same combination of mandatory and optional parts of the same standard. Manufacturer-specific extensions in field devices or systems from different manufacturers may prevent interoperability.

5.7 Interchangeability

Ability of two or more devices, regardless of manufacturer, to work in one or more distributed applications using the same communications protocol and interface, with the data and functionality of each device so defined that, if any device is replaced with another of the interchangeable devices, any distributed applications involving the replaced device will continue to operate as before the replacement, including identical dynamic responses of the distributed applications.

6 Functional requirements

6.1 General

The functional requirements, which are expressed in the following subclauses, are defined at an abstraction level suitable to be well understood by both end-users and vendors.

This abstraction level differs from the FB abstraction level, which is directly used by vendors and system integrators.

The correspondence between the functional requirements defined in this part and the FB concept defined in IEC 61804-2 is not necessarily one-to-one. It may be one-to-one, one-to-many or many-to-one.

Annex B is required to be read for a more complete understanding and use of the following requirements.

The description of the device functional requirements explicitly addresses this functionality of a measurement channel which is located in the field device (for example, sensor, transmitter, etc.). It is required that similar requirements are valid for the field device part of the actuation channel (valve, actuator, positioner, etc.).

6.2 System (or channel or device) status

In principle, there are three main values of the status (as shown in Figure 4), whatever the qualified item:

a) validated capability;

b) acceptable degradation(s);

c) out of service (unacceptable degradation).

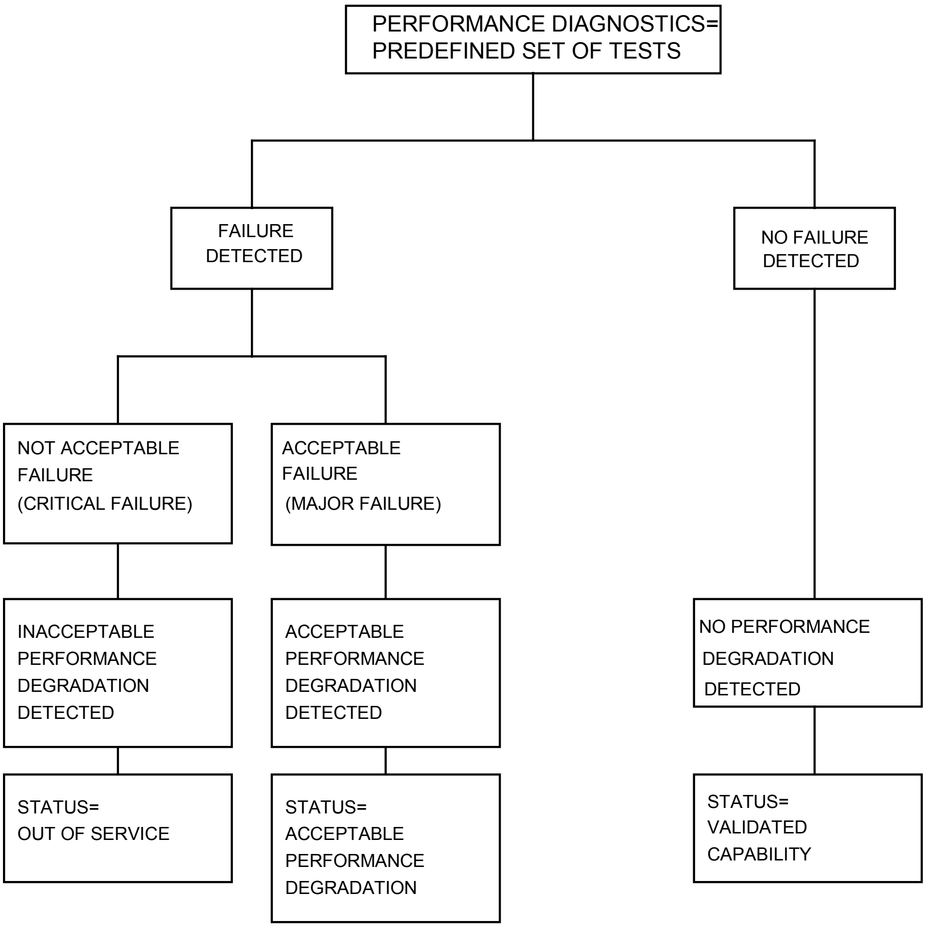


Figure 4 – Device (or channel or system) status

The degradation(s) are defined as (a number of) performance degradations which are considered acceptable by the user of the system (or device) according to his specific needs and the supporting diagnostics.

For practical purpose users need two kinds of information on this status: only one synthesized status and several detailed statuses, each one as required for each task of the several operators.

a) Synthesized status

The synthesized status expresses the actual degree of capability of the item (device or system or channel) to perform the required functions. This information is intended to support immediate actions specified by the users for the control, maintenance and technical management of the systems.

b) Detailed status

The detailed status information makes explicit the detailed diagnostic information needed for the maintenance and technical management of the system (or device). In the user vision at least three needs are required to be covered, as follows:

1) detailed information to document the behaviour of each replaceable component as needed to guide the intervention on the component;
2) information which helps diagnose the faulty part and provides all the details which are useful to properly repair that part (this diagnosis usually requires more details to be documented);

3) as 1) and 2), but here the details are those needed to properly judge the behaviour of each component for technical management purpose.

6.3 Validity index (VI)

The quality is a relative quality, judged on the basis of achievement of the predefined requirements for the properties of the channel generating the information. A basic set of quality criteria to be considered is

a) accuracy and uncertainty criteria (see ISO Guide to the expression of uncertainty in measurement),
b) timeliness criteria,
c) good/bad criteria,

In practical implementations, the criteria used to generate the VI have to be explicitly stated.

The VI is intended to allow immediate and correct use of the qualified information.

The VI is generated in real time taking into account the results of the actual diagnostic and validation of the whole channel that has produced the information. Therefore the VI explicitly makes the user of the information aware of the actual situation (deterministic VI) resulting at varying operating conditions, normal and abnormal, after an error detection and correction.

6.4 Signal processing

Functions producing raw measurements and data as inputs for measurement information processing and device management.

a) Sensor limit thresholds (diagnostics dedicated to fault management support).
b) Sensor ranging.
c) A/D converter control.
d) Data quality determination (VI associated to the raw measurement after the A/D converter due to the results of hardware diagnostics).
e) Influencing quantity compensation (hardware).
f) Process attachment and sensor diagnostics (run time auto tests).
g) Support for maintenance tests on demand.

6.5 Measurement information processing

6.5.1 General

This processes information provided by the measurement signal processing.

The main purpose of the information processing component is to provide the higher levels of the measurement channel with the processed measurement closely associated with its VI. The complete semantics of the measurement and validation information delivered by the measurement device is required to be known. (All the functional units that have taken part in its elaboration are required to be known.)

Validation is an essential part of the measurement processing and determines the degree of credibility of the measurement delivered by each step in the measurement processing. The validation of the information achieved incrementally at each step of elaboration is a fundamental part of the measurement itself in order to allow a correct use of the measurement information to which it refers.

Another important aspect is that, through the convenient use of the VI and of the device status, the device can work under degraded conditions, leaving to higher levels of management the decision on the use of the data. In this sense the fault tolerant behaviour of the transmitter consists of the diagnosis of the fault and in the indication on how much this impairs the measurement (VI) and the device capabilities (device status).

The functions that have to be covered by the measurement information processing are given hereafter.

6.5.2 Fault-tolerant behaviour processing

Fault identification and confining. Information contribution to the processing of the corresponding measurement VI.

6.5.3 VI processing

a) Measurement validation

   This processing has to be accomplished along with each phase of the measurement processing and not separately, in such a way to guarantee mutual consistency between data and VI.

   It may include treatment of time redundancy based on comparison among a number of successive samplings of the same quantity successively taken within the sampling period defined for the application.

b) Uncertainty processing

   The parameter (uncertainty) may be, for example, a standard deviation (or a given multiple of it), or the half-width of an interval having a stated level of confidence. Uncertainty of measurement comprises, in general, many components. Some of these components may be evaluated from the statistical distribution of the results of series of measurements and can be characterized by experimental standard deviations. The other components, which also can be characterized by standard deviations, are evaluated from assumed probability distributions based on experience or other information. It is understood that the result of the measurement is the best estimate of the value of the measurand, and that all components of uncertainty, including those arising from systematic effects, such as components associated with corrections and reference standards, (that) contribute to the dispersion.

6.5.4 Measurement channel processing functions

a) Scaling
b) Linearization
c) Influencing quantity compensation (software)
d) Filtering
e) Dumping
f) Engineering units conversion
g) Limit thresholds (regarding process alarm application) and all relevant data treatment
h) Transformed measurement processing
i) Measurement time stamping
j) Measurement safe values (generally by replicating the last valid value for an assigned number of samplings)
k) Measurement trend processing

NOTE There may exist additional functions.

6.6 Device diagnostics and test support

6.6.1 General

The following subclauses concern functions producing information related to device status.

6.6.2 Device diagnostics

a) Power-on self-tests

b) Run-time self-tests

6.6.3 Maintenance-related processing signals and counts all events useful to perform device degradation analysis

The target is to allow the implementation of algorithms in the maintenance systems to foresee the degradation of a device or of a part of it. The manufacturer has to define which parameters are significant and include in the device the necessary FBs that recognize these situations and report their occurrence to the maintenance system and to the technical management system.

Examples of these parameters that have been found useful include

a) the time of operation spent outside nominal working conditions (defined through for example temperature limits, pressure limits, etc.),

b) the number of abnormal events (such as pressure spikes for a P, DP transmitter),

c) the number of electrical shocks of defined characteristics.

NOTE There may exist additional parameters.

6.6.4 Support for tests on demand (remotely or locally issued)

This support, as needed for actuation and measurement remote or local tests (for example calibration), is divided in the two following categories:

- intrusive tests: the execution of these tests interferes with the normal operation of the device/channel (for example it requires that the attachment with the process is switched off);
- non-intrusive tests: the execution of these tests does not interfere with the normal operation of the device/channel.

According to the requested tests, the necessary authorization is needed, and the device/channel have to be put in the appropriate operating mode.

The required functionalities are

a) intrusive device tests execution:

   1) support for calibration procedures (both for primary and transformed measurements)

      The aim of this function is to give the maximum help to the maintenance operator responsible for device calibration. As he will perform calibrations on many different devices of many different manufacturers the help he needs is achieved through

      – use of standard terminology where already available,

      – application of a standard procedure when applicable,

      – direct availability of calibration reports and results in electronic form in order to be transferred to the maintenance system through the network or electronic support.

   2) measurement processing tests against reference inputs;

   3) process attachment intrusive diagnostics tests;

b) non-intrusive device tests execution:
   1) process attachment non-intrusive diagnostics tests;
   2) measurement processing tests against reference inputs (if executable at run time during time periods agreed upon by the control operator; for example, test of a redundant part at a time);
c) tests results retrieval:
   1) retrieval of device tests results (information about self-diagnostics, test results and maintenance interventions);
   2) direct availability of actual information about self-diagnostics, test results and maintenance interventions in electronic form in order to be transferred to the maintenance system through the network or electronic support.

6.7 Local interfaces attachment

Availability of a local access, through a suitable terminal, to

a) access rights management function (see 6.8.10),
b) pre-defined list of information,
c) pre-defined command inputs,
d) required local access to support field operator interventions.

6.8 Device (and system and channel) management

6.8.1 General

The following functions are described as the expected results of the cooperative actions of the management functions of the device and those of the system manager; of course, the contribution of the device agent manager belongs to the device functional units.

In this perspective, a channel is a subsystem.

6.8.2 Device identification function

a) Vendor related device information retrieval. Information as needed by the several device-users interacting with the device along its life cycle (for example, device components part number, materials, software and hardware versions, working ranges, etc.).
b) User-related device information retrieval (complementary information usually written during commissioning, for example, device tag, installation date, etc.).

The above information is expected to be written in the device in the relevant phases of the life cycle by the vendor operator and the commissioning operator using the following functionality:

a) vendor-related device information modification;
b) user-related device information modification.

6.8.3 Configuration function (and help for system configuration)

When the transmitter is programmable, the software for the specific functions in the processing and acquisition parts of the transmitter has to be downloaded from the configuration system. For instance, if a DP transmitter is used as a level or flow transmitter a download of the specific software is needed. Of course, this logical action may have several practical implementations (for example, EPROM replacement, selection of pre-installed functions, etc.).

Configuration will include all the software components needed for supporting the different tasks assigned to the transmitter such as measurement, diagnostics, test routines, etc. A specific user requirement is the availability of reports on demand, which allow the operator to check the actual configuration (consistency check versus a specified configuration). Along the device life cycle the different users will take advantage of the following functionality:

a) vendor configuration modification;
b) user configuration modification (selection of needed functions between the available ones);
c) configuration retrieval;
d) read available functions;
e) read vendor configuration revision;
f) read user configuration revision;
g) read selected (active) functions.

6.8.4 Parameterization function

These functions consist in setting parameters to completely define generic functions in order to satisfy the plant application constraints.

The processing functions parameterization depends on the functions selected/downloaded during configuration; parameters are required to be loaded in non-volatile parameter support and their integrity is required to be continuously checked.

Parameters to be set are, for example, the range, the offset and the engineering units of the measured variable, warning and alarm levels, sampling frequency, filters time constants, etc. A specific user requirement is the availability of reports on demand, which allow the operator to check the actual parameterization (consistency check versus a specified parameterization).

Device parameters are divided into the following two categories:

– intrusive parameters: the modification of these parameters interferes with the normal operation of the device;
– non-intrusive parameters: the modification of these parameters does not interfere with the normal operation of the device.

According to the requested parameter modification request, the necessary authorization is needed, and the device/channel has to be put in the appropriate operating mode.

The required functionality is

a) intrusive device parameterization modification (single parameter/for defined groups of parameters),
b) non-intrusive related device parameterization modification (single parameter/for defined groups of parameters),
c) device parameterization retrieval.

6.8.5 Measurement management function

This function is part of the device management. Its relationship with the system management is clarified hereafter.

This function produces proper execution events as needed for each kind of measurement execution timing. The needed synchronization among the several device managers is produced by the system manager, by means of suitable network mechanisms.

It produces the following services according to the needs.

a) Execution event for cyclic asynchronous measurements (i.e. not synchronized with a system time reference)

   This is the execution event needed for the measurement to be cyclically acquired asynchronously with respect to its use and to be cyclically made available to all its users within the times necessary to guarantee data consistency for the specific application

b) Execution event for cyclic synchronous measurements (i.e. synchronized with a system time reference)

   This is the execution event needed for the measurement to be cyclically acquired synchronously with respect to its use following the reception of sync. Command and to be made available to all its users within the times needed for the specific application

c) Cyclic measurements with asynchronous start and stop

   This is the execution event needed for the measurement to be cyclically acquired asynchronously with respect to its use and to be cyclically made available to all its users within the times necessary to guarantee data consistency for the specific application

   Acquisition and distribution of measurement are not continuous but are started and stopped following a specific command or an asynchronous event

d) Execution event for measurements on demand

   The measurement is acquired following a specific command issued by the user of the measurement. The measurement will be made available after a command of data request issued by the user of the measurement

e) Execution events for detection of logic values variations

f) Event notification of value modification of binary data

6.8.6 System time management function

The system time management function is aimed at coordinating the several time managers of each device.

In some applications it is, for example, required to time-stamp measurements and/or events with an absolute time information. A system (master) clock is required and all the devices are required to have an internal clock which is required to be maintained synchronized with the master clock. The accuracy required by the specific application is required to be guaranteed through a well-defined combination of clocks accuracy and a synchronization operating through the network.

It coordinates the managers of the several devices, as needed, to share

a) absolute time, with the requested time accuracy (typically 1 ms),
b) relative time, with the requested time accuracy (typically at least 1 ms),
c) time synchronization command, sent by the master clock at a defined rate as needed to guarantee the requested time accuracy.

6.8.7 Timeliness verification support

This function checks that data transmitted, or received, via the communication interface are always within the specified timeliness.

This processing contributes to the processing of the data validation.

6.8.8 Device failure management

The device failure management is aimed at supporting proper management of device failures.

a) Detection of failures

It takes into proper account all the detailed diagnostic information made available by the device diagnostics and test support. It uses this information to carry out activities of failure detection and to generate the device statuses (device detailed status and device synthesized status) as needed for specified internal actions and to be treated by the measurement channel status management function. This latter function will process all this information to provide the different users with the information required for the specific application (it is an end-user adaptation).

b) Device decisions (fault-tolerant procedures)

It carries out internal procedures aimed to confine detected faults (to prevent local failures from causing system failures) following requests received from a higher level of management or to be automatically activated inside the device.

It includes procedures to treat possible redundancy of sensors and attachment to the process.

6.8.9 Operation management function (extended for access rights management)

This is a system management function aimed at managing the transitions among the several operating modes. In particular, it manages the qualification transfer procedures between the different human or automatic operators. This is to support procedures of intervention in the field devices in order to realize intervention plans that are agreed among the human organizations. Of course, the device management is required to cooperate with the system management to achieve the result.

All the paper procedures that are used in a plant in order to achieve this function may now be simplified because of the new communication means between the devices and the intelligence embedded in the devices themselves.

It is required that for each needed interaction between the operators and the distributed system the device management enables the needed device functions and disables all the others. A management procedure has to be established in order to give the different operators the qualifications to access functions or devices as defined on the basis of agreed needs. This is intended to prevent unauthorized action and possible errors (see qualification access).

Typical operating modes, defined according to the several needed interactions, to be supported by the system management are

a) control remote auto (associated operator: control device),
b) control remote manual (associated operator: control remote operator),
c) control local (manual) (associated operator: control local operator),
d) maintenance remote (associated operator: maintenance remote operator),
e) maintenance local (associated operator: maintenance local operator),
f) commissioning remote (associated operator: commissioning remote operator),
g) commissioning local (associated operator: commissioning local operator),
h) parameterization (associated operator: parameterization operator),
i) self (associated operator: none; the device has lost the fieldbus connection).

NOTE There may exist additional operating modes.

6.8.10 Access rights management function

The access right management function is the qualification check on every access to normal device functions.

In each operating mode the devices themselves qualify which operators are allowed to perform on them certain pre-defined actions.

It is obvious, for example, that it is extremely important to disable the write access to the device for configuration, parameterization and tests on demand purposes when the measurement processing functions is required to be active (such as in control operating modes).

This device requirement is the complement of a corresponding requirement at the system level.

7 FB application requirements

7.1 System overview

The preceding clauses describe what are the requirements to a system from the end-user point of view. This clause focuses on the architectural point of view of a system.

Process control systems are combinations of applications controlling the process (the FB based applications), applications carrying out maintenance, monitoring, HMI, commissioning functions and applications for the configuration functions of the application software and hardware in a device, also known as system management functions see Figure 5. A device performing the process control in terms of FB application has to provide all functionality to interact with the FB devices each other and the external applications in an interoperable way. Therefore the components of the system and the devices, including their relationships, have to be defined. This subclause covers only the FB-based devices. The device and system architecture is based on the definitions in IEC 61499-1.

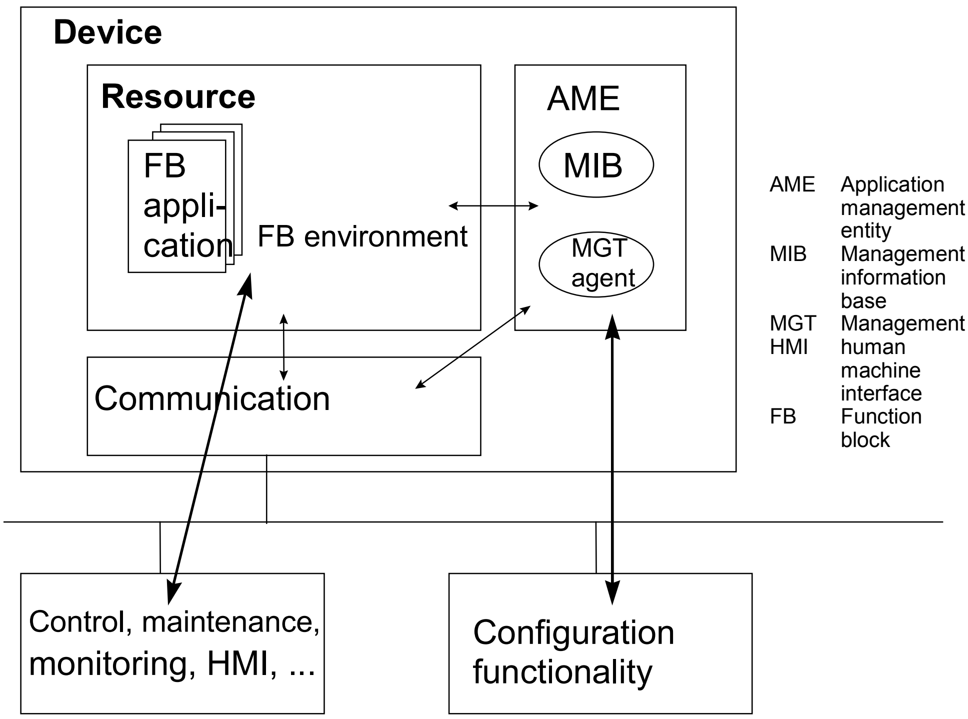


Figure 5 – Components of a FB device

In a real device or resource, there are only data and function (program) codes as well as interfaces to the communication, process and HMI. Only via the communication system, by a configuration or design tool, these data and program code are seen as FBs, management agents or data connections. The specification of the device model is an external view to the device implementation. The device model describes details of the overall system architecture. The device model is an abstraction of the real device.

– The basic components of a device performing FB applications are the FBs (see Figure 6) .

– FBs are composed of data and event inputs and outputs, internal data, contained data and algorithms. They have the following general requirements.

NOTE The term FB is used as a synonym for blocks in general (function, device and technology blocks) as well as for FB in particular.

Further description is given in 7.3.

a) FB environment

FBs are embedded in an operating environment shown as FB environment. The FB environment performs special process control functions, which are a priori available application portions without any programming design activities, for example alarms. These FB environment functions are used by external applications independent of, i.e. parallel to, the FB application (see 7.5).

NOTE Common functions and data of the FB environment can be encapsulated in FBs too, for example trending.

b) Application management entity (AME)

Many process control devices need adoption to the used hardware configuration from the functional point of view. For instance, the available FBs in a modular device have to be configured or the model version with or without an additional input for measurement compensation. This causes the addition or deletion of FBs. These functions will be initiated by the so-called AME, i.e. addition functions in a device modifying their application software. The IEC 61804 series does not define the AME. This shall be done within the framework of the IEC 61158 series. Annex F describes the management requirements .

NOTE The AME contains the so-called system and network management functionality.

c) Communication

The data transfer between FBs, the interactions between FBs/FB environment and maintenance, HMI monitoring and the interaction between FB environment/AME and configuration functions have to be supported by the communication services and protocols. In the IEC framework, the communication system is within the scope of IEC 61158 series and IEC 61784. The specification of the FBs and the FB environment is independent of the communication system. A mapping sublayer between the FB environment and communication system will adapt the application and the communication.

Parameters, blocks, objects, and functions in the FB application process and the system and network management application process (i.e. AME) are required to map efficiently to the underlying communications protocols and services. Communications services and protocols used with the FB application process and the system and network management application process are required to be specifically designed for use with distributed application processes, and are required to provide the services required by those applications.

Three different types of communication requirements are required to be considered:

1) Time-critical communications

Communications services and protocols used with the FB application process are required to support the unique requirements of time-critical communications, including

i) deterministic transfers,

ii) spatial consistency of transfers,

iii) temporal consistency of transfers.

2) Demand communications

Communications services and protocols used with the FB application process are required to support the unique requirements of time-available communications, including segmented transfers of large data blocks. Communications services and protocols used with the FB application process are required to also support the unique requirements of “report by exception” communications.

3) Event communication

Communications services and protocols used with the FB application process are required to support the unique requirements for communicating events, and in particular be designed to

i) minimize loading during quiescent periods,

ii) prevent overloading during high-activity periods.

d) Resources

A resource is considered to be a logical subdivision within the software (and possibly hardware) structure of a device. Resources have independent control of their operation. The definition of a resource may be modified without affecting other resources within a device. A resource accepts and processes data and/or events from the process and/or communication interfaces and returns data and/or events to the process and/or communication interfaces, as specified by the applications utilizing the resource. An interoperable network view of applications is provided through device resources. Each resource specifies the network visible aspects of one or more local applications (or parts of distributed applications).

7.2 Overview of basic FB types

7.2.1 General

The basic FB types that may be accessed through their associated resources are listed below (see Figure 6). They are

- resource block;
- technology block;
- FB;
- view block;
- trend block and
- alert block.

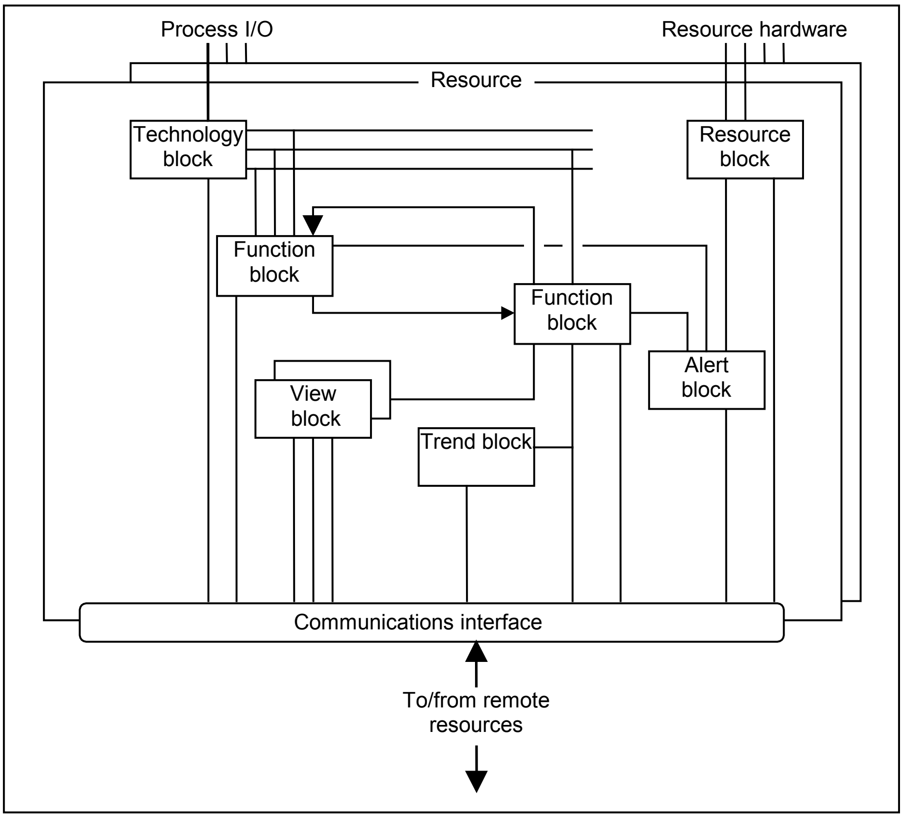


Figure 6 – Device structure for a FB application for the process industry

7.2.2 Resource block

The characteristics of the physical subcomponent associated with a resource may be described by a set of parameters which are included in a resource block. The resource block may also contain variables that are common to FBs and technology blocks for example set fail-safe. These variables are defined in the device block (see 7.4.4). The terms device block and resource block are used interchangeably.

7.2.3 Technology block

Technology blocks insulate FBs from the specifics of I/O devices, such as sensors, actuators, and switches. Technology blocks control access to I/O devices through a device independent interface defined for use by FBs. Technology blocks also perform functions, such as calibration and linearization, on I/O data to convert them to a device-independent representation. Their interface to FBs is defined as one or more implementation independent I/O channels.

7.2.4 FB (function block)

The FB is the primary means of defining monitoring and control in a FB application. FBs represent the basic automation functions performed by an application, which is as independent as possible of the specifics of I/O devices and the network. Each FB processes input variable and technology block input according to a specified algorithm and an internal set of contained variables. They produce output variable and output to technology blocks.

Based on the processing algorithm, a desired monitoring, calculation or control function may be provided. The results from FB execution can be reflected in parameters which are included in the function block for operation or diagnostic information. In addition, processing results may be reflected in the output to a technology block or to one or more output variables that may be linked to other FBs (see 7.3).

7.2.5 View block

The FB environment includes data structure definitions, called view blocks, to allow access to related block parameters as a group. View blocks facilitate fast operator display response when viewing FB data.

For each FB type, view blocks are defined for each of the following block parameter groupings:

a) operations dynamic parameters;
b) operations static parameters;
c) all dynamic parameters;
d) other static parameters.

7.2.6 Trend block

FBs are required to include data structure definitions, called trend blocks, to allow access to a sequential set of time-stamped samples of a single block parameter as a group. Trend blocks eliminate the communications and system processor overhead required for scanning parameters at a fast rate for trending.

Trend blocks definitions are required to include the definition of standard sample types and their associated sampling functions (for example, spot sample, integrated average, minimum, maximum, etc.).

7.2.7 Alert block

An alarm is the detection of a block when it leaves a particular state and when it returns to that state. The time at which the alarm state was detected is included as a time stamp in the alert message. Also, the priority is included to indicate whether this is an advisory or critical alarm.

The FB environment is required to include data structure definitions, called alert blocks, and associated resource and FB functions to allow the controlled transfer of alarm and event information within the system. Alert blocks predictably and efficiently route alarms and events to a selected destination (or destinations) within the system.

Alert block definitions include the definition of standard exchange protocols for initiating, sending, and confirming alert block reports, acknowledging alerts, interpreting standard and custom reason codes, and configuring alert functions such as alert key, alert priority, alert auto-acknowledge, etc.

Alerts are used by resource, technology and FBs to communicate notification messages when alarms are detected.

7.3 FB requirements

7.3.1 FB type specifications

The process FBs are required to include

a) standardized data structures,
b) standardized semantic meaning,
c) common behaviours associated with standardized data,
d) standardized definitions for basic input, output, and control functions.

7.3.2 FB normal and abnormal operation requirements

A FB performs normal and abnormal operation (see C.2.6). The normal operation of a FB is carried out under positive conditions of the process and automation devices during stable operation (operating point). The initialization, the warm or cold start as well as more advanced operations like going in a safe position are additional functions of the process control FBs. These operations are fixed parts of the FBs that are default linked to the associated events. These operations are known as abnormal operation or exception handling.

7.3.3 FB functional requirements

FB functional requirements valid for both discrete and continuous control are

a) the instances name of FBs will be defined as the block tag. Tags are unambiguous within the scope of a system at one plant site. Tag names may have syntax rules that are not compatible with control language rules (for example, length, special characters);

b) a status which is hard combined with the main process signal flow through the FB application, i.e. the FB inputs and outputs, not the contained parameter, are required to have status;

c) FBs intended for use in cascade control structures are required to include back calculation parameters for inputs and outputs to carry defined standard cascade initialization, output limitation, and input or output value status information;

d) FB functional specifications are required to include defined standard “bumpless” control-mode shedding and recovery actions on detection of bad input, output, or transfer values, or on detection of initialization requests or control limits from back calculation parameters;

e) FB functional specifications are required to include defined standard fail-safe actions on detection of bad input, output, or transfer values, or a FB application process resource fail-safe command. When a block executes its fail-safe action, it is required to also send an alert message via an alert block. The fail-safe alert message is required to include a reason code that identifies what triggered the fail-safe action;

f) FB functional specifications are required to include defined standard setpoint and output tracking behaviour. FB functional specifications are required to include defined standard bias and ratio set-point ramping behaviour;

g) FBs are required to include parameters to simulate inputs and/or outputs and their status (“simulate parameters”) while viewing the actual inputs and/or outputs and their status, along with a parameter to enable and disable each simulation parameter. The FB specifications are required to include defined standard behaviour for the simulate parameters and for the simulate enable/disable parameters. The defined standard enable/disable behaviours are required to accommodate a mandatory simulation disable hardware jumper or switch;

h) FBs are required to include parameters and standardized functional definitions for process measurement (PV) and deviation (DV) alarms;

i) FBs are required to have mode, which controls the FB internal flow of information from inputs to outputs and the variations of the algorithms of a FB;

j) FBs are required to include standardized functional definitions for initialization and restart for each block. Behaviour is required to be defined for the following operating circumstances:

   1) new device;
   2) cold restart (extended power failure);
   3) warm restart (short power failure);
   4) return from device fail-safe.

7.4 Initial sets of FBs derived from I&C

7.4.1 Minimum set of FBs derived from ABs (rich FBs)

The following examples give the minimum set of FBs for process automation (see Table 2) required to ensure integrity of these lower level controls and to ensure that a large section of the experience embodied in the last 10 years of fieldbus standardization effort is not wasted. Of course, the minimum set allows extensions.

Table 2 – Example of initial FB set

FB name	Description	Why included in the minimum list
Loop control (Proportional Integral Derivative – PID)	PID with configuration for proportional on error, proportional on measurement change, velocity and position outputs Interactive and non-interactive algorithms include simple set-point ramp option including ratio bias on the set-point input Include error squared and non-linear option	To ensure all PIDs operate in the same way given the same configuration and tuning parameters. This will help to ensure maximum loops are well tuned. Will promote understanding of need for different tuning sets. Will promote standard tuning tools and methods
Selector for control outputs	Multi-input high/low selector, with initialization and status passing	To allow common override and constraint control implementation with secure initialization and rest wind-up prevention
Selector for measurements	Highest, lowest, average and middle of three measurement selectors	For two out of three voting and extra security on unreliable measurements (for example, analytical)
Splitter	For split range controls	Support of back calculation for initialization and tracking for bump-less transfer, wind-up prevention
Analogue input	Scaling linearization quality checking	To ensure that scaling and conversion work is kept to a minimum and performed in the same way. To allow all manipulations in a control scheme to be done in floating-point engineering units in all implementations
Analogue output	With interlocks, check-back read-back linearization for non-linear valves	To ensure that common nomenclature, fail-safe operation, constraint controls and output checks are performed in the same standard way in all implementations
Output fan-out	Fan-out of controller outputs with auto manual switch	To ensure that common nomenclature, fail-safe operation, constraint controls and output checks are performed in the same standard way in all implementations
Discrete variable/pulse input	Single and multiple status inputs	To ensure common nomenclature, fail-safe operation, checks, counts are performed in the same way in all implementations
Discrete variable output	Single and multiple digital control outputs	To ensure common nomenclature, fail safe operation, checks, counts are performed in the same way in all implementations
Lead lag	First order lag and lead function	To allow common feed-forward and constraint control implementation with secure initialization
Dead time	Table driven dead time	To allow common predictor, feed-forward and constraint control implementation with secure initialization
Control/incremental summer	To sum outputs from incremental algorithms or to combine controller outputs	To allow common feed-forward and constraint control implementation with secure initialization
Control ratio	Ratio outputs of controller with initialization	To allow common feed-forward and constraint control implementation with secure initialization
Input switch	To switch inputs on event	To allow common predictor, feed-forward and constraint control implementation with secure initialization

FB name	Description	Why included in the minimum list
Characterize	Twenty-point interpolated two-way characterize block	To allow simple characterizations such as tank strapping and linearization
Timer	General timer to time functions, count time on, produce delays timed pulses, etc.	To allow simple batch/sequence functions to be implemented with secure initialization
Integrator	Integrator for flow mass power, etc., with reset and trip functions	To allow simple batch functions to be implemented with secure initialization
Continuous variable alert/alarm	General alarm block "hi", "hi hi", "lo", "lo lo" with fail safe actions, dead bands, ignore, count, dynamic deviation, filter time stamps	To ensure common alarm functions are implemented securely with initialization, and audit trails
Discrete variable alerts and alarms	General alarm block for discrete variables with fail-safe actions, invert, group, ignore, count, dynamic deviation, filter time stamps	To ensure common alarm functions are implemented securely with initialization, and audit trails
Device control block	General block for controlling motor driven devices, valves pumps, etc. with start stop open close stopped, started opened closed, travelling, faults statuses.	To ensure common motor control and other discrete output control functions are implemented securely with fail-safe, initialization, and audit trails and to ensure maximum re-use of standard tested functional code
Continuous variable manual entry	General block for operator entry of analogue variable with checking and initialization	To ensure entry functions are implemented securely with initialization, and audit trail
Discrete variable manual entry	General block for operator entry of discrete variable with checking and initialization	To ensure entry functions are implemented securely with initialization, and audit trail
Set-point ramp	General 16-point ramp-soak block for batch-type cycles	To ensure maximum re-use of common code and to enforce initialization and audit trails

Annex D shows an example of modelling an analogue input FB.

7.4.2 FBs derived from EFBs

The EFBs cover logic-mathematical algorithms for the information processing of the variables. In the area of process control, the implementation of these algorithms needs add-ons for safety reasons. The additional algorithms provide default values after cold or warm start of a device, values after initialization or a status of the success of the algorithm (for example, bad at division by zero). The additions are necessary to start application parts that bring the process into a safe state. Generally speaking, every algorithm and process-related variable in the FB application have to be combined with a state providing the confidence of the algorithm execution and variable value. The main algorithm is the equation that determines the calculation of the output. The additions determine the state, default values and others. Table 3 gives an overview of commonly used FB derived from EFBs.

Table 3 – Example of common list of EFBs

			<-similar FBs -->	
Category	Description	Suggested Name	IEC 61131-3	VDI/VDE 3696
arithm., 1 input	absolute value	ABS	ABS	ABS_
	arc cosine	ACOS	ACOS	ACOS_
	arc sine	ASIN	ASIN	ASIN_
	arc tangent	ATAN	ATAN	ATAN_
	cosine	COS	COS	COS_
	exponential function	EXP	EXP	EXP_
	natural logarithm	LN	LN	LN_
	logarithm base 10	LOG	LOG	LOG_

			<-similar FBs -->	
Category	Description	Suggested Name	IEC 61131-3	VDI/VDE 3696
	sine	SIN	SIN	SIN_
	square root	SQRT	SQRT	SQRT_
	tangent	TAN	TAN	TAN_
	dead zone (= dead band)	DEADZ	-	-
	limiter	LIMIT	LIMIT	LIMIT_
	linear scaling	SCAL	-	SCAL_
	Non-linearity (support points)	NL_SUP	-	NONLIN_
	Non-linearity (polynomial)	NL_POL	-	-
	splitter for split-range control	SPLIT	-	(in OUT_A)
arithm., >=2 inp.	Add	ADD	ADD	ADD_
	Divide	DIV	DIV	DIV_
	Exponentiation (in1**in2)	EXPT	EXPT	EXPT_
	Modulo function	MOD	MOD	MOD_
	Multiply	MUL	MUL	MUL_
	subtract	SUB	SUB	SUB_
	average of n signals	AVER_N	-	-
	flow-rate correction by P,T	FCOR	-	Y_FCOR
boolean + edge	boolean and	AND	AND	AND_
	not (negation)	NOT	NOT	NOT_
	boolean or	OR	OR	OR_
	boolean exclusive or	XOR	XOR	XOR_
	falling edge detection	F_TRIG	F_TRIG	FTRIG
	rising edge detection	R_TRIG	R_TRIG	RTRIG
counter, flip-flops	universal counter	CT	CTUD	CT
	bistable (reset dominant)	RS	RS	RSFF
	bistable (set dominant)	SR	SR	SRFF
	greater than, or equal to,	GE	GE	GE_
	greater than	GT	GT	GT_
	not equal	NE	NE	NE_
	less than, or equal to,	LE	LE	LE_
	less than	LT	LT	LT_
	switch (+ alarm or message)	SAM	-	SAM
dynamic and control	running average of one signal	AVER_1	-	AVER
	(filtered) differentiation	DIF	-	DIF
	High-/low-band filter	-	-	(FIO/SEO)
	pulse width modulator	PWM	-	PWM
	rate limitation	RLIMIT	-	-
	second order dynamic	SEO	-	SEO
selection	convert 1-of-n-bit to number	BIT_N	-	BIT_N
	demultiplexer	DEMUX	-	DEMUX_x
timer	boolean off delay	TOF	TOF	TOF1
	boolean on delay	TON	TON	TON1
	boolean pulse	TP	TP	TP1
	dead time	DEADT	-	-
trend storage	registration (trend storage)	R or TREND	-	R

7.4.3 Technology block

a) Technology blocks are special FBs performing all functions necessary to transform physical signals into digital ones and vice versa. That includes analogue-digital/digital-analogue transformation linearization and others. The definition of the technology block functions is accompanied by parameters. The technology blocks represent the actuation and measurement type. All measurement and actuation principle specific functions and parameters are encapsulated in this block.

The functional classification of measurement related FBs is shown in Figure 7.

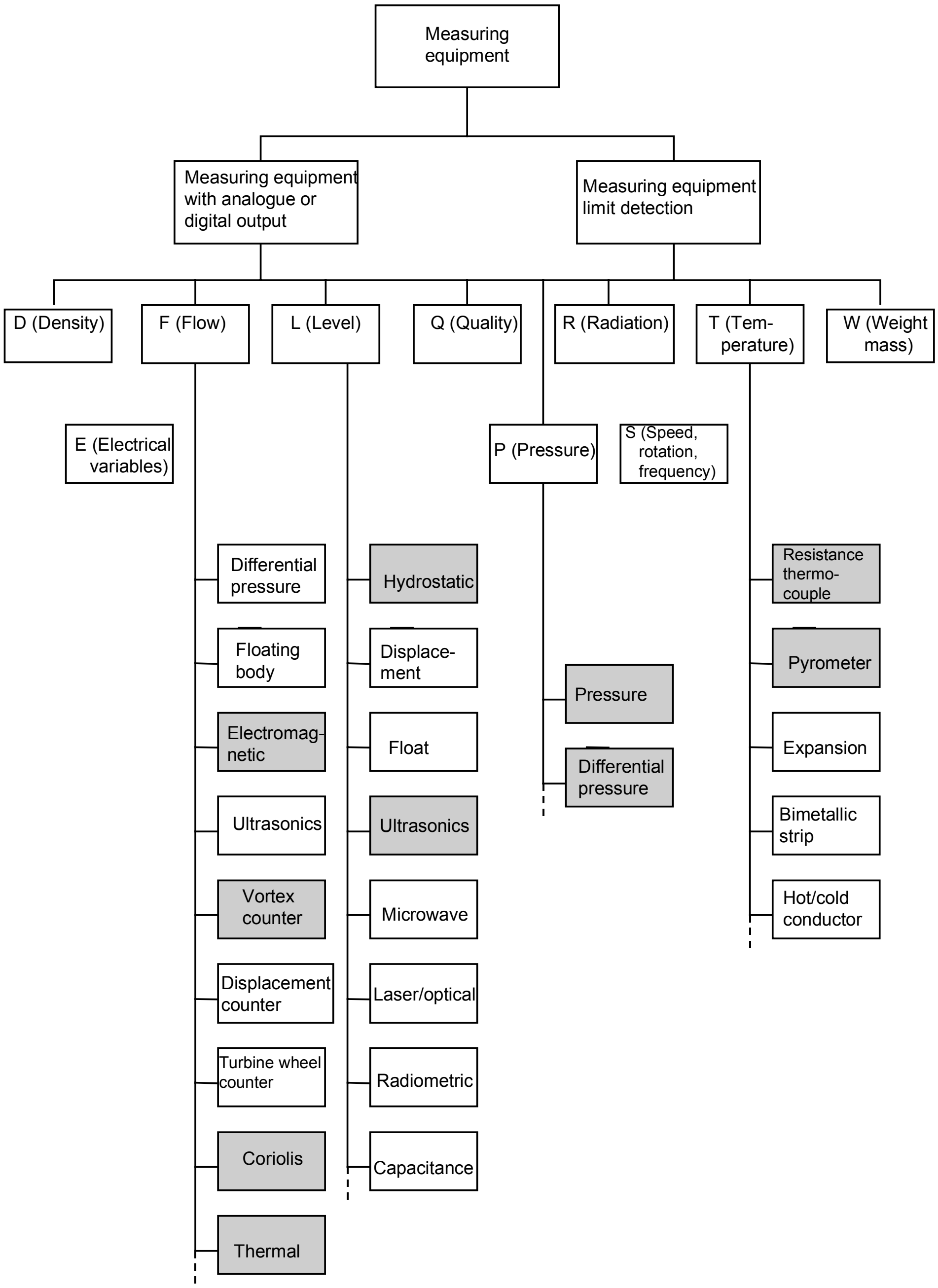


Figure 7 – Functional hierarchy of sensors: example

b) For each path and level of the tree, the characteristics and required properties, parameters and function are required to be pointed out.

NOTE The borderlines between the functional classified FB types are sometime soft. A support is required that allows the users to identify the main purpose of the FBs.

c) The technology block is required to include a CHANNEL parameter, which is required to be used for data connections to other blocks within the resource. The CHANNEL parameter is required to serve as the technology block “tag”, and other parameters within the technology block is required to be addressed as “CHANNEL.Parameter”.

7.4.4 Device block

a) Device blocks are special FBs containing all parameters and functions necessary for the identification of the device itself (device plate), for service purposes like memory space for certification date or maintenance messages. The device block represents the device hardware and software in general.

b) The device block is required to contain a description of the physical part containing the FB application process, including the resource tag reference, the resource device description reference, and the resource object dictionary.

c) The device block is required to contain information needed to manage the FB application resource, including control of the resource state, dynamic memory space allocation, and updating of non-volatile memory.

d) The device block is required to provide indication of optional FB application process features supported by the device and resource, including string encoding (ASCII or Unicode) and input-output hardware types.

e) Device block parameter and functional specifications are required to include a resource fail-safe command which, when set, is required to cause all technology and FBs within the resource to execute their defined fail-safe actions.

f) Device block parameter and functional specifications are required to include a resource fail-safe disable function that, when activated, is required to disable the fail-safe actions for all technology and FBs within the resource. The device block is required to send a fail-safe disabled alert via an alert block (see 7.2.7) and repeat said alert at a user-configured rate, whenever the fail-safe disable function is activated.

g) Device block parameter and functional specifications are required to include a resource isolation timer function. This function, if activated, is required to sum the number of occurrences the FBs in the resource have exceeded their “stale count limit”. When this sum exceeds a configured value before timely communications resume, all technology and FBs within the resource are required to be forced to execute their defined fail-safe actions. This function allows a secondary in a cascade to wait a relatively long time before executing its configured fail-safe action.

7.5 FB environment requirements

7.5.1 Object dictionary

The FB environment is required to include an object dictionary, which contains descriptions for all blocks and objects within the device. Applications external to the device utilize the object dictionary to obtain descriptions of blocks and objects within the device via the communication system, and to determine internal storage indices for parameters.

7.5.2 Link object

The FB environment is required to include data structure definitions and associated functions to map resource and FB parameters to communications relationships, called link objects.

7.5.3 FB services

The FB environment requires that certain services be performed with regard to all communications to and from other FB environments in other resources or devices, in order to provide consistent operation of all devices in the system regardless of manufacturer. They also provide necessary coordination between the FB and system management application processes.

7.5.4 FB schedule

a) The FB environment is required to specify a standard method or methods to control the execution of FBs. The specified method or methods are required to synchronize block execution with
   1) the communication of block inputs and outputs,
   2) the execution of other blocks.
b) Calculation of parameters that control the schedule of communications and FB execution may be carried out for the system by an application outside the scope of the FB environment (i.e. "off-line"). Once a schedule is established, the FB environment is required to control and maintain FB execution in accordance with resource parameters defined by the schedule.
c) The synchronization of block execution with the communication of block inputs and outputs are required to accommodate network transfer times.
d) The synchronization of block execution with the execution of other blocks is required to accommodate device resource loading.
e) The FB environment is required to also accommodate non-standard or "manufacturer-specific" methods to control the execution of FBs within a device.
f) A means is required to be provided to identify the method of FB execution control within a device (i.e. "standard" or "manufacturer-specific"). The identification method is required to be expandable to allow for multiple "standard" execution control methods (i.e. to allow for future additions to the standard).

7.5.5 Revision management

The FB environment is required to specify parameters, functions, and protocols for revision control of all software and configurable parameters within a device. Main parameters are

a) device profile,
b) device profile revision,
c) static data revision,
d) write lock,
e) access permissions parameters.

7.5.6 Maintenance block

The FB environment is required to include data structure definitions and associated resource functions to allow retention in the device and controlled transfer within the system of maintenance information, called maintenance blocks.

Main parameters of maintenance blocks contain the following information concerning device characteristics and maintenance events:

a) wetted parts and material codes;
b) manufacturer-entered text strings;
c) user shop-entered text strings;

d) user field-entered strings;

e) user-defined maintenance activity codes;

f) results of manufacturer-defined interactive calibration procedures.

7.6 Communications requirements

a) Parameters, blocks, objects, and functions in the FB environment and the system and network management application process are required to map efficiently to the underlying communications protocols and services.

b) Communications services and protocols used with the FB environment and the system and network management application process are required to be specifically designed for use with distributed application processes, and are required to provide the services required by those applications.

c) Communications services and protocols used with the FB environment are required to support the unique requirements of time-critical communications, including

   1) deterministic transfers,

   2) spatial consistency of transfers,

   3) temporal consistency of transfers.

d) Communications services and protocols used with the FB environment are required to support the unique requirements of time-available communications, including segmented transfers of large data blocks.

e) Communications services and protocols used with the FB environment are also required to support the unique requirements of "report by exception" communications.

f) Communications services and protocols used with the FB environment are required to support the unique requirements for communicating events, and in particular be designed to

   1) minimize loading during quiescent periods,

   2) prevent overloading during high-activity periods.

8 Additional requirements

8.1 Cooperation with external applications

An overall automation application includes a couple of non-time-critical control operations (for example, monitoring, maintenance and configuration) which may not be specified in terms of FBs. These operations may also be applied in cooperation with non-FB process control systems. The non-FB applications need an on-line data access to the variables and parameters of the FBs as well as to other device specific information see Figure 8[4].

This subclause provides an overview of the key requirements.

[4] The cooperation with non-FB applications can be viewed from two standpoints. For the purpose of this footnote a FB application is assumed to be is located in a server. First, from a FB application point of view, it may be an advantage to model the client also as a FB application. This is the view of the IEC 61499 series. This would require the client being modelled using FBs. Most PCs (at this time at least) are programmed using non-FB languages. Second, from a client's point of view, control, maintenance and monitoring are required to be independent of the FB model/language. This is because of the generic nature of these applications. Standardized control, maintenance, and monitoring applications (the protocol required to run these) provide functions to be used in many other control environments, i.e., device applications programmed in C, C++, or any other non-FB language. IEC standards like IEC 60870-6-503 and IEC 60870-6-702 (TASE.2) provide services and protocols for monitoring, control, and any information modelling and access. On the other hand, the configuration application is by definition specific for a FB model/language.

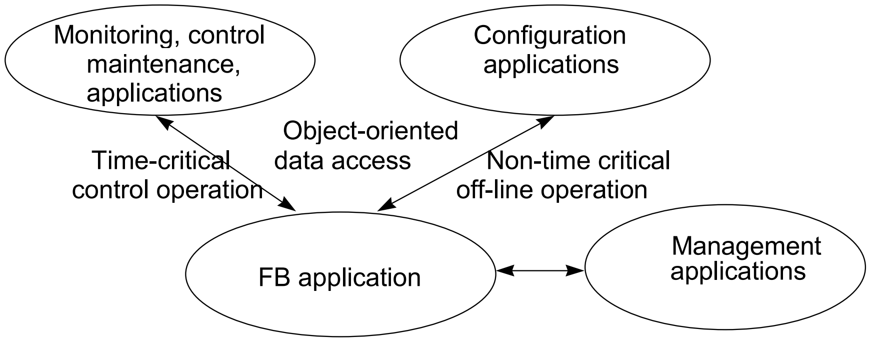


Figure 8 – Data access of non-FB applications

8.1.1 Cooperation with control, maintenance and monitoring applications

8.1.1.1 General

NOTE There are some requirements overlapping with those arising from FB applications. However, the specification has to consider both points of view [4].

Cooperation with control, maintenance, monitoring applications requires a common set of application functions (data access, data reporting, data logging, control functions, etc.) that are found in most real-time automation devices especially in FB-based applications. The use of a standardized set of application services allows for

a) isolation of the modelling efforts from the communication details,
b) a high degree of *application* interoperability, not just in the message syntax but also in the semantics of the data exchanged,
c) reduced integration costs, in that there is a consistent access and representation mechanism across all of the real time data.

8.1.1.2 Distribution of end systems

Cooperation with control, maintenance and monitoring applications is required to support

a) end devices and processors, situated anywhere within the plant territory, including in corporate facilities, in control centres, within cells, and at customer premises,
b) communications with systems external to the process control system, including over public communications systems and the public Internet,
c) an overall system environment, which may include high-speed communication channels, low-speed channels, shared channels, media with unique transmission characteristics such as radio, mobile radio, satellite, power line carrier, etc.,
d) the current and future network configurations, which may include hierarchical sub-networks, peer-to-peer communications within a subnetwork, and full peer-to-peer communications across the enterprise.

8.1.1.3 Device/processor capabilities

Cooperation with control, maintenance and monitoring applications is required to

a) take into account that the processing, computational, and storage capabilities of specific devices may range from very powerful to severely compute and memory constrained devices,
b) be able to support from one to hundreds of devices on the same network or subnetwork.

8.1.1.4 Dialogue characteristics

Cooperation with control, maintenance and monitoring applications is required to

a) support different communications dialogue and data flow requirements, which will include
    1) one-way only,
    2) two-way alternating (such as a hierarchical polled architecture),
    3) two-way managed by the network protocol or media (such as fieldbus, token passing, CSMA/CD, and radio-based "intelligent" media, for handling unsolicited reporting),
    4) two-way simultaneous (such as a full duplex WAN);
b) support messaging across a variety of intermediate devices, local subnetworks and WAN links;
c) support multiple concurrent communication subclauses between the same devices;
d) support prioritization of messages and/or the ability to interrupt long messages;
e) support application level security;
f) maximize network throughput by supporting efficient encoding or data compression techniques.

8.1.1.5 Addressing of devices

Cooperation with control, maintenance and monitoring applications is required to support

a) addressing of devices which range in number from a few to hundreds on the same network,
b) addressing schemes for handling short addresses valid only within local subnetworks, while permitting access to these addresses across wide networks,
c) addressing schemes for handling broadcast and multicast requirements.

8.1.1.6 Network traffic characteristics

Cooperation with control, maintenance and monitoring applications is required to support

a) data rates from very low speeds (10 s of bits per second or even less) to ultra high speeds (gigabits per second) for transferring information and potentially images and graphics,
b) differing frequencies of transmitting data ranging from low-frequency transmissions (once or twice a year) to high-frequency transmissions (ms) to ultra high frequency (in special cases on the order of every 50 µs),
c) transmission of messages ranging from a few bytes of data to very large files.

8.1.1.7 Timing issues

Cooperation with control, maintenance and monitoring applications is required to

a) support clock synchronization of devices,
b) support time stamping with at least 1 ms of accuracy,
c) minimize time measurement skew, with at least 1 ms of accuracy.

8.1.1.8 Application service requirements

Cooperation with control, maintenance and monitoring applications is required to support

a) concurrent access of a system by multiple users;
b) scheduled and unscheduled data exchange;

c) remote data retrieval on demand by the client;
d) unsolicited reporting of data, driven by events at the server;
e) publisher-subscriber data exchange;
f) report by exception transmission of data, in which only changed data is transmitted;
g) remote device control commands (constraints to be supported before control is executed: immediate control, control on specific conditions like setpoint value between limits or setpoint value limited to a specific configured value, etc.);
h) remote device control commands are required to be allowed by multiple clients, clients may be allowed to connect to a server any time;
i) prevention of concurrent control commands by restriction to one operator at a time or by a semaphore mechanism;
j) remote event recording and logging of events;
k) broadcast and multicast data to devices;
l) remote program control;
m) download and upload of device configuration files;
n) error management and reporting.

8.1.1.9 Application data format

Cooperation with control, maintenance, monitoring applications is required to support

a) object-oriented data structures and functions,
b) data types common to FB applications,
c) self-defining naming conventions for data.

8.1.1.10 Operational requirements

Cooperation with control, maintenance and monitoring applications is required to support

a) network management requirements, including
   1) recording network protocol statistics, such as number of temporary and permanent link failures, data retransmissions, and link utilization;
   2) monitoring, detecting, and notifying of critical network failures to all appropriate users on the network;
   3) dynamic bandwidth allocation;
   4) online re-configuration of remote network devices;
   5) charge-back services to reallocate costs to network users or departments;
   6) accounting management facilities for use of third party network facilities;
   7) management of security services and mechanisms to ensure maintenance of overall network system security.
b) directory services, including
   1) identification of applications, processors, devices, and data objects;
   2) integration with outside directory services, such as public e-mail providers;
   3) restriction of information only to authorized users;
   4) search and sort capabilities.
c) security mechanisms, including mechanisms to ensure:
   1) confidentiality, by preventing the disclosure of transmitted data to unauthorized parties;

2) integrity, by detecting the modification, insertion, deletion, or replay of transmitted data;
3) data-origin authentication, by demonstrating that the origin of transmitted data is as claimed;
4) non-repudiation, by preventing either the sender or the receiver in a communication from denying their participation;
5) user authentication, by demonstrating that the identity of a user or system is as claimed;
6) access control, by guarding against unauthorized access to resources including the attempted use of resources in an improper manner.

8.1.2 Cooperation with configuration applications

In this context, configuration (of a system or device) is required to be understood as a step in the system or device design with the following activities:

a) selecting functional units;
b) assigning their locations;
c) defining their interconnections.

Cooperation with configuration applications is required to perform the cooperation with the following devices:

a) remote creation of data types, FB types and instances as well as data connections among FB instances;
b) remote deletion of data types, FB types and instances as well as data connections among FB instances;
c) remote starting, stopping, killing applications and FBs;
d) remote query of all data types, FB types, FB instances, data connections, variables.

NOTE 1 Remote means, in this context that the cooperation requires an application protocol (as opposed to an Application Layer of, for example, a fieldbus). This protocol defines all the information to be exchanged, its meaning as well as the order and sequence of the information exchanged. In addition, it requires a precise and complete set of definitions of the procedures initiated by reception of the exchanged information. The protocol also defines any exception that may occur during the cooperation.

NOTE 2 There is an overlapping of online configuration and online control. As an example the setting of a setpoint may be realized using online configuration or online control.

8.2 Additional characteristic requirements

8.2.1 General

The process control user requirements in this subclause address the following considerations:

a) FB environment;
b) communications functions;
c) revision management.

Many items listed in this subclause have the appearance of technical implementation details rather than user requirements. However, because of the multivendor nature of intended systems, users require a standardized implementation of these technical details in order to produce operable systems. Furthermore, users require specific implementations of certain technical details, because the distributed nature of the application process requires a specific implementation in order to produce operable systems. Finally, underlying the requirements of specific implementation details are decades of experience with different implementations by different system suppliers, and a broad understanding of what will perform best in a majority of process control applications.

8.2.2 Multivendor systems

A process control user application, called a FB environment, is required to be suitable for the construction and operation of multivendor systems. The application model is required to enable users to easily determine the degree of compatibility between devices. Device compatibility is required to be described in a way that makes it clear that the device is interconnectable, interworkable, interoperable or interchangeable.

The application model is required to define standard data structures, behaviour, FBs, and profiles for commonly used process control functions and devices, as an important step forward to enable interchangeable devices. The model is also required to include provisions for combinations of standard and custom data structures, behaviour, FBs, and profiles to enable interconnectable, interworkable and interoperable devices.

8.2.3 Extensible

The process control application model is required to be extensible for new standard and manufacturer-specific data structures, behaviour, FBs, and device profiles. Extensions that conform to the process control application model are also required to meet the user requirements for multivendor systems. The process control application model is required to include a device descriptive language to support the required extensibility.

8.2.4 Distributed applications

8.2.4.1 General

The application model is required to provide for process control applications that are distributed in different devices.

8.2.4.2 Communications for distributed control

Distributed process control applications have unique communications requirements that are required to be met by the embedded communications protocols. These include secure communications relationships between distributed applications, functionality for synchronization of application events and functions, and provisions for time-critical communications, time-stamping of data and events, standard processing of alarms and events, execution of distributed atomic operations, and segmenting and re-assembly of large data sets.

8.2.4.3 Optimized for fieldbus

The process control application model is required to function with field communication standards, for example, the IEC 61158 series, EN 50170, etc.). The process control application model is required to be optimized to maximize the utilization of available bandwidth in low speed, intrinsically safe, fieldbus-based control systems. The data structures and communications functions defined in the application model are required to map efficiently to the fieldbus communications services and protocols.

8.2.4.4 Compatible with ISO/OSI Model

The process control application model is required to function with various open systems communications networks, and therefore is required to be compatible with the ISO/OSI model for data communications. The process control application model will provide common application usage semantics in the context of the OSI model. The process control application model does have unique communications services requirements. These include – but are not limited to – time-critical communications, system management services, and protocol speed and efficiency. The unique communications services requirements of the process control application model are required to be specified, to enable users to easily determine the degree of compatibility of a particular OSI-compatible protocol with a conformant process control application.

8.3 Conformance requirements

8.3.1 Organizational support

The process control application model is required to be supported by an active organization of technology developers, equipment suppliers, and end-users, responsible for implementing and maintaining the model in real systems.

8.3.2 Ongoing changes and additions

No process control application can be complete and unchanging, and still remain useful to meet the changing needs of real applications. The standardized process control application model is required to have the support of an efficient and effective sponsoring organization, capable of implementing periodic and timely additions, deletions and changes, to accommodate the changing needs of the process industries.

8.3.3 Conformance testing and certification

The efficient and effective implementation of real systems in real applications requires the unambiguous identification of devices and systems that conform to the standard process control application model. The model is required to include objective and measurable criteria needed by independent test organizations to certify compatible devices and systems.

8.3.4 Training and support

Developers, suppliers and end-users alike need ongoing sources of interpretation and training to support implementation of real systems that conform to the standard process control application model. The model is required to have the support of an efficient and effective sponsoring organization, capable of providing such ongoing training.

9 Device descriptive language

9.1 Background

The rapid development in the last years concerning the factory and process automation has made one basic concept inevitable: engineers and technicians have to be supported by computer-based tools. This becomes even more critical for distributed process control systems. Fieldbuses carry the data between the controllers and the sensors/actuators and between the engineering station and all devices. In all the different phases of the engineering process one or several tools running on a PC, a workstation or a terminal need to know what kind of devices they are connected to. This covers the vendor information, the version of hard- and firmware release, the data format to be exchanged, the reaction time, the physical unit, etc. In most cases, the fieldbus controller, located in the engineering station also needs to know the communication abilities concerning responding time, supported baud rates, etc. With the development of remotely accessible devices, most of the device features are put together in a device description, which is delivered in addition to the appliance. The term electronic device description is used, because it is more then a language; indeed, a technology which shows how to integrate devices in the allover DCS life cycle.

9.2 Basic requirements

In order to work out, the general user requirements, the meaning of the term *user* has to be defined first. Two kinds of users can be distinguished:

– the end-user or operator of a plant or machine;

– the system integrator.

Do their requirements for an electronic device description differ although they are working with the same distributed system? The following paragraph will analyse this question.

For the end-user of a distributed control system, the most important thing concerning device description is its transparency. The end-user wishes to see only the graphical user interface of a device represented in the SCADA-software or other human machine interfaces (HMI). Therefore, the electronic device description has to be constructed following the plug and play concept. Furthermore, exchanging a device in an application is required not to lead to a big engineering task but is required to be done simply and securely by a technician. The device description has to support this.

In contrast to the end-user, the system integrator, who does the engineering, installation and documenting has different targets to meet. His goal is required to be to reduce the engineering time for solving interoperability problems but to enhance the engineering effort to design the optimal application with regard to the quality of the product to be produced with the process control system. The effectiveness of his work, comprising all pre-producing phases of the plant/machine life cycle, depends directly on the support of a well-defined, standardized and complete machine-readable description of the devices he has to deal with.

The conclusions from this analysis are, that the user requirements have to be regarded mainly from the system integrator's point of view, because he has to deal directly with the electronic device description. Because computer tools, known as Process Control Systems, are the state-of-the-art instruments the system integrator uses, the requirements for an electronic device description will be developed from what these tools need for the best support of the engineering process.

Process control systems provide control and supervision of production processes. They connect people (for example, the operator) and machines. They consist of input/output devices, data processing units, human machine interfaces and communication systems.

Traditionally, PCS are seen from a run-time point of view (function, device, and system). The engineering aspect becomes more important due to its increasing complexity and the costs involved. The complexity results from several factors influencing the engineering process such as different device components (input/output, data processing, HMI, communication), process physics (mechanical, electrical, etc.), life-cycle phases and device vendors. Today, the integration of these different worlds is usually done by building hardware and software interfaces and carrying out expensive commissioning processes based on trial-and-error approaches. This will become impossible under a true cost of ownership principle. Integrated tools are required to support the whole engineering process in order to avoid data losses and inconsistencies. The use of paper documents and even of electronic means does not provide a sufficient solution as long as there is no common transfer syntax and a standardized data model.

The fieldbus community developed some approaches to exchange data by electronic means. This includes Device Descriptive Languages (DDL) and Device Data Base (GSD) (HART, FF, and PROFIBUS).

The requirements for the Electronic Device Description (EDD), including the relevant language, can be summarized as follows.

a) An EDD (in terms of a file) is required to be delivered by the device vendor together with the device.
b) EDD is used in the engineering process of the distributed control system, supporting planning, commissioning, operation, diagnostics and maintenance.
c) EDD is used within the context of a determined architectural model and supported by a corresponding chain of software tools (editor, tokenizer, and interpreter) with standardized interfaces.
d) Two different EDD presentations are required:
    1) source (human readable and computer interpretable),
    2) optional: binary (created by tokenizer, interpreted by interpreter).

e) The EDD is required to be stored on disc and is required to be stored within the device (transport via fieldbus).

f) The EDD is required to be independent from the underlying fieldbus system.

g) The EDD is used to describe information identifying each item and defining relationships between them (hierarchical, relational).

h) The EDD is required to describe at least the FB model.

i) The EDD is required to offer language elements for presentation within a HMI and for communication access.

j) The EDD is required to include object-oriented features.

k) The EDD is required to offer language elements for timing considerations.

l) Additionally, the following assumptions are made:

   1) the integration of the EDD into the device development itself will not be considered;

   2) an EDD-file is required to represent a static description, i.e. the declarative part of the device, therefore only the external interface/behaviour of the device is to be described (no interest in internal code).

9.3 General requirements

If there is a complete FB model of a field device, it is only consequent to derive the requirements for the describing language from it. The following points are the key concepts concerning language properties.

a) In order to reach a uniform, non-redundant specification a hierarchical structure of the EDD reference model is required. *Generalization/Specialization technique (Gen/Spec):* this concept is used above all in object-oriented analysis and guarantees a clear survey of a certain application field. For the expansion of a chosen hierarchy structure in the future, this concept is fundamentally important.

b) In order to guarantee interoperability with future IFD (Intelligent Field Device)-EDD specifications and a version control, a conformity class is required to be defined for each specification.

   Container class principle: as above, this concept is naturally supported by the modern object-oriented approaches.

c) To reach a certain level of standardization and give the vendor or user additional room to introduce proprietary features (replace also the following services) at the same time, there are three different classified subsets of services:

   1) basic or mandatory services: required to be implemented;

   2) optional services: fully specified, but not assumed to be implemented;

   3) vendor services: not specified, but implemented by choice of the vendor.

d) *Interface technique:* likewise developed in the object-oriented world, the interface concept allows the logical representation of a group of related methods or functions.

   NOTE For example, the *CORBA* or *OLE for Process Control* (OPC) specifications uses this technology.

e) The PRIAM model defines at a high level of abstraction the required (actuation or measurement) channel to be implemented by using field devices completed, if necessary, with additional software. According to the PRIAM recommendations, a lower level of abstraction could be used to help the mapping on the FB model: at this level the device data and functions could be grouped in so-called device functional units that should represent typical hardware or software components. This would require describing the field device in a functional decomposition model where also the data flows are visible. *Functional decomposing technique:* this recommendation may lead to an object modelling method to describe applications in general. The model, once designed, would require supporting the complete life cycle of a distributed process control system. This overall model is outside the scope of this Technical Specification.

The overall concept is required to use some concepts of the object-oriented technologies. *Database concepts* are required to be supported. Standardized description tools to avoid the expense of developing a totally new computer language, existing formal standards and tools have to be examined.

Annex A
(informative)

Life cycle of the system

A.1 FB derivation and implementation

A.1.1 The steps to the FRD

A.1.1.1 Heading

A.1.1.1.1 Control function

There are different types of control functions:

a) measurement control functions delivering direct or indirect process measurements;

b) control functions:
   - open loop control functions controlling on/off actuators,
   - closed loop control functions, single or cascade, controlling modulating actuators,
   - sequence control functions, coordinating lower hierarchical control functions.

A.1.1.1.2 Control hierarchy diagram (CHD)

Graphical schematic and structured representation of the hierarchy of the control functions of a process control application. The lowest layer is composed of the control functions interacting with the transmitters and actuators. The upper layers are composed of control functions coordinating lower layer control functions.

From the CHD, process engineers define the structure of the documentation of the FRDs by allocating control functions into the folios.

A.1.1.1.3 Folio

FRDs are structured as a set of folios. A folio is a detailed description of the requirements of the control function(s) allocated to the folio. A graphical FB language is used to describe the behaviour of the control function(s) as a network of EFBs and application FBs.

A.1.1.1.4 FB (function block)

An EFB (EFB) is a simple logico-mathematical operation usable in different types of process control applications (chemical, petrochemical, energy, etc.). An EFB is independent of any supplier programming language. A library of standardized EFBs is currently under consideration .

An AB is a complex logico-mathematical application, which is defined and validated by, or for, a company. An AB depends on the type of process control application (chemical, petro-chemical, energy, etc.). An AB is independent of any supplier programming language.

NOTE It is necessary to specify the behaviour of an AB to fulfil interoperability. The specification should define how to link FB, grouping in library, identification when used in resources, version control, etc.

A.1.1.1.5 Functional Requirement Diagram (FRD)

Detailed requirements of the control functions described, by process engineers, with a neutral FB language composed of standardized EFBs and ABs built with standardized construction rules.

A.1.1.1.6 Process Flow Diagram (PFD)

This is a schematic and graphical representation of a part of a process. On the PFD are represented the main mechanical units, the piping between these units, the remote transmitters and actuators. Process engineers define a partition of the Process Circuit Diagram into process elementary operation.

A.1.1.1.7 Extended Piping and Instrumentation Diagram (P&ID)

On the conventional P&ID, only the control loops are represented; on the extended P&ID all the control functions are represented.

A.1.1.1.8 Process elementary operation

A process elementary operation is a part of a PFD representing a typical transformation of material and energy. The PFD is split into process elementary operation to highlight the main transformations of material and energy to be controlled.

A.1.1.2 From the process down to the FRD

A.1.1.2.1 Overview

The aim of this subclause is to illustrate what are FBs used for. The requirements for FBs are illustrated as key elements for engineering companies to specify control.

The FRDs are the specifications of the control functions to control a process. The FRDs can be obtained through different studies, depending on the industrial area.

These studies should be summarized in the following life cycle. As FRDs totally depend on the process to control, the Process Elementary Operations which have to be controlled have at first to identify on the PFD. On the P&ID, the control functions of the Process Elementary Operations and of the process have to be defined (network of elementary process operations to control). In the CHD the documentation of the control functions are structures as a set of folios. On each folio, the control functions are specified in detail as networks of FBs to achieve the FRDs. This life cycle is summarized in Figure A.1.

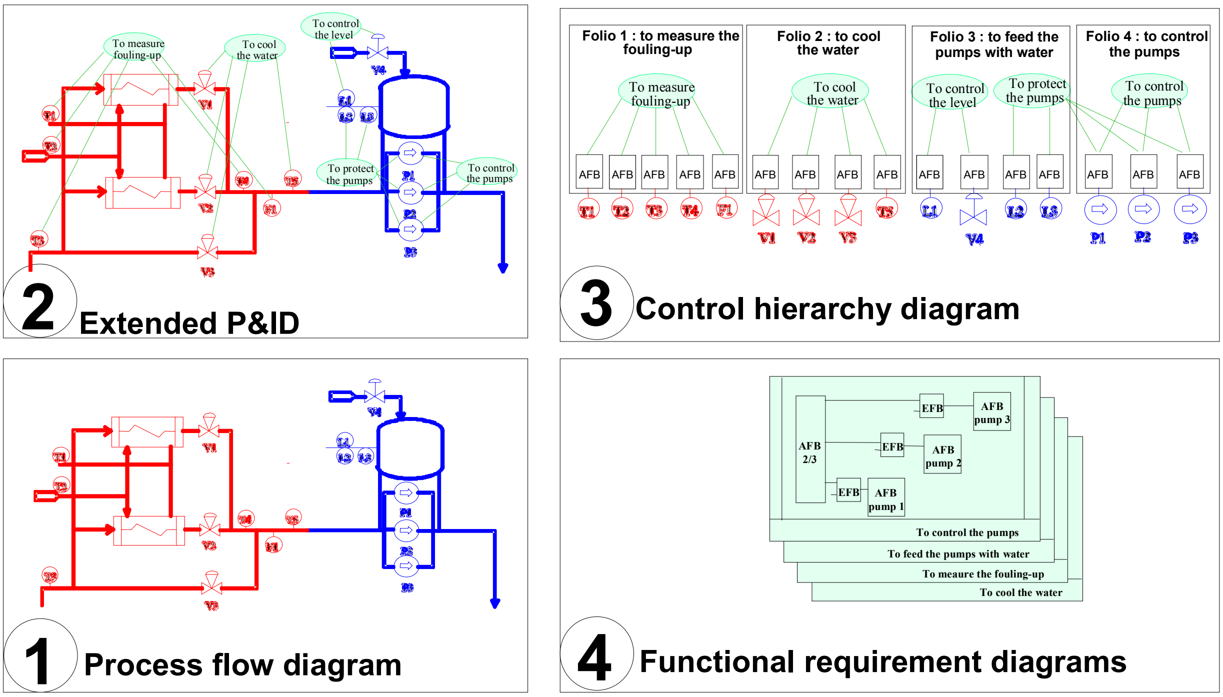


Figure A.1 – Life cycle from the process circuit to the FRDs

A.1.1.2.2 Use of PFD

The PFD is a schematic representation of the process to be controlled. On this diagram only process information necessary to understand the operation of the process appears. On the PFD any process engineer will see the process as a network of equipment supporting the major process elementary operations.

On the PFD, the graphic symbols should be in accordance with the mechanical symbol standards. Unfortunately, there is not a unique standard. In the fields of chemical and petrochemical industries the symbols are mainly standards from the ISA. There are also IEC graphic symbol standards as well as some major company standards. These standards are not homogeneous.

A.1.1.2.2.1 Identification of the process elementary operations to control

As control functions totally depend on the type of process, the elementary operations of the process to be controlled have to be identified. Process equipment units support an elementary operation. An elementary operation will transform material and energy, for purposes such as

– to cool water;
– to make water circulate;
– to feed pumps with water.

Figure A.2 shows an example of PFD composed of two process elementary operations, to cool water and to make water circulate. In this example, the process elementary operations are presented with different colours.

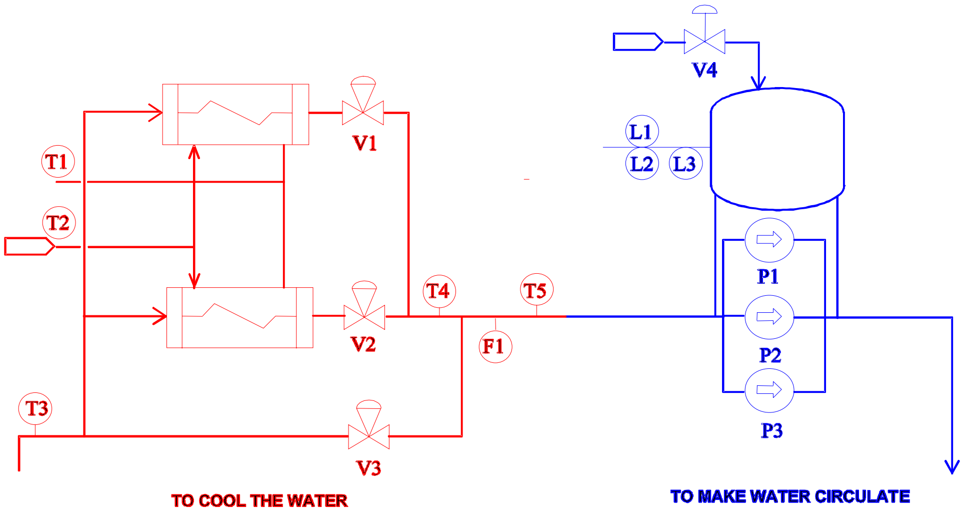


Figure A.2 – PFD composed of two process elementary operations

From the PFD, any process engineer can see this process as a connection of two elementary process operations. The second elementary process operation could have been split into two suboperations, for example, to load the pumps and to pump water. The definition of the elementary process operation depends on company know-how.

From this PFD, the process engineer will define control functions to control the two process elementary operations and the process composed of these two process elementary operations.

A.1.1.2.3 Extended P&ID

To illustrate the relationships between control functions and the process, an extension of the so-called P&ID commonly used in chemical and petrochemical industries is used. On the conventional P&ID, remote transmitters, remote actuators and controllers are mainly represented.

In addition to remote transmitters and actuators, all the control functions on an extended P&ID is represented. Process engineers distinguish different types of control functions:

- measuring control functions: elaboration of a measurement from one or several transmitters;
- open-loop control functions: controlling on/off actuators with respect to one or several measurements and/or with respect to an order from an operator or a sequence;
- closed-loop control functions: controlling modulated actuators with respect to one or several measurements and a set-point;
- sequence control functions: controlling on/off actuators, sequencing open- and closed-loop control functions and other sequences.

In addition to controllers, which are closed-loop control functions, the other types of control functions, such as measuring control functions, open-loop control functions, sequence control functions are also represented in the process control diagram.

On the extended P&ID, a control function should be represented as a graph between symbols representing inputs, the control processing and outputs. The control processing is a separate graphic symbol within the tag of the control function inside. The inputs of a control function should be measurements and/or data from other control functions; they will be represented as lines between the control processing symbol and the symbols of the remote transmitters and/or of the other control function symbols. The outputs of a control function should be orders to actuators under control and/or data to other control functions, they will be represented as links between the control processing symbol and the symbols of the remote actuators and/or of the other control function symbols.

Figure A.3 is an example of an extended P&ID for the process described in Figure A.2.

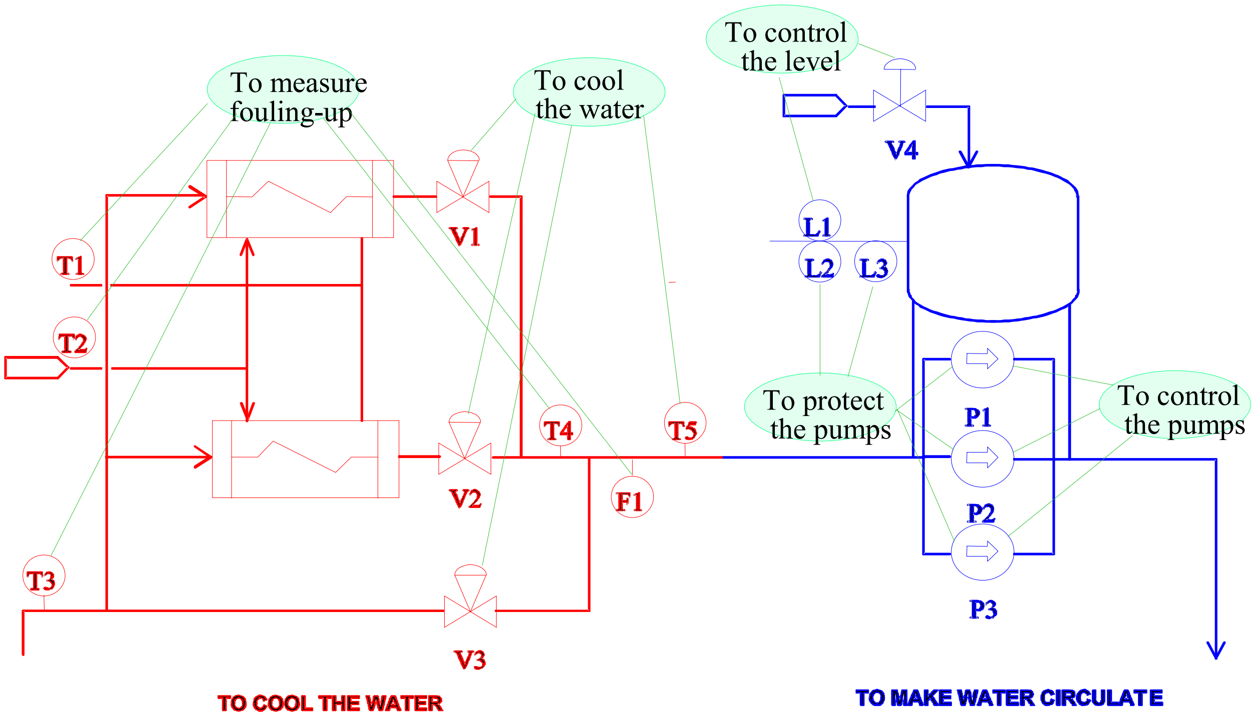


Figure A.3 – Control functions explicitly represented on the extended P&ID

On the extended P&ID, for each control function, the causal relationships between the observations (from transmitters and other control functions) and the commands (to actuators and other control functions) are explicitly visible.

To control the first process elementary operation "to cool the water", process engineers define two control functions. The first control function is a closed loop control function "to cool water", controlling the temperature of the water from the remote transmitter T5 and actuating the modulating valves V1, V2 and V3. The second control function is a measuring function "to measure the fooling up", calculating the fooling up of the exchangers from the remote temperature transmitters T1, T2, T3, T4 and the remote flow transmitter F1. This measuring function controls the operation of the current exchangers in the nominal range; in case of trouble, a signal is sent to the operators.

To control the second elementary operation "to make water circulate", process engineers define three control functions. The first control function is a closed-loop control function "to control the level" of the tank, controlling the level of the tank from the remote level threshold L1 and maintaining the level of the tank by controlling the valve V4. The second control function is an open-loop control function "to protect the pumps" controlling the level of the tank from the two remote level thresholds L2 and L3 and switching off the pumps P1, P2 and P3 when the level of the tank is too low. The third control function is an open-loop control function "to control the pumps" allowing the operation of two pumps among the P1, P2 and P3 pumps to make sure the water is circulating.

The extended P&ID is a diagram usable both by process engineers and I&C engineers and facilitating the communication between these two trades. Once the control functions have been identified and tagged, the framework of the documentation describing the control functions should be organized.

A.1.1.2.4 Control Hierarchy Diagram (CHD)

A.1.1.2.4.1 Overview

The aim of the CHD is to structure the documentation of the control functions. The CHD is first an extraction of the control functions from the extended P&ID, second the definition of the requirements of the control functions and in particular of the ABs, third the structuration of the documentation of the FRDs.

A.1.1.2.4.2 Extraction of the CHD from the extended P&ID

The CHD is dedicated to control, so data about the process is no longer needed. If such data is needed the PFD or the extended P&ID should be used. The CHD is obtained first by the extraction of the control functions from the extended P&ID. In Figure A.4, there is an extraction of the control functions from the extended P&ID of Figure A.3

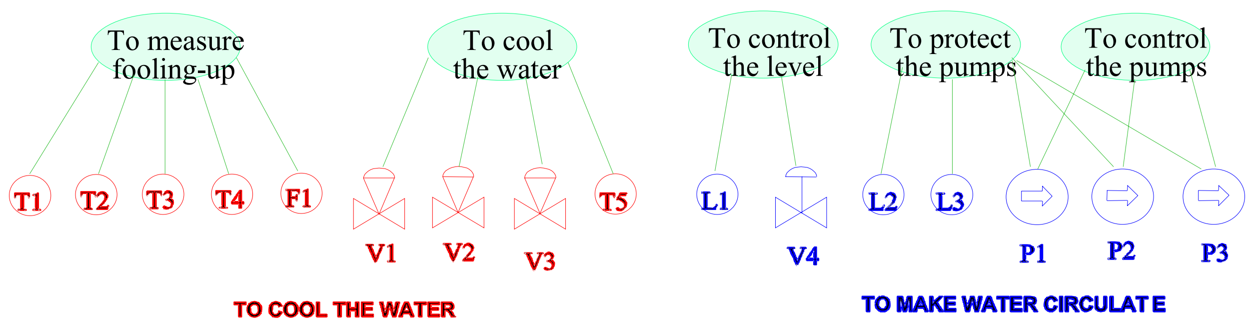


Figure A.4 – Extraction of control functions from the extended P&ID

In the example shown in Figure A.4, the hierarchy of the control functions is flat, and there is only one level of control functions. This is due to the process, simple to control, and to the low level of automation of this control application. For a complex process or for a high level of automation, the control functions above the first level of control functions can be seen.

A.1.1.2.4.3 Requirements of the control functions and of the standardized ABs

The CHD is a hierarchical representation of the control functions, explicitly describing the relationships between the control functions and the process interface, the transmitters and the actuators. On this CHD, process engineers will specify the requirements of the control functions and of the ABs.

The ABs are standardized blocks of a private application library. The aim of the re-use of standardized blocks is to reduce the cost and to improve the quality of the control application. For each control function, process engineers select and specify the requirements of the ABs suited to the control functions and the process to control.

For example, for the control function "to measure the fouling-up", process engineers select from a library of ABs those which are suited to the different types of transmitters T1, T2, T3, T4 and F1 and set the requirements of each AB with respect to the type of transmitter and to the process constraints. The requirements of the ABs are filled in for the other control function "to cool water", "to control the level", "to protect the pumps" and "to control the pumps".

Figure A.5 improves upon Figure A.4. On the CHD, in progress, the selection of the Application FBs can be seen for each control function.

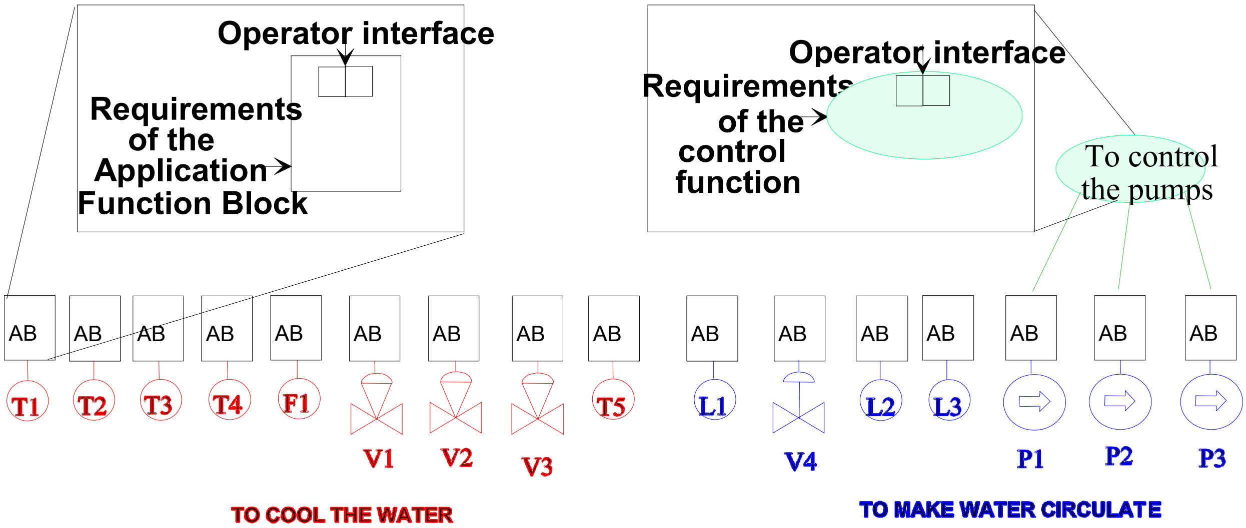


Figure A.5 – Requirements for control functions and ABs

NOTE The dotted line shows that requirements are the basis of the ABs and the operator interface is derived direct from the ABs.

Process engineers select and fill in the requirements of different types of application FBs such as measurement, actuation, switch over etc. The requirements of application FBs are constraints, for example, accuracy, sampling time, number of operator interface from which state and status, response time between operator command and actuator operation, interface availability, safety and others can be read.

The requirements for control functions will be also completed, for example, sampling time for control loops, response time between alarm detection and protection operation. The requirements for both the ABs and the control functions are functional: at this stage, it is not known in which I&C system the control functions will be implemented but there is enough information to evaluate the size of any I&C and to prepare the call for tender.

A.1.1.2.4.4 Structuration of the documentation of the FRDs

The aim of Figure A.6 is to show the organization of the documentation of the FRDs. The FRDs are a set of folios on which the control functions will be specified in details with FBs. Structuring the documentation of the FRDs means distributing the control functions into folios.

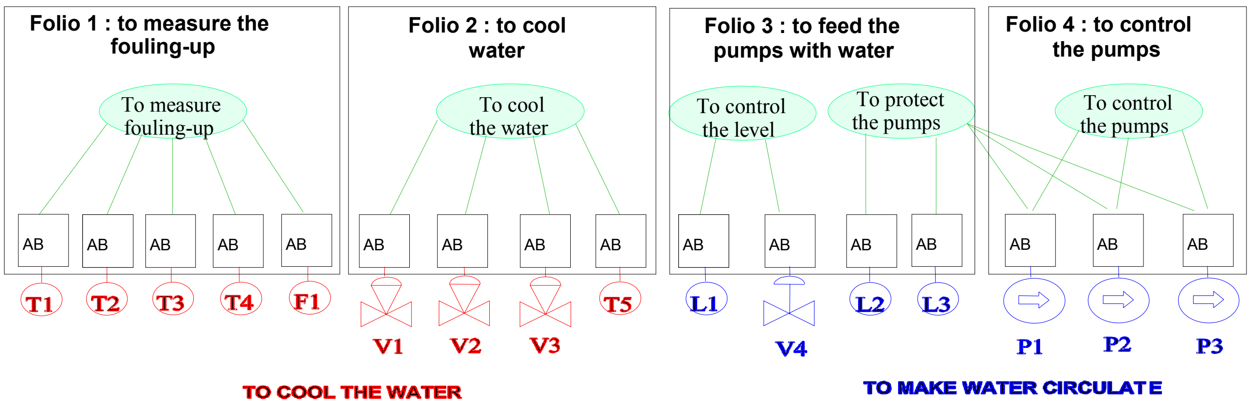


Figure A.6 – Structured documentation for the requirements of the control functions

One folio can support one or several control functions. Process engineers make this choice. Figure A.6 is an example in which process engineers assign the control function "to measure the fooling up" to folio 1, "to cool the water" to folio 2, "to control the level" and "to protect the pumps" to folio 3, "to control the pumps" to folio 4. Once the requirements of the control functions and of the ABs is completed and the control functions are distributed into the folios, clear and explicit documentation of the requirements of the future control system is available.

A.1.1.2.5 Use of FRD

A.1.1.2.5.1 The folio: the key of the structure of the documentation of the FRD

The FRDs are the detailed requirements of the control functions. These detailed requirements are described as a network of EFBs and ABs in the folios. These folios have been selected in the CHD.

For a complete description of a folio in the FRDs, it is necessary to fill in the four sections of a folio. On the left, the input signals are described; on the right, the signal outputs. In the centre of the folio the signal processing is described in terms of a network of FBs. At the bottom, the title of the folio is described. The four sections of a folio are summarized in Figure A.7.

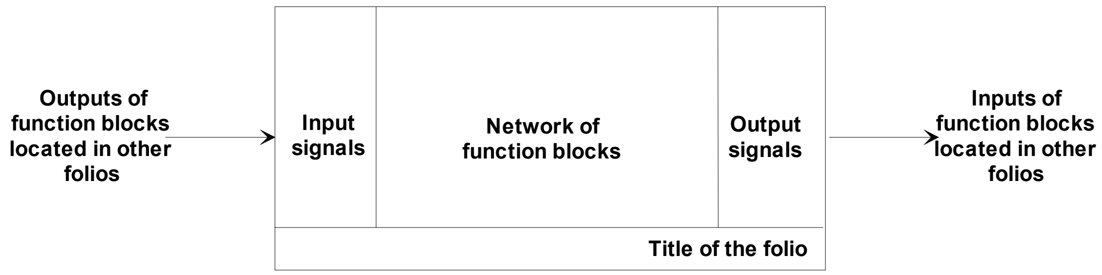


Figure A.7 – The four sections of a folio

The input signals of a folio come from outputs of other folios. The input signals of a folio are used as inputs for some FBs located in the folio.

The output signals of a folio are data used as inputs of other folios. The output signals of a folio are outputs from some FBs located in the folio.

A.1.1.2.5.2 Going into the folios for detailed requirements of control function(s)

In the CHD, process engineers distributed the control functions into the folios (one folio can support one or several control functions) and selected the different types of ABs for example for measurements and actuations. To detail the requirements of the control function(s), process engineers go into the folios.

Going into a folio, process engineers have an empty folio except the ABs previously selected in the CHD. To detail the requirements of the control function(s) located in a folio means to build a network of EFBs and ABs. To network the ABs with EFBs process engineers will select the inputs and the outputs of the ABs. Figure A.8 summarizes going into a folio and selecting the inputs and outputs of ABs.

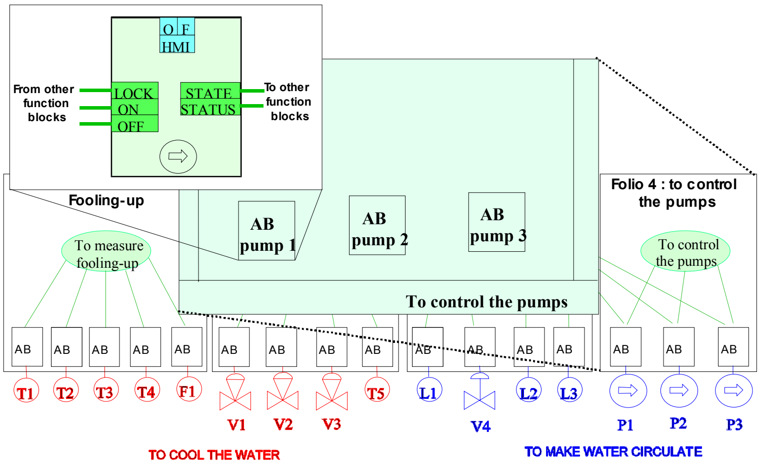


Figure A.8 – Selection of a folio from the CHD

A.1.1.2.5.3 Completion of a folio supporting control function(s)

To complete the detailed requirements of control function(s) allocated to a folio, process engineers fill in the four sections of the folio.

The input signals of the folio have to be consistent with the signals coming from other folios. The output signals of the folio have to be consistent with the signals used in other folios. For example, with reference to Figure 6, there is a link between the control function "to protect the pump" and the control function "to control the pumps"; this is a protection signal to switch off the pumps when the level of the tank is too low. As the control function "to protect the pump" is detailed in folio 3 and the control function "to control the pumps" is detailed in folio 4, this link will be represented as an output signal on folio 3 and as an input signal on folio 4. On folio 4, this signal input will be connected to the three ABs of the pumps.

For the description of the behaviour of the control function(s), process engineers network EFBs and ABs with respect to the requirements and the aim of the control function(s).

Figure A.9 is an example of a folio completed for the control function "to control the pumps". In this figure, which is a real example, the behaviour of the control function is a detailed network of EFBs and ABs (three pump actuation ABs and one switchover 2/3 AB).

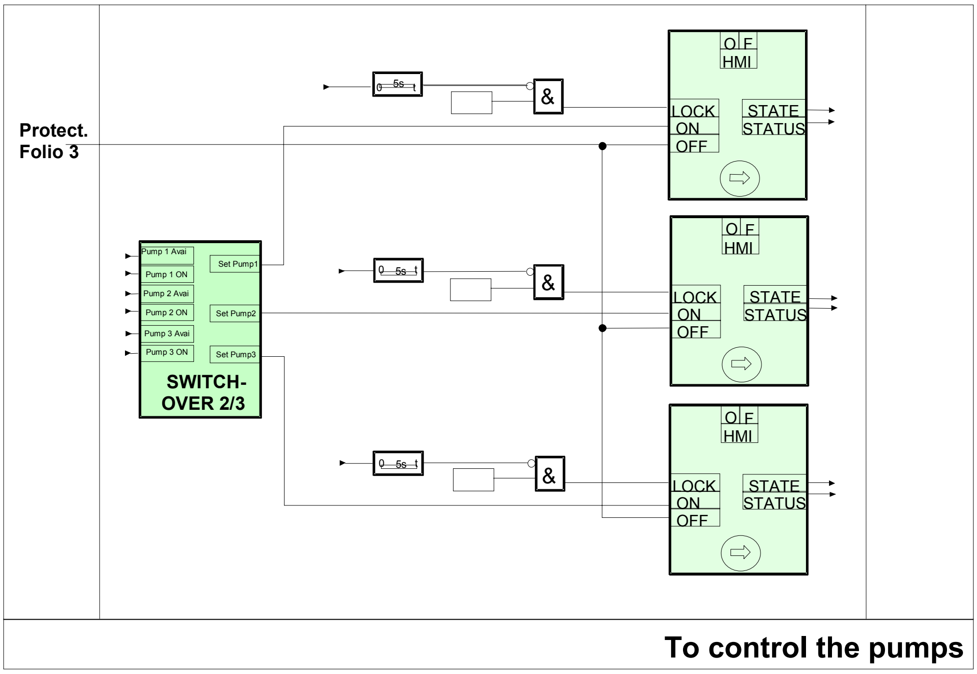


Figure A.9 – Folio for detailed requirements of the control function "to control the pumps"

A.1.1.2.5.4 FRDs as a set of folios

Process engineers will detail all the control functions distributed into the folios. For each folio they fill in the four sections of the folio. For the behaviour of each control function, they add details with a suited network of EFBs and ABs. Once the description of the folios is completed the FRDs are ready.

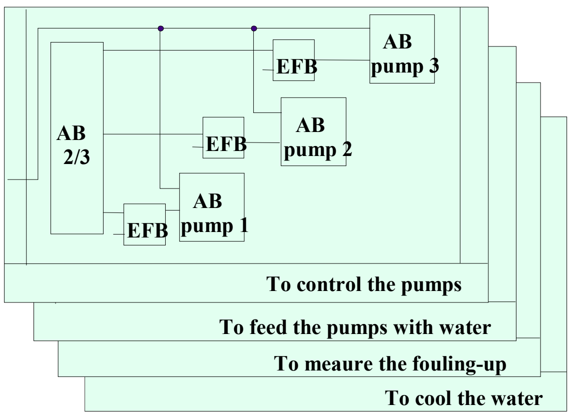


Figure A.10 – Example of FRDs

In Figure A.10, the four folios of the FRDs describing the detailed requirements of the control functions allocated to these folios on the previous CHD. The graphical representation of control function "to control the pumps" has been simplified.

Process engineers also define constraints attached to the EFBs and Abs, such as safety, availability and time constraints, for example, FBs processing ordering, constraints of implementation of the EFBs and Abs. Some EFBs and ABs have to be implemented together into the same I&C device; some other EFBs and ABs do not have to be implemented together into the same I&C device.

The FRDs should be considered as the specifications of the control functions to control the process. The FRDs are not programming schemes. FRDs describe the control functions (a networks of EFBs and ABs), the performances and the constraints (availability, safety) of the future I&C system in which the FRDs will be implemented.

Figure A.11 illustrates the FRDs independent of I&C system implementation

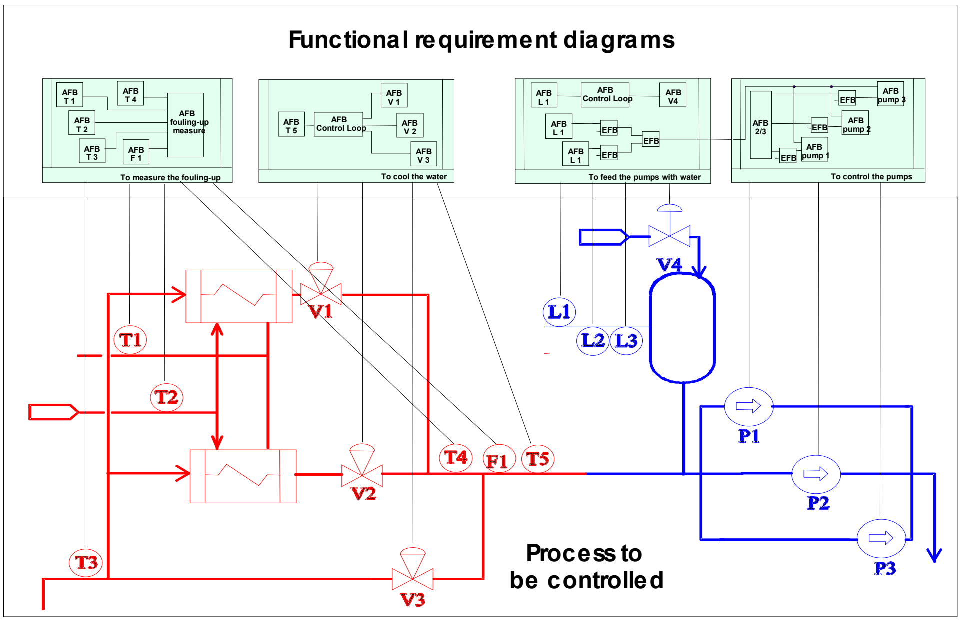


Figure A.11 – Functional requirements diagrams independent of I&C system implementation

A.1.1.2.6 Requirements for a standardized FB language

A.1.1.2.6.1 Overview

In the previous clauses, the FRDs of any control application have been established. Engineering companies need engineering FB language to describe FRDs, I&C suppliers need programming FB language to implement these FRDs into any type of I&C systems through.

The compatibility between engineering FB languages and programming FB languages is needed as summarized in Figure A-12. This means that a standardized FB language is needed as a kernel to define either engineering FB languages or programming FB languages.

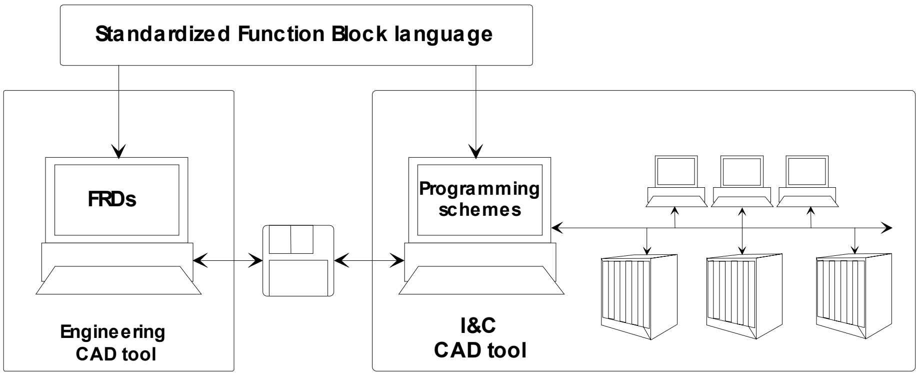


Figure A.12 – The need for a standardized FB language

The requirements of this standardized FB language are that

- it should a kernel to build engineering FB languages and programming FB languages;
- it should be built on standard foundations, in particular,
  - – it should be built on standardized EFBs,
  - – it should be built on standardized construction rules to define dedicated ABs (energy, manufacturing, chemical etc.) implementable into I&C systems,

This standardized FB language should be used by engineering companies to define their own engineering FB language allowing process engineers to describe the FRDs of control applications.

Describing the FRDs with an engineering FB language does not mean describing the FRD without taking into account the capabilities of the I&C devices available on the market. It means defining FRDs implementable into whatever the I&C system is compliant with the standard.

This means also FRDs will be implementable into current scalable I&C systems and upgradable I&C systems of the future compliant with the standard according to A.1.1.1.4 .

To describe FRDs with any engineering language implementable into any programming language, the IEC standardized FB language will provide

– a library of EFBs,

– rules to build AB.

A.1.1.2.6.2 Requirements for EFBs usable into FRDs

An EFB is a repetitive logico-mathematical treatment, which is embedded into a block. An EFB is a specific processing module restricted to the process control domain. An EFB cannot be split!

A library of EFBs should be standardized. This library of EFBs should satisfy the needs of the different types of industries (chemical, energy, food, etc.) (see Figure A.13). This should be a standarized kernel library of EFBs common to the different types of industries and libraries dedicated to the types of industries, for example, for chemicals, energy, food, etc.

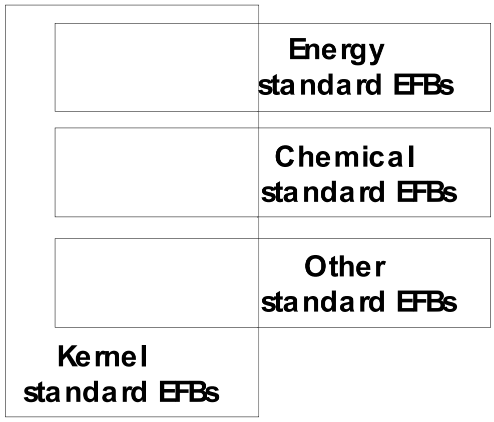


Figure A.13 – Library of standardized EFBs

The EFBs should be standardized, for each block both a standardized external graphical representation and a formal internal description of the behaviour of the block is needed.

The external graphical representation will be used within both FRDs and programming schemes.

The formal internal description of the behaviour should be carried out with a formal language proposed by IEC. The ST language proposed by the IEC 61131 series is not complete.

As EFBs will be implemented into digital I&C systems, the internal description will include the normal behaviour of the blocks and the abnormal behaviour of the blocks. The IEC 61131 series does not standardize the abnormal behaviour of the blocks. For process control and specially for safety constraints, the definition of the abnormal behaviour of each block in the case of general failures of the technology supporting and impacting the behaviour of some blocks (such as initialization, hot reset, cold reset) or in the case of specific failures impacting some blocks (such as, input signal failure from a transmitter, I&C input card failure impacting input blocks or overflow and underflow impacting computation blocks, for example) is necessary.

A.1.1.2.6.3 Requirements for ABs usable into FRDs

An AB is a reusable piece of control depending of the type of process control application.

As for EFBs, an external and internal representation of ABs is necessary.

Within FRDs the external graphical representation of ABs are used only because the FRDs should be implementation independent. The internal behaviour of the ABs, for example, for measurement and actuation ABs, totally depends on the type of technology supporting the measurement and the actuation, so the internal behaviour of the ABs will be described within the programming schemes with the programming FB language, once the technology supporting the ABs is selected.

A.1.2 From FRDs to implementation

The aim of this subclause is to complete the requirements of the FBs for control applications. Engineering companies need EFBs and ABs to specify FRDs with engineering FB language. These EFBs and ABs should be implementable through programming schemes into any I&C systems and devices compliant with this Technical Specification.

Once the FRDs are defined and validated, I&C engineers have to design the programming schemes from the FRDs and to implement the programming schemes into the I&C system and devices as summarized in Figure A.14.

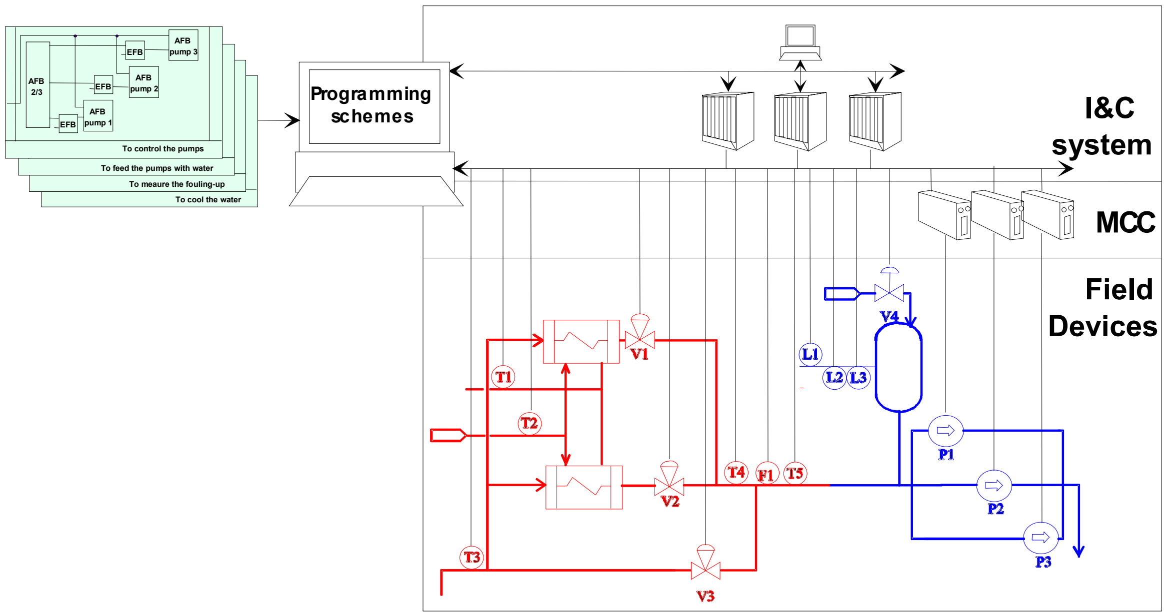


Figure A.14 – From FRDs down to the I&C system and devices

FRDs are I&C system independent; this means that FRDs are perennial specifications of any control applications. This implies FRDs should be implementable into any scalable I&C systems and devices for new applications. For existing I&C systems and devices, it also implies that for the maintenance or the renewing of existing I&C systems (partly or in total) the FRDs should be implementable into upgrades of new I&C systems and devices.

A.1.2.1 Design of ABs

A.1.2.1.1 Overview

To move from FRDs to programming schemes, the internal behaviour of ABs should be designed.

An AB should be composed of different internal blocks delivered by different suppliers, these internal blocks should be distributed and implemented into different I&C devices. An AB can be split into different internal blocks.

An AB can have interfaces with the process (with transmitters and actuators) and human machine interface. An AB can have no interface with the process or with HMI. Two examples are given in the following subclauses.

A.1.2.1.2 Detailed design of an actuation AB

Figure A.15 represents an actuation and the internal blocks of its actuation AB. This example focuses on the control point of view. An actuation should be typically composed of different devices delivered by different suppliers. For example, an intelligent actuator delivered by one supplier, an intelligent Motor Control Centre (MCC) delivered by a supplier and a processing block implemented into a device of an I&C system by a supplier.

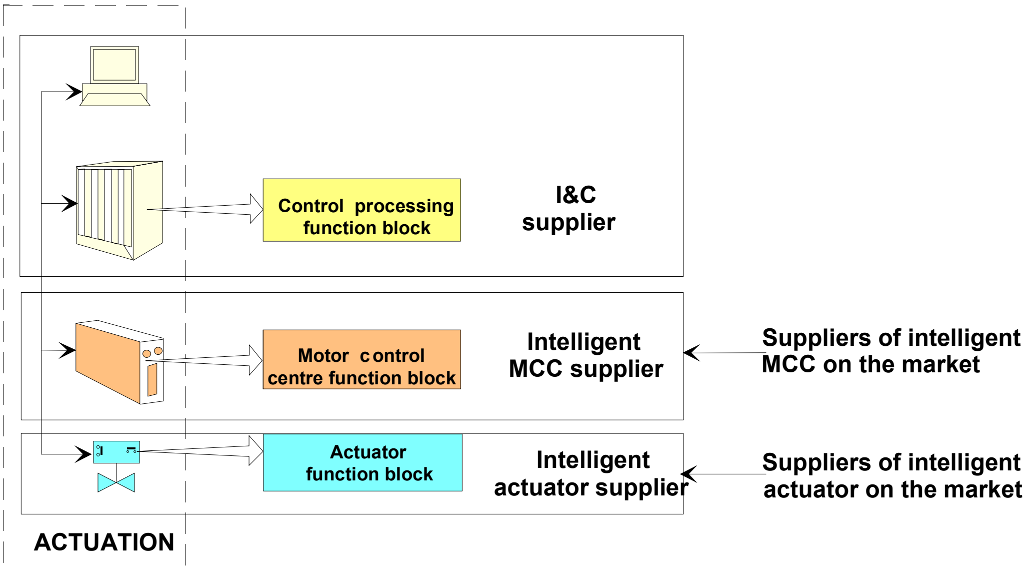


Figure A.15 – Design of an actuation and its AB from off-the-shelf actuation devices

This is typically the case for the different types of actuation. For each type of actuation different suppliers delivering actuation devices. I&C engineers select, off the shelf, actuation devices in conformances with the requirements specified by process engineers. Whatever the actuation devices are, these devices will be integrated together and will have to operate together to achieve the actuation required by process engineers. To inter-operate, the internal blocks of the actuation devices should be built in conformance with an IEC standard. IEC 61804 series allows actuation device suppliers to build internal blocks. To achieve a consistent actuation, the internal blocks are implemented into actuation devices; they are connected together via hardwired point-to-point data connections or via busses and they inter-operate.

Figure A.16 summarizes that within each FRD, process engineers describe the requirements and the performances of each AB independent of implementation because process engineers do not know in which I&C devices the ABs will be implemented.

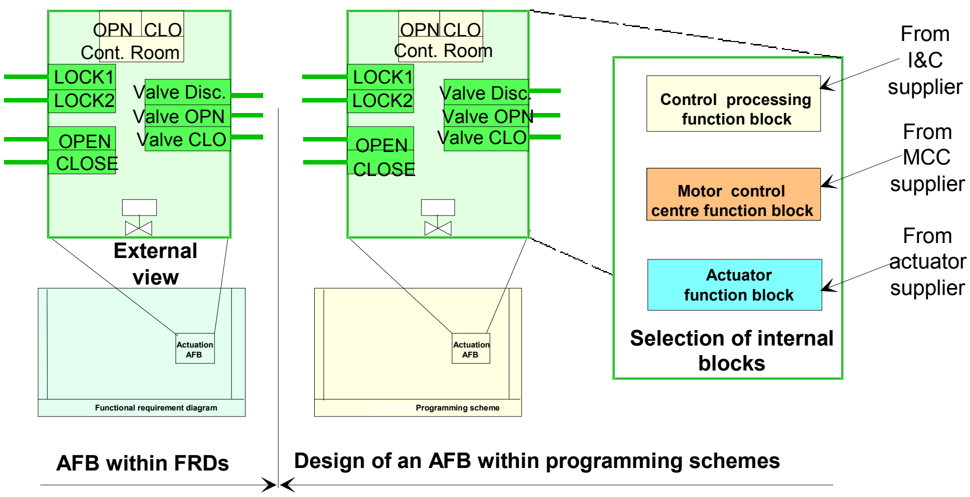


Figure A.16 – Designing internal behaviour of actuation AB from actuation devices off the shelf

For the design of each actuation AB, I&C engineers should select from suppliers, off the shelf, actuation devices in conformance with the requirements and the performances specified by process engineers. I&C engineers select actuation devices and the internal blocks of the actuation AB.

Once the internal blocks are selected, I&C engineers have to verify that inputs and outputs of the internal blocks are compatible, then I&C engineers link together the different internal blocks and verify the inter-operation of these internal blocks is consistent with the required actuation AB.

In some cases, where existing internal blocks would not satisfy requirements, I&C engineers should add extension or create internal blocks, for example, into I&C devices.

As summarized in Figure A.17, I&C engineers have also to network the actuation AB to EFBs and ABs of the control function. The inputs of the actuation AB are coming from outputs of upstream EFBs and ABs. The outputs of the actuation AB are going to inputs of downstream EFBs and ABs. The inputs of the actuation AB are connected to inputs of internal blocks and the outputs of the actuation AB are connected to outputs of internal blocks.

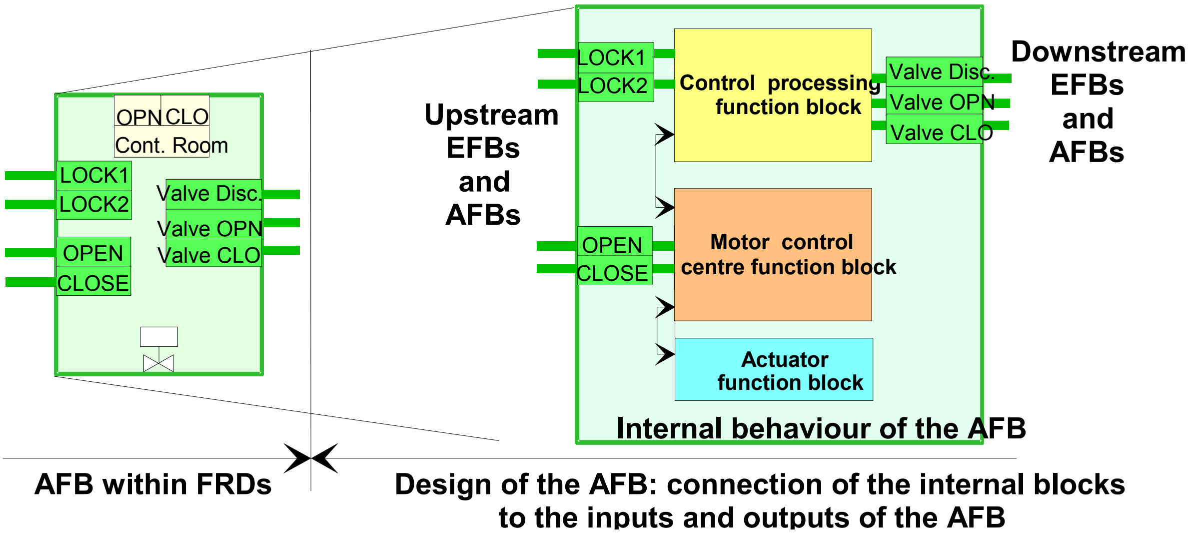


Figure A.17 – Networking of the internal blocks inside the AB and with upstream and downstream EFBs and ABs of the control function

In this example, the actuation AB should be composed of three internal blocks. The control processing internal block should be embedded into an I&C device, this block should support the Human Machine Interface, different types of commands (auto, manual, command from sequence), control data (state and status of the actuation for example) and orders to the Motor Control Centre. The Motor Control Centre internal block should be embedded into the Motor Control Centre, this block should support the priority of orders, control data (state and status of the Motor Control Centre). The actuation internal block should be embedded into the intelligent actuator, this block should support control data (state and the status of the intelligent actuator).

This actuation AB has two types of interfaces, one interface with operators, one interface with the actuator (see Figure A.18).

The operator interfaces should be explicitly summarized, within the FRDs and the programming schemes, on the top of the graphic symbol of the actuation AB by tags (one tag could summarize the type(s) of operator interface, another tag could summarize the type(s) of command available from the operator interface).

For the interface with the actuator, currently there is no information on the graphic symbol of the actuation AB to explicit the interface with the actuator. The actuator internal block should be connected to the instrumentation (torque and position) of the actuator, this data connection should not be explicit on the icon of the actuation AB within FRDs and programming schemes.

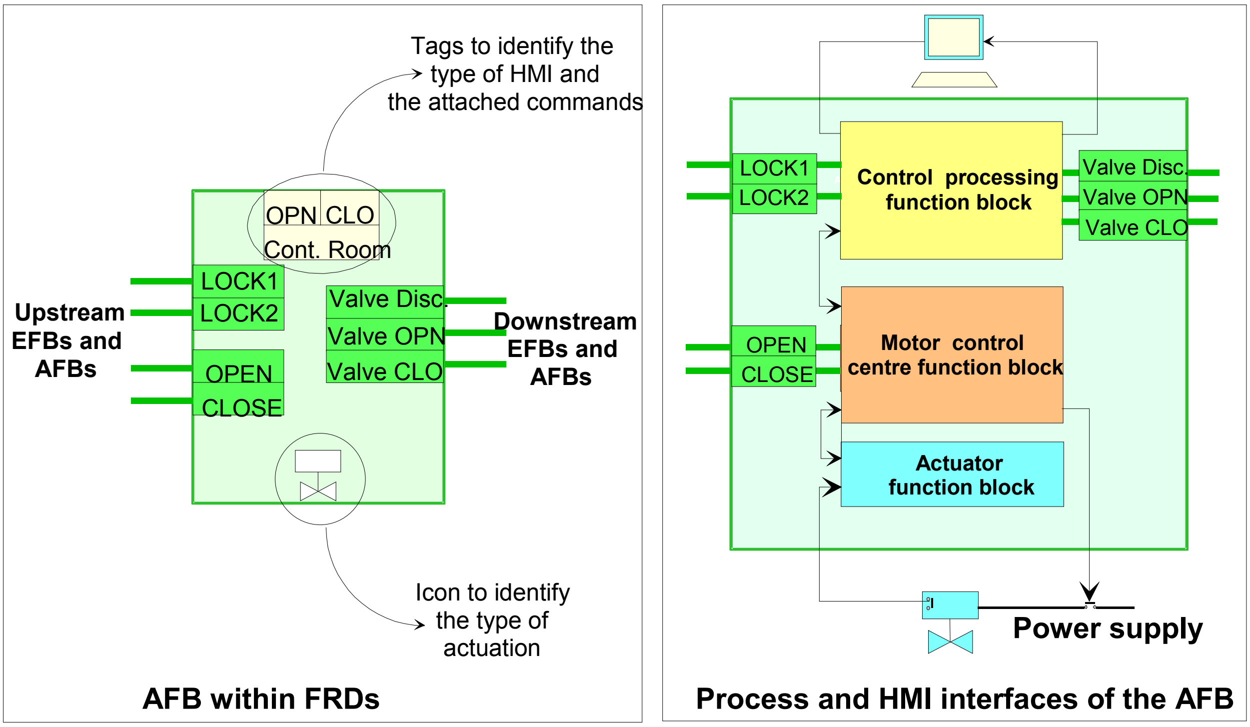


Figure A.18 – Actuation AB graphic symbol and implicit internal description

In Figure A.19, using the reference model of IEC 61499-1, the representation of the actuation AB with the distribution of the internal blocks into the I&C devices is shown.

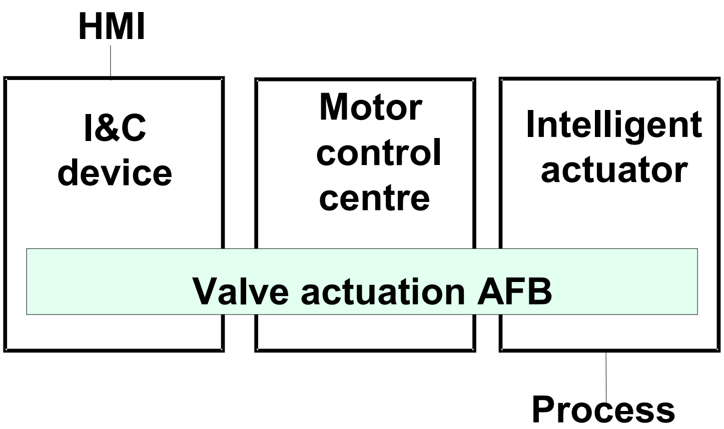


Figure A.19 – OFF/ON value actuation AB using the IEC 61499-1 system model

Figure A.20 is a representation of the internal blocks of the actuation AB using the model reference of IEC 61499-1.

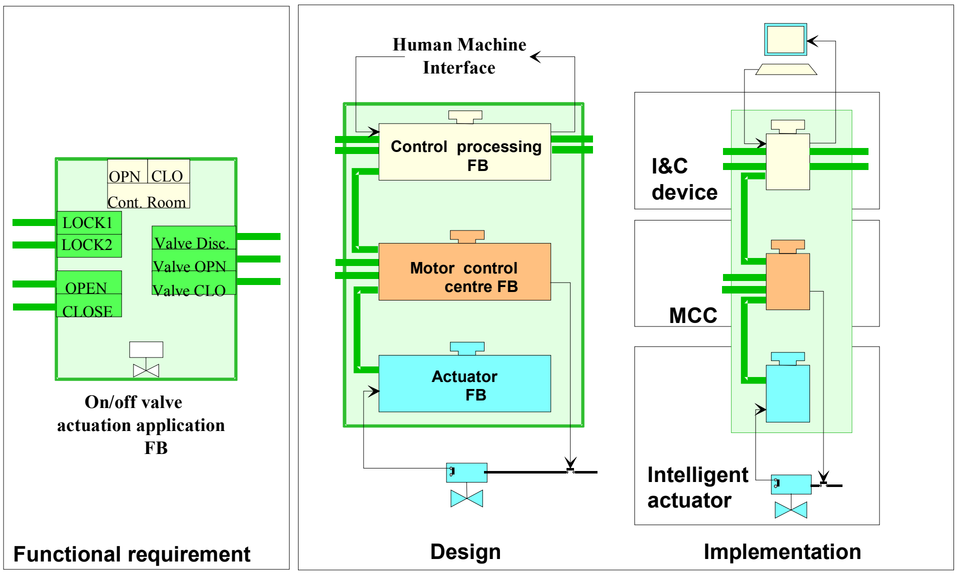


Figure A.20 – OFF/ON value actuation AB using the IEC 61499-1 system model

A.1.2.1.3 Detailed design of a safe switch over 2/3 AB

Figure A.21 gives an example of an AB without process interface nor HMI, a safe switch over 2/3. In the case of standardization of the switch over 2/3 functionality as EFB, this EFB is implemented in a single programmable controller.

For safety applications, safe switches over 2/3 implemented into several PLC are necessary to avoid a common failure. In this case, safe switch are built over 2/3 AB. In this case this switches over 2/3 AB is composed of three internal blocks. These three internal blocks should be implemented into three PLC.

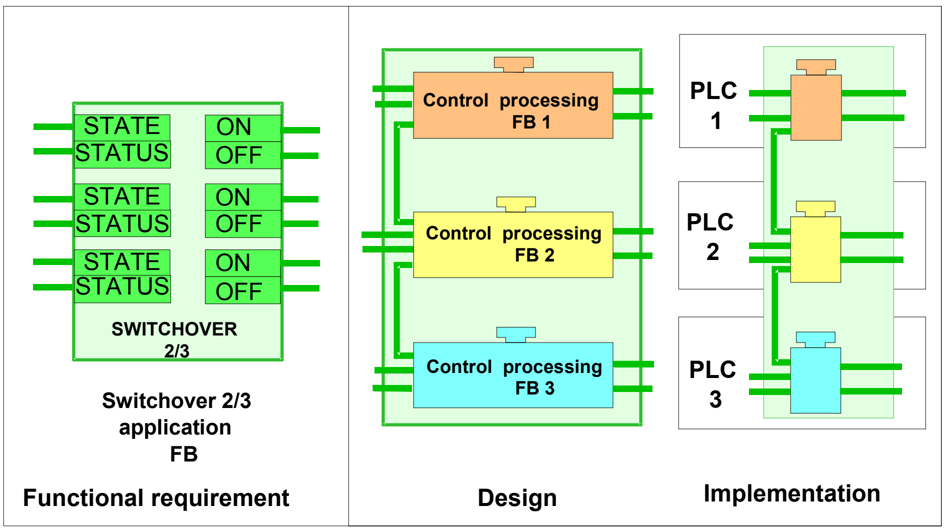


Figure A.21 – Example of a safe switch over 2/3 AB

Figure A.22 provides the representation of an actuation AB using the reference to the model of IEC 61499-1.

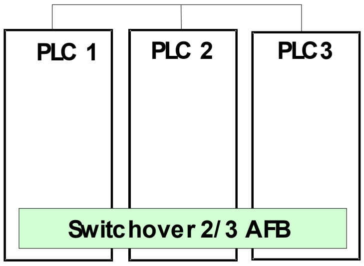


Figure A.22 – Switchover 2/3 AB using the IEC 61499-1 system model

A.1.2.2 Implementation of a control function into I&C devices

A.1.2.2.1 Overview

The architecture of an I&C system is currently a set of equipment distributed into different layers:

– layer 2: the HMI;
– layer 1: the processing;
– layer 0: the power interface and the field devices.

With the arrival on the market of intelligent devices, these I&C layers will evolve. The emerging technology will facilitate the integration of current isolated islands of control, maintenance and management (technical and financial).

Layer 2 will support the control, maintenance and management HMI. Layer 1 will support control, maintenance and management processing. Layer 0 will support intelligent power interfaces and field devices.

The following subclause focus on control functions, as well as maintenance and management are considering as extensions on the same principles.

A.1.2.2.2 From FRD to complete design of a control function

As an example, folio 4 dedicated to a single control function "to control the pumps" is presented. The FRD is shown in Figure A.23. The control function "to control the pumps" is described as a network of EFBs and ABs.

The design of the ABs should be completed by I&C engineers when the I&C architecture and the different types of devices are selected. Moving from the FRD to the programming scheme, I&C engineers design the ABs taking advantage of existing devices for the pump actuations and of existing libraries within the I&C system for the 2/3 safe switchover, in conformance with the requirements and with the architecture of the control system.

For the complete design of the control function, the four ABs have to be designed. The internal blocks of the pump AB should have the blocks embedded into the different devices of the pump actuations. The internal blocks of the 2/3 switch over AB should be within the library of the I&C supplier. Figure A.23 shows an example of a complete design of the control function.

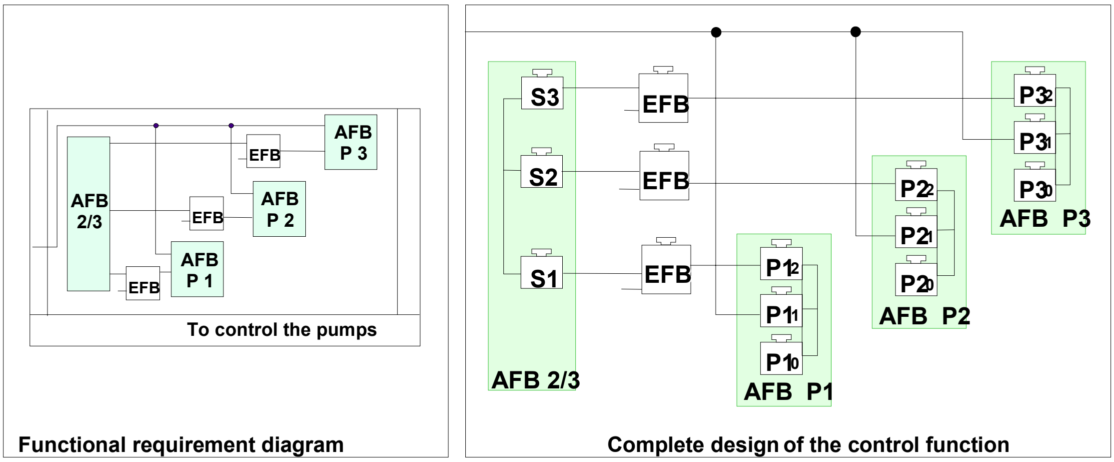


Figure A.23 – From a FRD to a programming scheme using IEC 61499-1 system model

For each pump actuation AB, the three internal blocks should be distributed into the three layers of the I&C system, for example, layer 0 into a MCC, layer 1 into a PLC and layer 2 into a HMI, as summarized in Figure A.24.

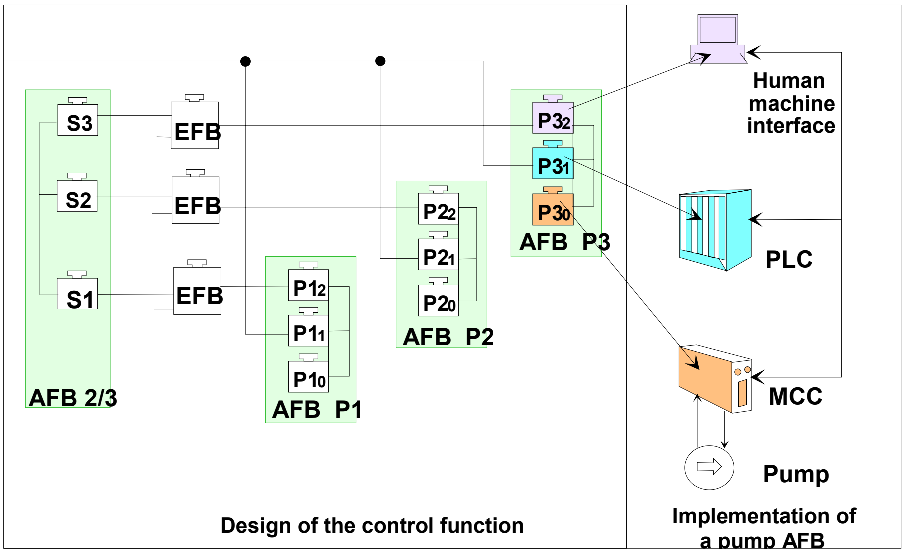


Figure A.24 – Distribution of the internal blocks of a pump AB

A.1.2.2.3 From the complete design to the distribution of the internal blocks of the control function into I&C architecture

Once the design of the ABs is completed, the EFBs and the internal blocks of the ABs will be distributed into the I&C system architecture

The allocation of the blocks into I&C devices is carried out by I&C engineers taking into account safety, availability and performances constraints required in the FRDs.

In Figure A.16 it is considered that, for safety reasons of safety constraints, the switchover 2/3 AB should be distributed into three different PLCs and the pump actuation ABs should be implemented into separate equipment, to avoid a common failure.

For the pump 1 actuation AB, the internal block $P1_0$ (white) will be distributed in the layer 0 into the MCC1, the internal block $P1_1$ (blue) will distributed in the layer 1 into the PLC1, the internal block $P1_2$ (pink) will be distributed in the layer 2 into the HMI.

For the pump 2 actuation AB, the internal block $P2_0$ (white) will be distributed in the layer 0 into the MCC2, the internal block $P2_1$ (yellow) will distributed in the layer 1 into the PLC2, the internal block $P2_2$ (pink) will be distributed in the layer 2 into the HMI.

For the pump 3 actuation AB, the internal block $P3_0$ (white) will be distributed in the layer 0 into the MCC3, the internal block $P3_1$(brown) will distributed in the layer 1 into the PLC3, the internal block $P3_2$ (pink) will be distributed in the layer 2 into the HMI.

For the switchover 2/3 AB, the internal block S1 (blue) will be distributed in the layer 1 into the PLC1, the internal block S2 (yellow) will be distributed in the layer 1 into the PLC2, the internal block S3 (brown) will be distributed in the layer 1 into the PLC3.

The EFBs will be distributed into layer 1, into a PLC (see Figure A.25). Each EFB will be implemented in the same PLC as the upstream and downstream internal block. For example, the EFB between the S1 internal block of the switch over 2/3 AB and the $P1_2$ internal block of the pump 1 actuation AB will be implemented into the PLC1.

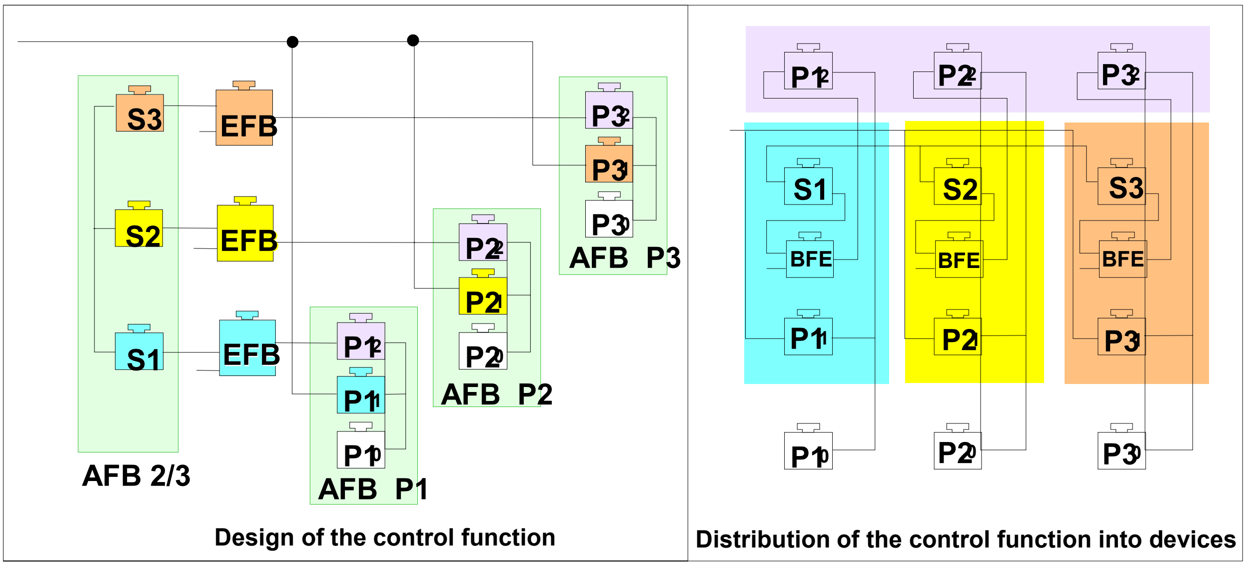


Figure A.25 – Distribution the EFBs and ABs of the control function "to control the pumps" into the I&C architecture

During the distribution of the EFBs and internal blocks of the ABs into the I&C devices, the communication needs between the I&C devices appear clearly. At this stage, I&V engineers know exactly the types and the amount of data exchanged between the different I&C devices. On this basis, I&C engineers should be able to validate the throughput available in the I&C devices, in accordance with the communication needs.

A.1.2.2.4 From the distribution to the implementation of a control function into I&C devices

Once the I&C engineer has completed the distribution of the EFBs and of the internal blocks of the ABs, the control function is distributed into the devices of the I&C architecture. The control function can be implemented into the equipment of the I&C system devices (see Figure A.27).

For layer 2, there is dedicated equipment as HMI multiplexes the three internal blocks $P1_2$, $P2_2$ and $P3_2$ of the three pump actuation ABs.

For layer 1, PLC1 multiplexes the internal blocks P11 of the pump 1 actuation AB, the internal block S1 of the switchover 2/3 AB and the EFB between these two blocks. PLC2 multiplexes the internal blocks P21 of the pump 2 actuation AB, the internal block S2 of the switch over 2/3 AB and the EFB between these two blocks. PLC3 multiplexes the internal blocks P31 of the pump 3 actuation AB, the internal block S3 of the switchover 2/3 AB and the EFB between these two blocks.

For layer 0, the MCC1 is dedicated to the internal block P10 of the pump 1 actuation AB, the MCC2 is dedicated to the internal block P20 of the pump 2 actuation AB and the MCC3 is dedicated to the internal block P30 of the pump 3 actuation AB.

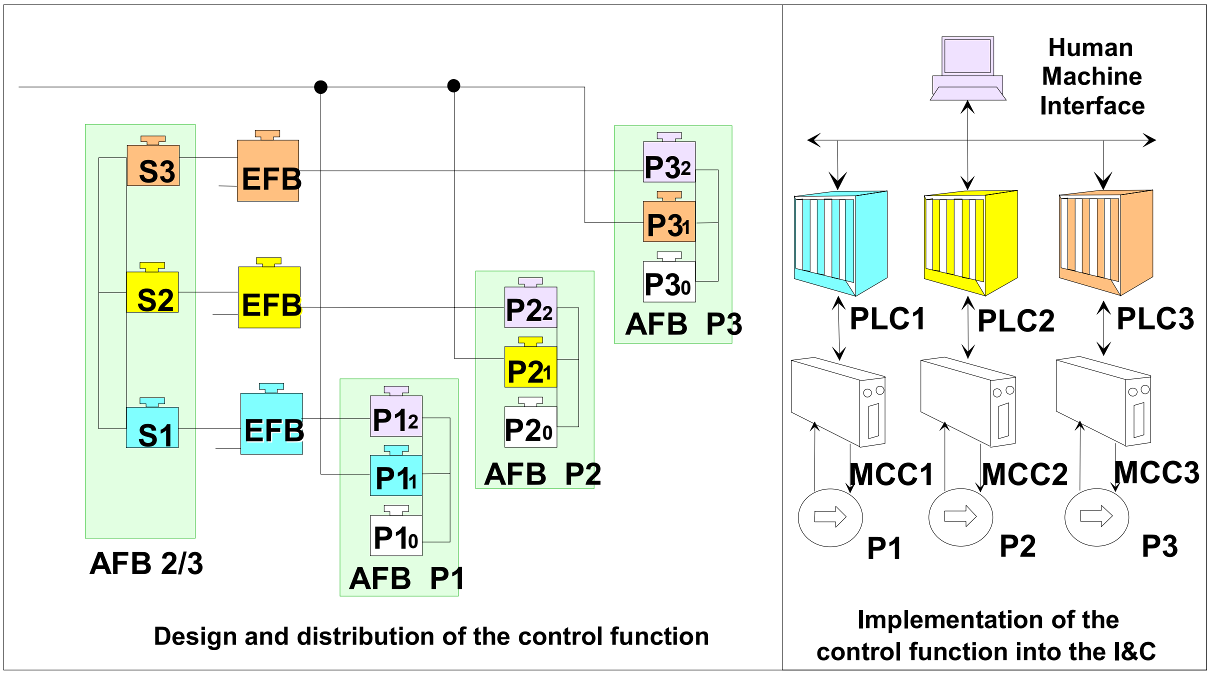


Figure A.26 – Implementation of the control function into the I&C devices

Using the reference model of the IEC 61499-1 series, the control system can be represented (see Figure A.26). Figure A.27 is an example of the implementation of the control function "to control the pumps„ into the I&C system devices.

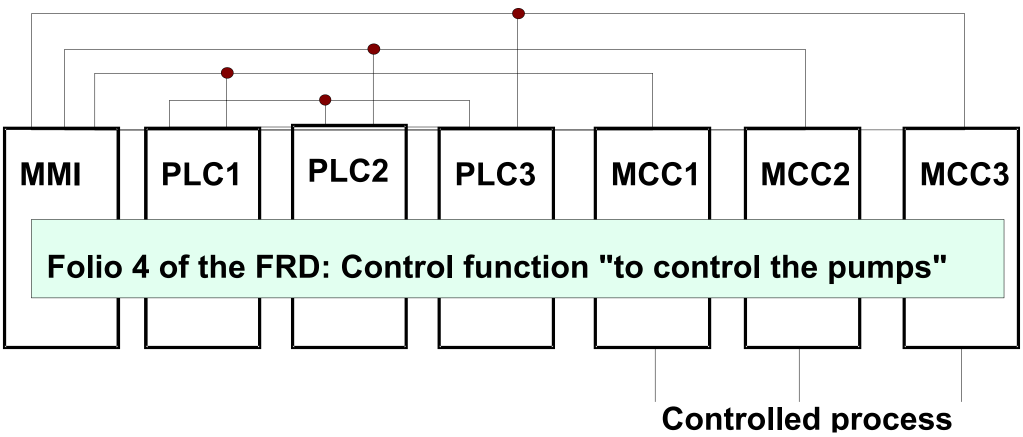


Figure A.27 – Distribution of the control function "to control the pumps" using the IEC 61499-1 model

A.1.2.2.5 From the FRD to the implementation of a control application into I&C devices

For each control function of the FRDs, I&C engineers should design the ABs. Internal blocks of the ABs are selecting taking into account the capabilities off the shelf of available I&C devices in accordance with the requirements. For some ABs I&C engineers should define internal blocks. Once the design of ABs is finished, the programming schemes implementable into the I&C devices are available, as summarized in Figure A.28.

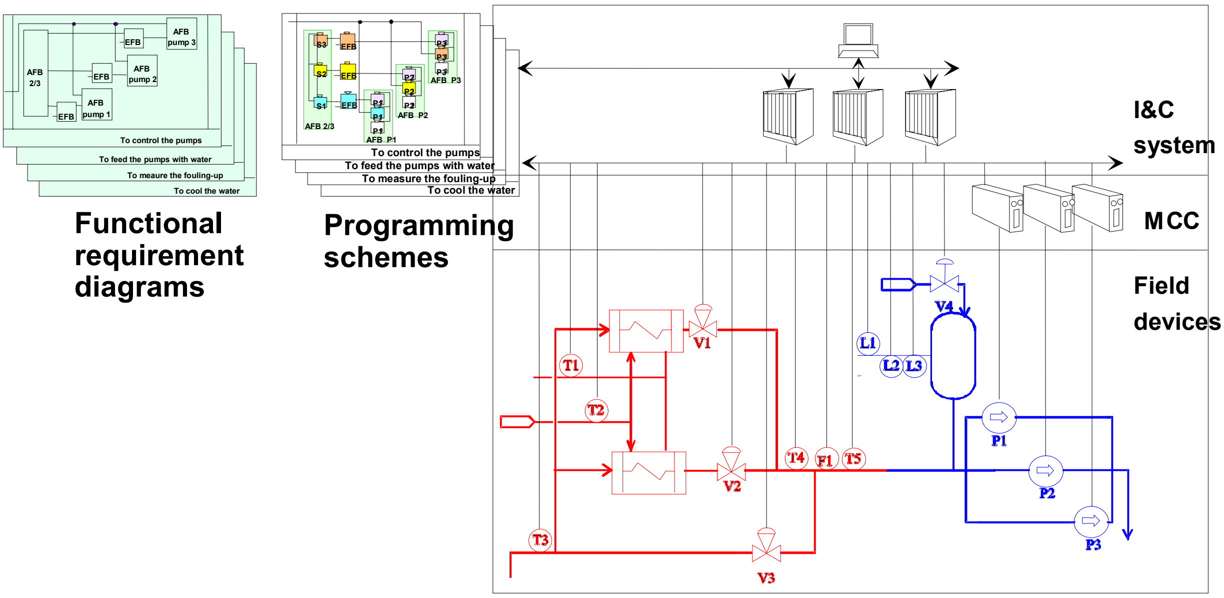


Figure A.28 – From FRDs to programming schemes

A.1.2.3 Requirements for the design and the implementation of control functions

The internal behaviour of EFBs and ABs should be described with a unique standardized formal language. It should be noted that IEC 61131 does not meet such requirements. A future IEC standard should propose construction rules and a language (syntax and semantic) to describe the internal behaviour of either EFBs or of the internal blocks of ABs.

With such a standardized language it should be possible to build ABs, to store these ABs into private libraries, to use these ABs in applications, to distribute and to implement these ABs into different technological supports compliant with this standard. This standard should be complete and will take into account the normal and abnormal operation of the technological supports, the processing and the communication supports.

This standard should take into account all the technical aspects of the distribution and cooperation of distributed and synchronized internal blocks of ABs:

- a formal language allowing I&C engineers to describe the normal and abnormal behaviour of the EFBs and of the internal blocks (types of variables, statement, syntax, semantic, formal proof, etc.);
- a standardized set of EFBs, including normal and abnormal operation;
- rules to build normal and abnormal operation of internal blocks;
- standardized behaviour of EFBs and internal blocks for abnormal operation of the device supporting these blocks, for example:
  - initialization, hot reset, cold reset;

  - inputs, output hardware failures, component failures;
  - miscount, overflow, underflow, (special care for round off error and propagation of round off);
- rules to synchronize and to order EFBs and ABs within a device and distributed into devices;
- rules to manage internal data of EFBs and ABs;
- rules to share data between EFBs, ABs and internal blocks of ABs;
- rules to exchange data between EFBs and ABs within a device and distributed into devices;
- rules to synchronize and to order the internal blocks of ABs within a device and distributed into devices;
- rules to exchange data between internal blocks of ABs within a device and distributed into devices;
- rules to exchange data with other EFBs and ABs.

Annex B
(informative)

FB functional requirements: the user's view

B.1 Overview

Modelling a system is an extensive task. A single model cannot capture all the information needed to describe a system. IEC 61499-1 specifies a generic standard for distributed FB applications for Industrial Measurement and Control Systems (IPMCS). The scope of IEC 61499-1 includes the following models:

- application model;
- device model;
- resource model;
- FB model;
- system model.

The detailed approach adopted by IEC 61804-2 in identifying the functional requirements was to take into account a system-user view model which describes the user's expectations.

This view provides a generic description of the use of the system (functionality required). This view is central. The final goal of the system is to provide the functionality described in this view along with some non-functional properties. Hence, this view affects all the others.

This view is also proposed for conformance test between the FB specification and the user expectation (asking "Which functional unit defined by users is covered with this FB specification?" and "To what extent is this functional unit covered by the specified FB?").

This annex incorporates all the concepts regarding this system-user view model as they were considered during the requirement specification.

B.2 Definitions

B.2.1 General

The following definitions complement those given in IEC 61499-1 for the purpose of this annex.

B.2.2 List of specific terms

B.2.2.1 IAM (Intelligent Actuation and Measurement)

The sum of all the requirements for all the actuations and measurements needed for the CMM system.

NOTE Intelligent here means provided with all the functionalities, as users need them.

B.2.2.2 Access right management function

This function provides the needed handshake between the different components of the IAM and the users of IAM. As a consequence of the system requirements, for each IAM operating mode some of the IAM functions are active and the others are inactive.

The actual operating mode of IAM and its components are affected by a management function of the automation system. This function is also responsible for a coordinated transfer from the actual Operating Mode to another one.

All interactions between the agents and the IAM (or part of it) are conditioned by a relationship to the operating modes in such a way as to enable the effect of only the authorized interaction for those functions which can be active in each operating mode as specified in the system requirements. Therefore, for each operating mode of IAM (or part of it)

– a set of IAM functions are activated, and
– only the specified interactions are completed; the other (non permitted) interactions have no effect.

It is proposed that a matrix should be standardized, where rows should list the potential operating modes and columns should list the potential interactions with IAM (or part of it). Where the authorizations of the interactions should be determined (by signing the intersections of the corresponding rows and columns) on the basis of the requirement of each user (for example, while parameterizing the system).

B.2.2.3 Agent

Whoever (for example, the operator) or whatever (for example, the device) uses the IAM, it is characteristic of each agent to interact with IAM along specific phases of the IAM life cycle.

B.2.2.4 Corrective maintenance

Maintenance carried out after a failure has occurred and intended to restore an item to a state in which it can perform its required function.

B.2.2.5 Functional validation (F. VAL)

- For measurement

  The F.VAL is a function that verifies the coherence of the measurement under check and a set of other measurements coming from the process under control. The link among such measurements is checked using a suitable model of the process under control.
- For actuation

  The F.VAL is a function that verifies the coherence among the valve's state and the measurements (coming from the process upstream and downstream the valve), which defines the actual effect of the state of the valve.

B.2.2.6 Maintenance

The combination of all technical and corresponding administrative actions intended to retain an item in, or restore it to, a state in which it can perform its required function.

NOTE 1 See "preventive maintenance" and "corrective maintenance" for a more detailed definition of maintenance.

NOTE 2 The required function may be defined as a stated condition.

NOTE 3 Here maintenance focuses on IA/IM channels and all their parts.

B.2.2.7 Operational validation (O. VAL)

- For measurement

  The O.VAL is a function that checks the coherence of a credible measurement against other redundant (two or more) measurements coming from redundant transducers (or redundant parts of a transducer).

  NOTE 1 This is a bottom-up process.

- For actuation

  The O.VAL is a function that verifies any state or variations of the actuator for consistency with the commands sent by operator or reflex processing devices.

  The O. VAL provides also for checking that the performance of the valve-positioning loop does not deviate from the design range.

  NOTE 2 This is a top-down process.

B.2.2.8 Preventive maintenance

Maintenance performed in accordance with pre-defined criteria (knowledge base), in order to reduce the probability of either equipment failure or service degradation.

- Scheduled maintenance (time or activity directed): preventive maintenance performed either on a pre-defined schedule or on units of use (for example, number of start-ups).
- Condition-based maintenance (conditional or condition directed): preventive maintenance performed on the basis of the documentation of the performance degradation of the equipment (resulting from, for example, auto-diagnosis, wear measurement). It is based on a proper visibility of gradual, partial and intermittent failures.

B.2.2.9 System (or device) properties

Properties segregated into several groups, to facilitate the assessment of a system as shown in Figure B.1.

NOTE 1 This is considered in IEC 61069-1 and IEC 61069-5.

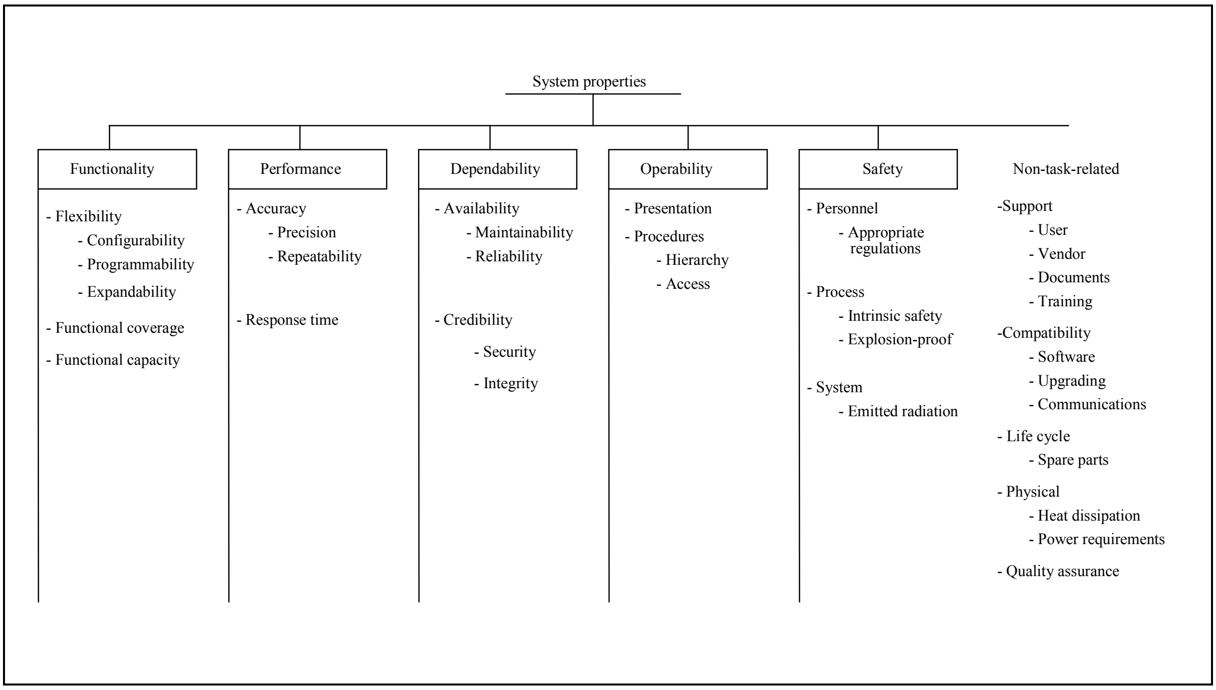


Figure B.1 – System properties

NOTE 2 The properties of the IAM system are as defined for the whole automation system. Therefore, the properties of the IAM system are dependent upon the properties of each individual part of each IAM channel and the way in which these parts cooperate in performing the channel functions. Of course, the properties of the channel may differ with respect to each of its functions.

B.2.2.10 Technological validation (T. VAL)

- For measurement
  - The T.VAL is a function that provides for a status checking of the electronics, the power supply, and the variable of influence associated to the transmitter, verifying that the related parameters are within the normal conditions.
  - T.VAL assures that the measurement has been produced by a transmitter that is working in detected conditions documented by the transmitter status report.
- For actuation
  - The T. VAL is a function that provides for a continuous status checking of the electronics, power supply and variables of influence associated to each part of the actuator:
    - processing capabilities;
    - power interface;
    - motor;
    - valve.
  - The T. VAL assures that the actuator operates under detected conditions in actuating and in reporting its state by the dedicated instruments.

B.2.2.11 Validation

a) A/IM-channel validation

Validation is a function, which is part of several IAM macro functions (state f., status f., etc.). It aims at checking the behaviour (status) and the quality of products (states: data, action, etc.) of each entire measurement channel and actuation channel, which are only partially implemented in the actuators and transmitters themselves.

An important example of behaviour is the production of the measurement (or actuation) with a required timeliness. Here timeliness defines to which extent the measurements of different quantities (or actuations) are produced at the same instant in time (within a specified jitter) or one measurement (or actuation) is produced with a given sampling period (with a specified jitter).

As regards each IA/IM channel, the validation produces the information named "status of the channel" (see Figure B.2).

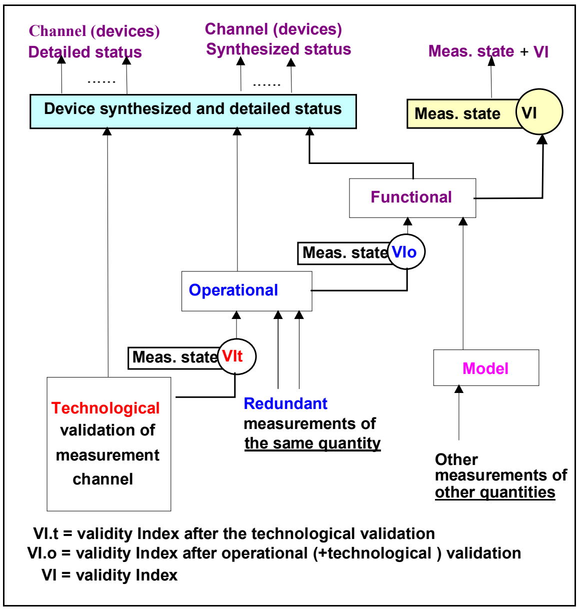


Figure B.2 – IA/IM channel validation

With regard to the IA/IM channel products (actuation value, measurement value, etc.), the validation determines the quality of these products, expressed by their VI (see measurement uncertainty, VI).

The IA/IM channel validation can be seen as being partitioned into the three levels defined in this dictionary and shown in Figure B.2:

- technological validation;
- operational validation;
- functional validation.

NOTE 1 Partitioning evolves (for example, the operation validation could be seen as a part of the technological one).

NOTE 2 At implementation level, the validation function is distributed among the field devices and the rest of the channel.

b) Field device validation

Field device validation is a part of the complex function of IA/IM channel that checks both their behaviour and the quality of their products (data, action, etc.). For each field device, the validation produces the information named "status of the device", and, with regard to the field device products, each product is completed with its partial VI, as result of the partial validation solved in the field devices.

B.3 Functional view of process control applications

B.3.1 Overview

Technologies, such as digital field communications (for example fieldbus), intelligent field devices (transducers, actuators, switch gears, etc.), software embedded in controllers are producing a powerful evolution of the automation. They, in fact, enable the manufacturers to transfer into the field devices part of the functions which have traditionally been located, even if only partially and less effectively, in the centralized computers. This is a huge opportunity for the market to provide better support to the user requirements provided that these are more explicitly and completely specified by the end-users and well understood in terms of implementation.

B.3.2 Relationship between functional requirements and FBs specifications

B.3.2.1 General

The functional requirements which are expressed in this Technical Specification are defined at a certain abstraction level which should make the definition well understood by both end-users and vendors.

This abstraction level differs from the FBs abstraction level, which is directly used by vendors and system integrators.

The correspondence between the “functional requirements” provided at a higher level of abstraction, and defined in this Technical Specification, and the "FBs" provided at a lower level of abstraction, and defined in IEC 61804-2, is not necessarily one-to-one. The correspondence between the two levels of abstraction may be one-to-one, one-to-many or many-to-one.

The functional requirements have to be seen as the target to be reached with distributed applications.

It is mandatory that in the definition of each FB a clear relation to the addressed part of the functional requirements described herein is stated.

The distribution addressed during the requirements was at the level of the field instrumentation because this is already (at least in part) a reality. The same logic approach is applicable at the level of control blocks, which is expected to become a reality in a near future.

Hereafter the user vision model, which was used as the basis for the description of the functional requirements, is described.

B.3.3 Intelligent Actuation and Measurement (IAM) and Control Maintenance Management (CMM)

B.3.3.1 Concept

Key concepts of the user vision model include the following.

– CMM concept, describing the integrated user needs related to control for plant operation, maintenance for on-site trouble-shooting and technical management.

– IA channel/IM channel concept, describing the integrated user needs related to each actuation or measurement as specified for the CMM applications; the notion of channel corresponds to whatever proper integration of field device(s) with some complementary functions in the systems (controllers) which is able to answer the defined user needs.

By definition, automation complements the human organization to achieve a defined goal. Hence, user requirements identification starts from an analysis of the logical sum of supports needed by all the human organizations which are interacting with the automation system. The reference functional architecture for distributed automation systems shown in Figure B.3 is the user vision model. The system is shown as the proper cooperator of the human organizations. The activity of the human organizations is proposed to be described with tasks categorized in seven classes: seven macro-functions with the same names (access rights, identification, configuration, parameterization, tests on demand, measurement /actuation, diagnostics) represent the corresponding functional support to be provided by the system.

This applies also to the actuation subsystem and to the measurement subsystem and hence to all the IA channels and IM channels composing the subsystems.

Once described for a channel as a whole, the necessary functional support represents the target to be reached by summing up the functional contributions given by all the components used by the system integrator to construct the channel.

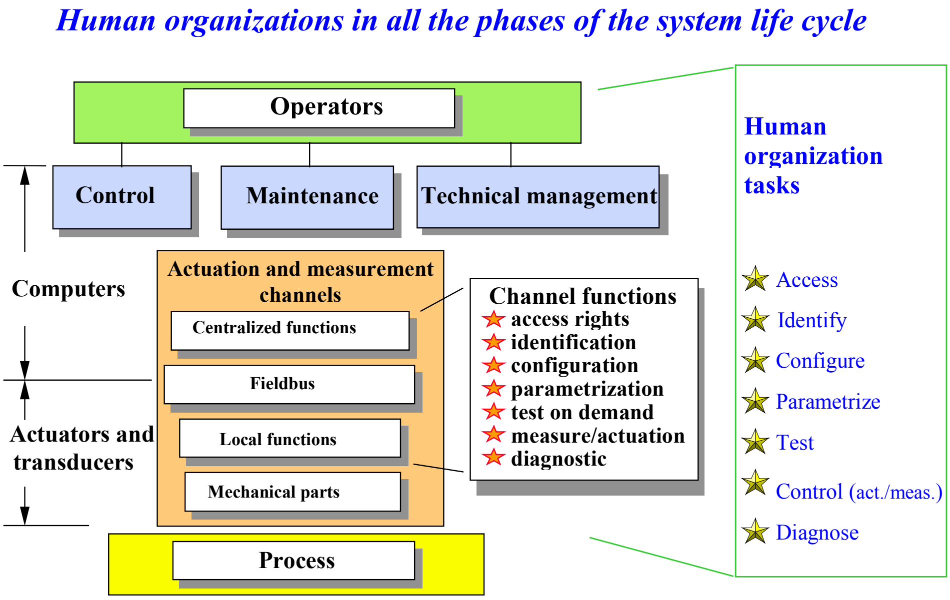


Figure B.3 – Reference functional architecture for distributed automation systems

The reference functional architecture was used as the basis for the identification of the list of all the possible functional requirements. It is also the guide for the users in identifying their functional requirements as a selection out of the complete list.

This Technical Specification is intruded not only for the end-users but also for all the others who are "using" each device integrated into respective IA/IM channel along its life cycle: the manufacturers (tests in factory), the system integrators, the contractors (commissioning tests), etc.

B.3.3.2 CMM (Control maintenance management)

The most significant CMM functional requirements are described hereafter. These functional requirements represent the final goal to be achieved with a proper help provided by the implemented IA/IM channels.

a) Control requirements

   They are extended by including the automated treatment of abnormal situations of the field instrumentation. This automated treatment has to correspond to the treatment, which is currently carried out by the control operator on the basis of the result of the maintenance operator's investigations. Benefits result, as an example, from the timeliness of the automated action. The use of the traditional automated system management approach and exception treatment rules is extended in order to include the field parts of the system. In general, one or more levels of degradation of the channel could be defined by the users for each application according to the corresponding behaviour of the channel which comply with the application needs.

b) Maintenance of the field instrumentation

   This will be helped by the direct and immediate visibility of the actual status of each replaceable part of each channel. For each replaceable part, predefined levels of degradation are recognized and documented. Thus, preventive maintenance can be organized as a condition-based maintenance. The maintenance interventions are only planned when they are really necessary. Interventions are helped by provision of additional information. Besides, information useful to speed up the repairing of the replaced parts may be collected as defined and agreed in the maintenance contract between the user and the repairer.

c) Technical management

This relates, in particular, to the evaluation of performance and reliability of specific items of equipment. For example, comparison of the number of cycles achieved before failure of two valves from different suppliers and performing an identical function in the same operating environment. Collection of the history of the use and “health” (condition) of each replaceable part of each channel is the basic requirement. This data is currently very limited. As a result, it is difficult to do the detailed and valid statistical analysis required for the management decisions and for the plant equipment modifications.

B.3.3.3 IA/IM channel

Each channel has to provide the functional support shown by the reference functional architecture.

Figure B.4 shows the minimum level of granularity indispensable for the identification of the user requirements. Mainly, during the specification of the functional requirements, this level of granularity was used. Sometimes, a thinner granularity was used.

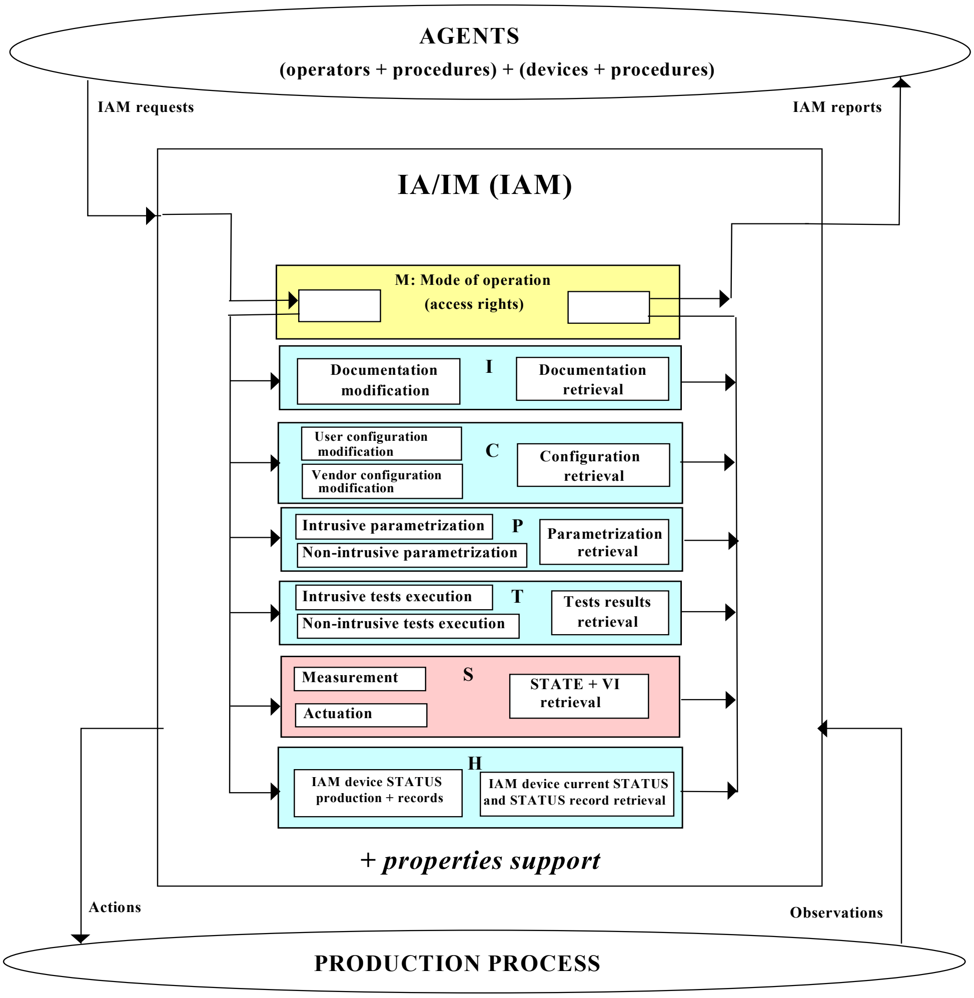


Figure B.4 – User view of the major component functions of each IAM function

NOTE 1 M = MODE OF OPERATION MACRO-FUNCTION

To support coordinated management of the modes of operation of the IAM system and its parts under control of the overall system management. Basically, this function manages a pre-defined access of each agent to any IAM functions listed above. Users, as part of the user requirements, define this access.

I = IDENTIFICATION MACRO-FUNCTION

To support storage and retrieval of information describing the IAM system (data sheet).

C = CONFIGURATION MACRO-FUNCTION

To support establishment or modification or visibility of the IAM system functional configuration. To be seen as System (or Channel or Device) Configuration.

P = PARAMETERIZATION MACRO-FUNCTION

To support establishment or modification or visibility of the IAM system parameterization. To be seen as system (or channel or device) parameterization

T = TESTS ON DEMAND MACRO-FUNCTION

To support execution and reporting of results of either automated or semi-automated IAM tests.

S = STATE MACRO-FUNCTION

To support execution with validation of a measurement or actuation and result report of the measurement or actuation completed with a predefined VI.

NOTE This function is complemented with all the diagnostics and validation needed to support the specified VI and associated quality criteria.

H = STATUS AND HISTORY OF STATUS MACRO-FUNCTION

To generate information documenting the IAM status as defined in system (or device) status. This function is complemented with all the diagnostic and validation needed to support the visibility of IAM behaviour as requested from each

NOTE 2 An action of interpretation of these macro-functions is the key to bind the detailed specification of the IAM devices and the definition of IAM user requirements. Close cooperation is mandatory to guarantee clear definition of the relationship between the users expectations and the supporting functions and properties implemented in the IAM devices.

The IAM functions are aggregated in the seven macro-functions which can be seen both as "purpose" (or user expectations) and "action" (or manufacturer's interpretations).

In Figure B.5 an IM channel is drawn as a black box with the explicit list of the information exchanged with the CMM and the operators (called agents with only one word).

An IA channel would be represented in a similar way, with the major replacement of "Measurement + VI" with "Actuation + VI".

As already indicated, this functional view of the entire channel is what has to be provided with a proper integration of a number of components of the channel.

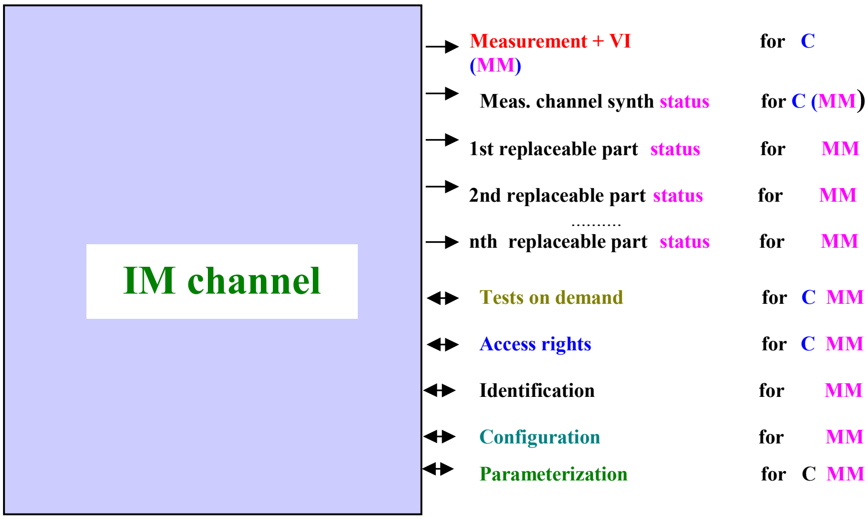


Figure B.5 – IM channel/CMM interaction: user vision

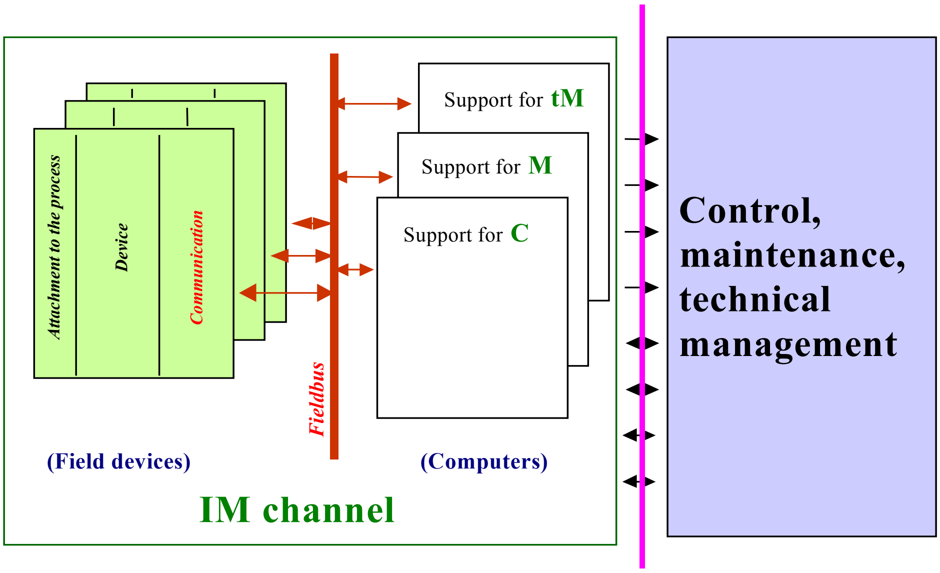


Figure B.6 – Distributed platform using fieldbus: physical composition

The technological composition of each IM channel necessary to perform each measurement as specified for the CMM applications is (see Figure B.6):

a) field parts:
   – attachment to the process (tubing, etc.);
   – sensor(s);
   – transmitter(s);
   – network (for example fieldbus);

b) complementary parts:
   – complementary processing to support the measurement;
   – complementary processing to support maintenance;
   – complementary processing to support technical management.

An IA channel has a similar composition, with obvious differences for the field parts, such as valves, motors, etc.

B.3.4 Channel and field-device functional units

As shown in Figure B.4, the field device contributes to construct the relevant channel.

In Figure B.5 the field-device functional unit reference model is shown, as regards a transmitter. The actuator model has a similar composition, with obvious differences for the field parts, such as valve, motor and actuator device.

Here the variety of data and functions in a transmitter and actuator is grouped in functional units. These functional units represent specific hardware or software components at a intermediate level of abstraction that is allocated between the higher level of the user vision represented in Figure B.3 and Figure B.4 and the lower level of abstraction of the FBs.

The description of the functional requirements is the result of the mapping of the channel model on all this functional unit device reference model. This is proposed as to help the mapping of the functional requirements on the FBs specifications.

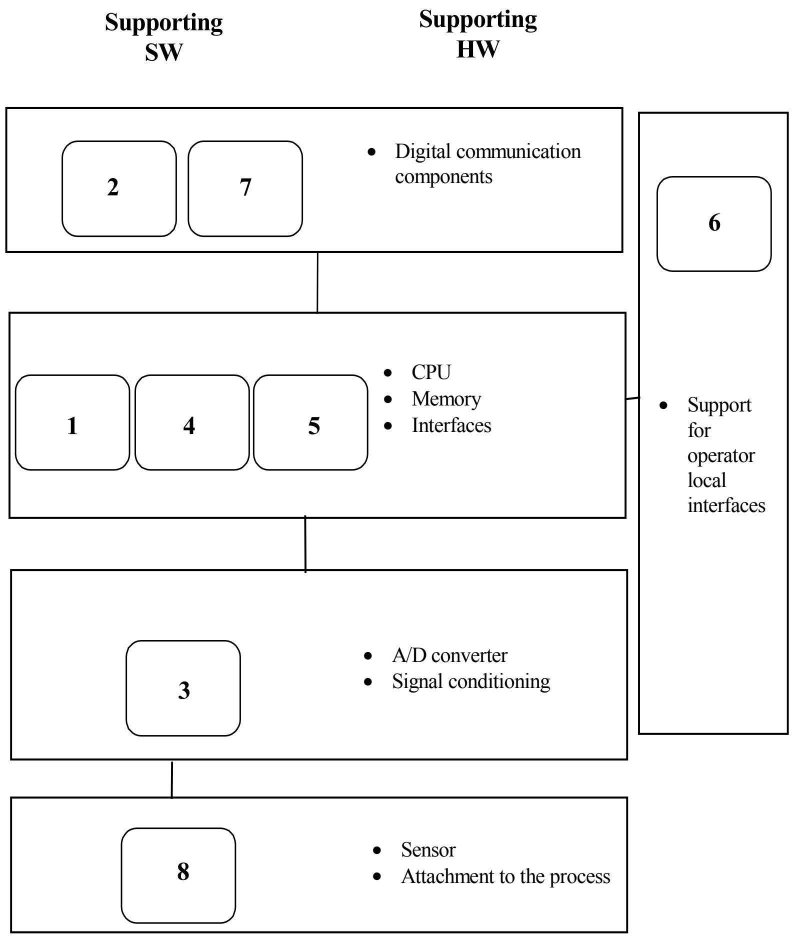
Of course, this field-device functional unit reference model has to be seen as complemented with the corresponding model covering the computer part of the channel and the field communication. This latter model shows the complementary functional units and the coordination of the management of all the channel parts, which is a system level coordination.

B.3.5 Field-device functional requirements list

The functional requirements reported in B.3.2 and B.3.3 were identified through the logical approach till here described. They concern the field device.

They are described by using a paragraph for each functional unit, with the only exception of the “device management” functional unit. This description, if complemented with the background given in this annex, should be sufficient to clarify the concrete meaning of the user requirements.

The technological evolution will enable the device manufactures to transfer into the field devices more functions as shown in Figure B.7.



Key:
1 Device management
2 Network management
3 Signal processing
4 Measurement information processing
5 Diagnostic and test support
6 Local operator interfaces attachment
7 Fieldbus communication functions
8 Measurement transduction

Figure B.7 – Intelligent transmitter reference model

In the case of the "device management" functional unit, because of the complexity of the requirements, the necessary functional decomposition is thoroughly addressed. The following aspects are defined.

- Identification management
- Configuration management
- Parameterization management
- Measurement management
- Time management
- Timeliness control management
- Failure management
- Mode of operation management
- Access rights management

B.3.6 System level functional requirements list.

NOTE 1 Technological evolution will enable the device manufactures to transfermore functions into the field devices. To help this evolution, see the following additional requirements.

NOTE 2 In some cases, there is the same description of a requirement for the device; of course, here it has to be seen at the system level.

The following requirements are at the level of the IAM channel seen as a whole.

Each channel has to provide all the requested functions, however distributed in the several parts composing the channel, and the coordination of a proper execution of the distributed groups of functions (across digital communications).

All the requirements are included in two main categories:

a) control, as regards whatever interaction with the control part, when operational, both
   - in the factory, and
   - in the plant.
b) maintenance and technical management, as regards all the other interactions both
   - in the factory, and
   - in the plant.

B.3.7 System requirements for control

B.3.7.1 Introduction

As far as requirements for distributed control applications are concerned, some major users' requirements have to be selected for the IAM channel to be analysed, such as managing access rights (to be negotiated with the person in charge of control), access to the state of the control algorithms and to the synthesized device status information. Requirements for system properties (dependability, etc.) applied to IAM channels are listed as well.

These requirements cover the entire device life cycle and therefore they partially concern the end-user and partially the manufacturer, the system integrators, etc.

A key issue is that the same real-time interaction among the distributed applications has to be guaranteed, whatever the communication standard selected. This requires that the description of the interaction of the application is extended to cover time synchronization aspects.

B.3.7.2 Measurement/actuation processing

B.3.7.2.1 General

Both macro-functions are defined in terms of some more or less complex processing; hereafter, major component functions are listed.

NOTE Both direct and indirect measurements require the same basic mechanisms.

B.3.7.2.2 Measurement validation

Up to three levels of validations could be required.

- 1st level validation: at device level use of diagnostic information to validate the raw measurement information.
- 2nd level of validation: comparison of the synchronized information produced by redundant transmitters, timeliness (temporal consistency of data), and measurement spread are checked.

- 3rd level of validation: comparison of validated measurements resulting from a previous level validation with a measurement information resulting from a process model.

NOTE For each measurement the user has to specify what part of this complete model is requested and the request has to be recorded in the channel configuration documentation.

B.3.7.2.3 Trend of validated measurements

Data file documenting a defined phenomenon (see 6.8.6).

B.3.7.2.4 Event notification (thresholds)

An event notification is asynchronously produced by a phenomenon detection algorithm. It may be characterized by a defined timeliness requirement. Typically it has not be lost. In other words, the correct reception has be acknowledged.

B.3.7.3 Measurement management

B.3.7.3.1 One-shot application triggering

The one-shot application triggering will be used to trigger the execution control of an application function that starts upon the reception of the trigger and then stops. A mechanism is needed which guarantees that, if the one-shot triggering signal is available at the consuming application within a pre-defined interval following generation time is completed with a positive timeliness information, if not it has to be completed with a negative timeliness information.

NOTE The interval should be specified (by the user/system integrator) in terms of the application needs.

B.3.7.3.2 Cyclic application triggering

This service can be used to resynchronize at low frequency (for example,a period of 5 s) a high-frequency pulse train generator (for example, a period of 100 ms, 10 ms, 1 ms). This pulse train will be used for example for the execution control of functions which have to be executed in different devices in synchronized time intervals; a typical case is to acquire redundant measurements from different devices.

B.3.7.3.3 Temporal consistency of data (timeliness)

At system level is required that data originated by the different devices are complemented (in production, transmission, reception) with a timeliness notation able to express the validity period of the data. An example is the reception of measurements or commands.

B.3.7.3.4 Distributed “absolute time” (year, month, day, hour, etc.)

This is composed of two parts:

- distribution of “Absolute time” reference data;
- synchronization of the device counters obtained with two mechanisms:
  - use of distributed synchronized clock (see previous example on train pulses),
  - synchronous setting of the absolute time value repeated at low frequency.

B.3.7.3.5 Distributed “relative time”

As the previous “absolute time” but a “relative time” is distributed (for example, the system starts with a zero time).

B.3.7.3.6 Distributed scheduling of an application

The system requirement is that the total time necessary to execute a distributed application is respected by the chosen distribution policy.

It should be verified if some intermediate synchronization has to be respected, for example, restarting the complete application.

B.3.7.4 Dependability

B.3.7.4.1 Support for redundancy

The support for redundancy is requested at two levels:

- functional level, at the level of application;
- at the level of resource redundancy.

B.3.7.5 Support access rights management function

B.3.7.5.1 Mode of operation management function (extended for access rights management)

This is a system management function aimed to manage the transitions among the several operating modes. In particular, it manages the access rights transfer procedures between the different human or automatic operators. This is to support procedures of intervention in the field devices, in order to realize intervention plans as agreed among the human organizations. Therefore, the device management should cooperate with the system management to achieve the result.

B.3.7.5.2 Access rights management function (qualification check on every access to device functions).

In each operating mode, the devices themselves have to allow operators to perform on them certain pre-defined actions only.

It is obvious, for example, that it is extremely important to disable the write access to the device for configuration, parameterization and tests on demand purposes when the measurement processing functions have to be active (such as in control operating modes).

B.3.8 System requirements for maintenance and technical management

B.3.8.1 General

With regard to requirements for distributed maintenance and technical management applications, some major users requirements are needed for IAM channels to be analysed. These include checking and modifying application parameters, managing access rights, reading the device identification, checking the functional configuration of the provided solution, checking/modifying application parameters, checking the status of the devices and supporting the tests on demand.

These requirements are covering all the device life cycle and therefore they partially regard the end-user and partially the manufacturer, the system integrators, etc.

B.3.8.2 Device identification

Device identification information needed to support maintenance actions is already described in the main part of thisTechnical Specification.

B.3.8.3 Configuration

The maintenance operator or the commissioning operator has to modify or check the functional configuration of the devices. They also need to be sure, in particular after the replacement of a device, that the channel functional configuration is consistent. It is a system level function aimed to check that all the functions implemented in the devices composing the channel are properly configured. As an example, if a filtering function is required in the channel, this function is implemented either in the field device or in the external device but not in both or nowhere.

The required functionality is as follows:

- configuration modification:
  - user-related device/channel configuration modification;
  - manufacturer-related device/channel configuration modification;
- configuration retrieval:
  - user-related device/channel configuration retrieval;
  - manufacturer-related device/channel configuration retrieval.

B.3.8.4 Parameterization

This function regards checking/modifying application parameters, during commissioning or maintenance interventions.

After every change in a device parameter, this function is required to be available at system level in such a way as to help the maintenance/commissioning operator. The operator should be sure that the function parameters are consistent among the devices composing the channel (for example the engineering unit or low-pass filter frequency expected by a higher level device really match with those used in the field device).

Device/channel parameters are divided into the two following categories:

- intrusive parameters: the modification of these parameters interferes with the normal operation of the device/channel;
- non-intrusive parameters: the modification of these parameters does not interfere with the normal operation of the device/channel.

The required functionality is as follows:

- parameterization modification:
  - intrusive device/channel parameterization modification;
  - non-intrusive related device/channel parameterization modification.;
- parameterization retrieval:
  - device/channel parameterization retrieval.

B.3.8.5 To monitor device statuses

Each user, at system level, needs to have a view of the health of the devices according to his needs.

The end-user will need to know which replaceable part he has to substitute; moreover, he needs the information the manufacturer requires to repair the device more quickly and properly (see “condition-based maintenance”).

The repairer will probably require more details to perform his intervention.

The required functionality is as follows:

- device/channel detailed status retrieval:
  - device/channel detailed status information retrieval.

B.3.8.6 To support test on demand

This support, as needed for actuation and measurement remote or local tests (for example calibration), is divided in the two following categories:

- intrusive tests: the execution of these tests interfere with the normal operation of the device/channel;
- non-intrusive tests: the execution of these tests does not interfere with the normal operation of the device/channel.

According to the requested tests, the necessary permission is needed, and the device/channel has to be put in the appropriate operating mode.

The required functionality is as follows:

- tests execution:
  - intrusive device/channel tests execution;
  - non-intrusive related device/channel tests execution;
- tests results retrieval:
  - device/channel tests results retrieval.

B.3.8.7 To support access rights management function

This is a system management function (see control requirements).

B.3.9 Network management

The following functions have to support the cooperative actions of the device and system management for the IAM channels:

- network commissioning support (network configuration/parameterization verification, etc.);
- network management operational support;
- communication monitoring (errors and statistics);
- communication failure management (with or without redundancy);
- timeliness control support.

B.3.10 FB functional requirements conformance testing

Some kind of conformance test has to be done to guarantee which part of the functional requirements is covered by each FB specification and the extension of the coverage. This testing is important because the results will later make easier both the conformance testing between the functional requirements and the market solutions which will implement the standard FBs.

Annex C
(informative)

Relation between IEC 61804 series and IEC 61499-1

C.1 General

IEC 61499-1 specifies a generic standard for distributed FB applications for Industrial Measurement and Control Systems (IPMCS). The following models are within the scope of IEC 61499-1:

- application model;
- device model;
- resource model;
- FB model;
- system model.

The detailed approach of the IEC 61804 series regarding the device and FB model is described in Clause C.2. This should give a short overview about common specifications and differences. The annex is completed by one example of a rich IEC 61804 FB described in terms of IEC 61499-1.

This annex contains all the results regarding the relation of IEC 61499-1 to the IEC 61804 series, which was reached during the requirement specification.

C.2 Relations to the IEC 61499-1 FB model

C.2.1 General

This clause points out differences in the use of the IEC 61499-1 FB model. Details of the IEC 61499-1 model are not given. For more details about IEC 64199, see IEC 61499-1 and IEC 61499-2.

C.2.2 General characteristics

IEC 61499-1 defines a FB model of which the graphical representation is shown in Figure C.1. It consists of the components FB head and FB body. The body carries the data flow (data inputs and data outputs, algorithms, internal data, and, as an addition to IEC 61499-1, the so-called contained data, which are not involved in the data flow, but adjust the algorithms) and the head of the event of flow (event inputs, event outputs and execution control).

These components characterize IEC 61804 FBs in general.

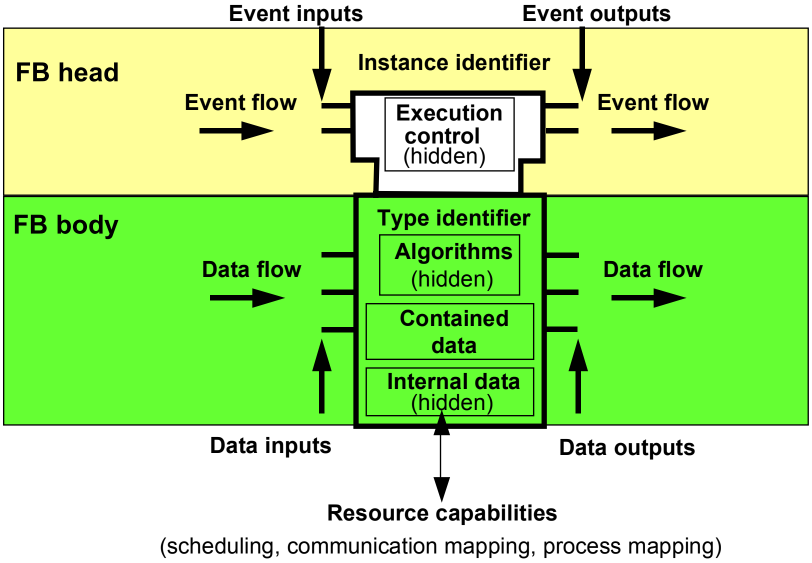


Figure C.1 – Structure of IEC 61499-1 FBs

An IEC 61804 FB type is characterized by a well-defined named event inputs and outputs, named data inputs and outputs, contained data, a certain set and detail of execution control and a certain set and detail of algorithms. The type is a FB interface description with some behaviour behind named data (i.e. algorithms). However, FBs with one event input and output and one algorithm are also within the scope.

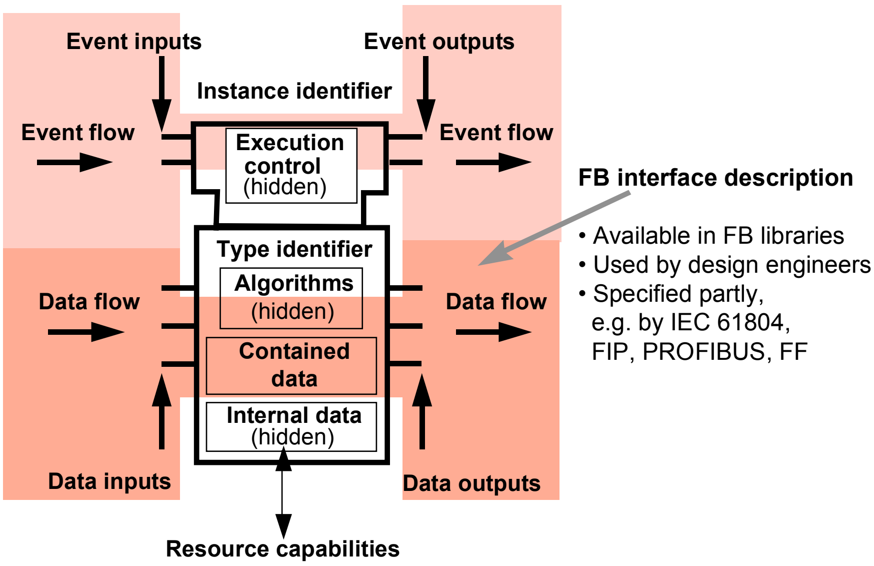


Figure C.2 – Type specific aspects of IEC 61499-1 FBs

Internal data, algorithm and execution control details are outwith the scope of the IEC 61804-2. These aspects are implementation details and hidden from an external view to the FB (see Figure C.3).

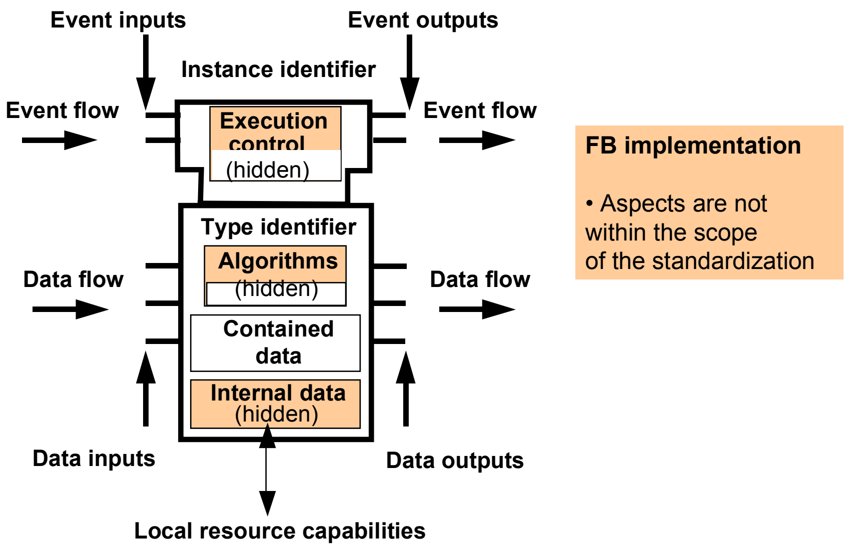


Figure C.3 – Implementation specific aspects

The execution order of FBs in a centralized program is determined by the order of the FB call in the programme, or by the scheduling of the task system of one resource (see Figure C.4). In the distributed environment, in terms of IEC 61499-1, the execution order of FBs is determined by the event flow of the FB heads. Because of the event connection between the execution control head of a FB and the local operating system which is seen as a scheduling function. The head of the FB is the configurable part of the distributed operation system. In other words, the distributed operation system is build by the local operating system of each resource and the connections between FB execution control heads.

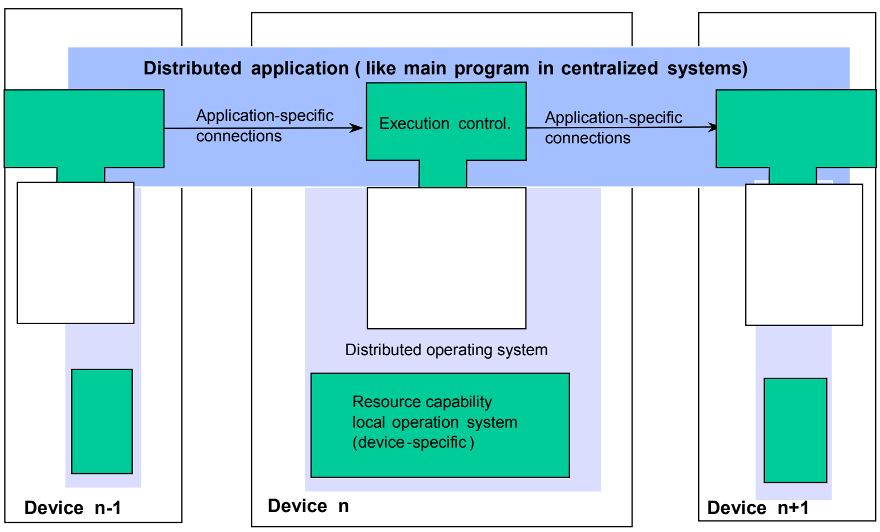


Figure C.4 – Distributed application and distributed operating system

To summarize the use of the IEC 61499 series, Table C.1 focuses on the standard aspects coming from IEC 61499-1, the type-specific aspects and the application-specific aspects which will be defined from the specific application design.

Table C.1 – Aspects overview of IEC 61499-1 FBs

	IEC 61499-1 aspects	Type-specific aspects	Application-specific aspects
Head	Event flow (edges)	Execution control Event inputs/outputs	Connections between events
Body	Data flow	Data inputs/outputs Part of algorithms contained data	Connections between data (data connections)

C.2.3 FB type specifications

The process FBs include

- standardized data structures,
- standardized semantic meaning,
- common behaviour associated with standardized data,
- standardized definitions for basic input, output, event inputs event outputs, and control functions.

C.2.4 Characteristics of FB instances

For the process industry, the instance name of FBs will be defined as the block tag. Tags are unambiguous within the scope of a system at one plant site.

C.2.5 Mapping of IEC 61499-1 FB elements to IEC 61804

FBs as used in the process industry may interpret the characteristics defined in IEC 61499-1 as followed (see Figure C.5).

- The events, execution control together with the scheduling function of the local operating system are used to trigger different algorithms in the FB body for example init, execution of the normal operation, execution of the abnormal operation (see C.2.6).
- The data flow between the FBs is carried out by variable records composed of variable value and its status, i.e. there is a difference between the use of variables in the data flow or by remote non-FB devices.

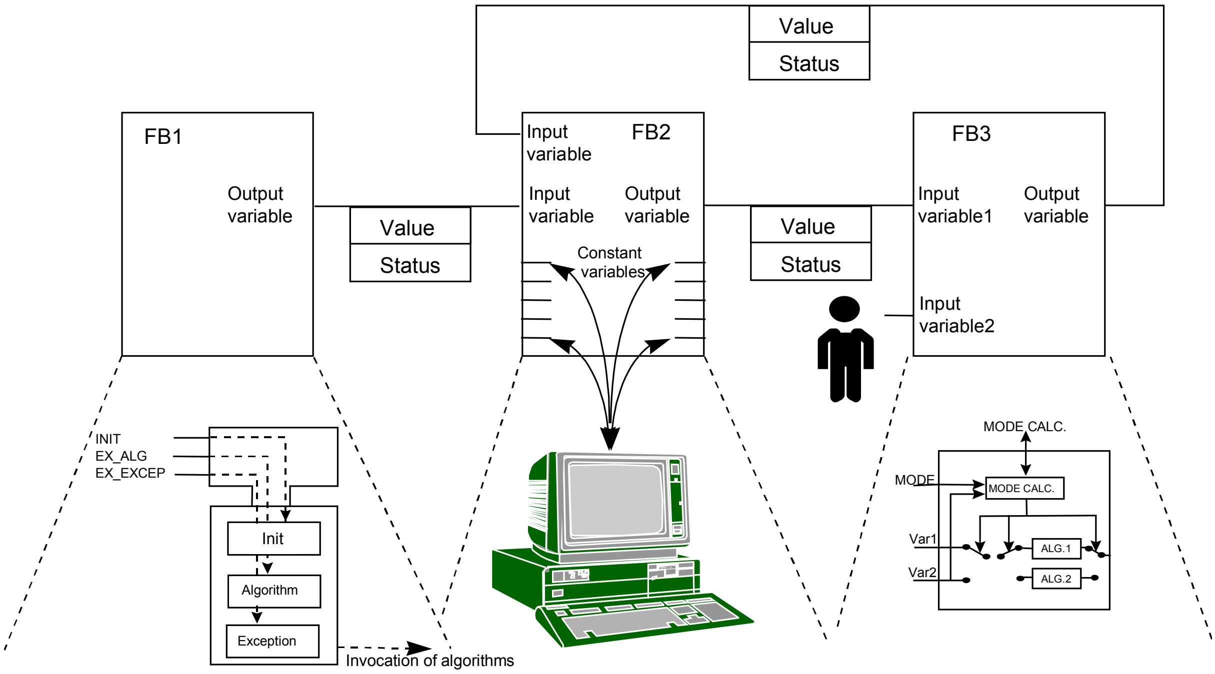


Figure C.5 – Basic concepts of process control FBs

- An additional set of contained data, which may be mapped to corresponding contained variables to interact with maintenance, HMI, monitoring and other applications.

  A contained variable is a variable whose value is configured, set by an operator, higher level device, or calculated. It cannot be linked to another FB input or output and therefore may not contain status. Based on the class of a block, additional variable may be supported in a consistent fashion.

- There is a special FB algorithm in each block, which controls the information calculation within the other algorithms of the FB. This algorithm is known as “modes of operation”.

- The number, names, data types and order of variables in the FB are defined.

C.2.6 FB algorithms

The FB consists of data inputs, data outputs and algorithms responsible for normal operation, i.e. the operation visible in the FRD (see Annex A). These algorithms are for example the PID, linearization or mathematical equation with all their associated variables and parameters. The normal operation of a FB is carried out under positive conditions of the process and automation devices during stable operation (operating point). The initialization, the warm or cold start as well as more advanced operations like going in a safe position are additional functions of the process control FBs. These operations are fixed parts of the FBs, which are default linked to the associated events. These operations are known as abnormal operation or exception handling.

The process engineer designs the normal operation in the FRD schemes, while the I&C engineer add the abnormal operation to come to a full FB configuration scheme (see Annex A).

FB algorithms are structured as shown in Figure C.6.

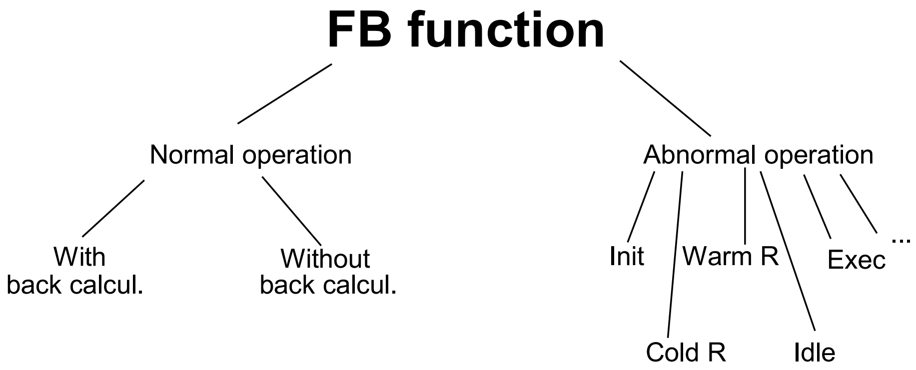


Figure C.6 – Functional components in process control FBs

A FB consists of the main algorithms performing the normal operation (one or more than one) and the algorithms for abnormal operation. The active event chooses which algorithm has to be executed. During normal operation most of the time the main algorithm are carried out. During commissioning, maintenance or critical process states algorithms of the abnormal operation are carried out. An example is shown in graphical representation of IEC 61499-1 (see Figure C.7).

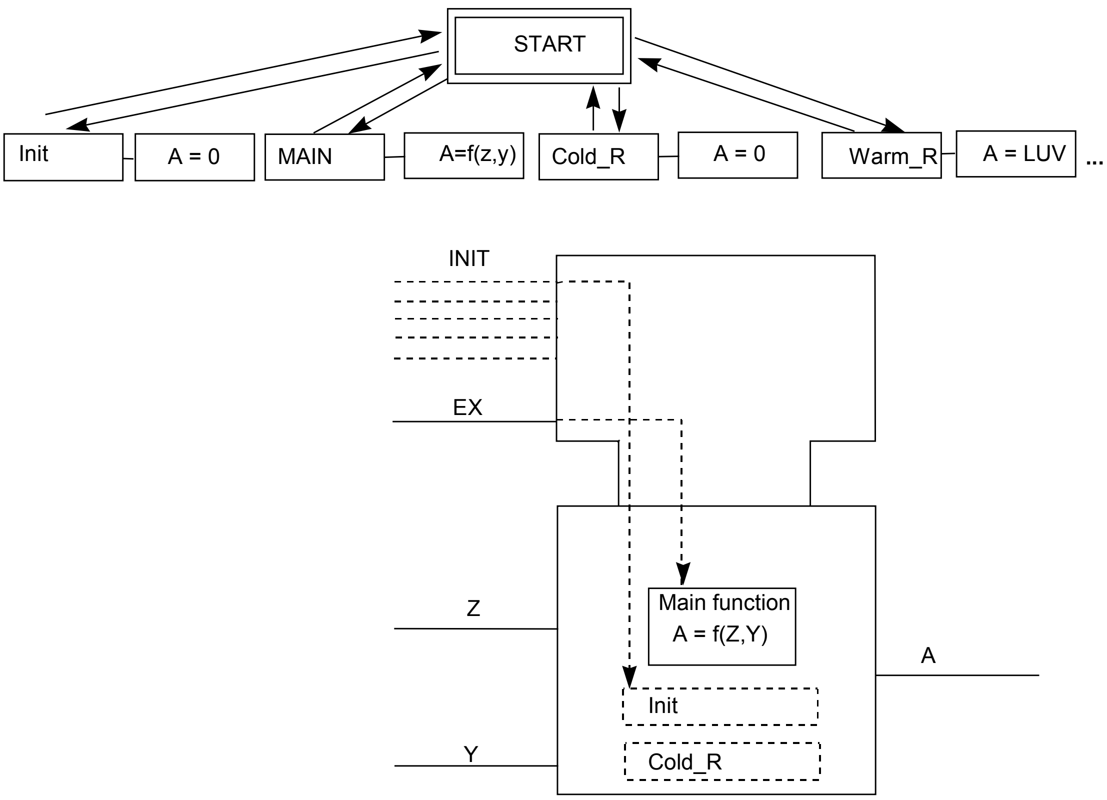


Figure C.7 – IEC 61499-1 graphical representation of a process control FB (example)

Figure C.7 shows the normal and abnormal operation algorithms which are triggered by the associated events. The combination of normal and abnormal operations is valid for AB and EFBs. Of course, the degree of implementation of the abnormal operations depends on the application area. Therefore, conformance classes should be defined.

FBs should include standardized functional definitions for initialization and restart for each block. Behaviour should be defined for the following operating circumstances:

- new device;
- cold restart (extended power failure);
- warm restart (short power failure);
- return from device fail-safe.

Figure C.8 gives the full structure of an EFB.

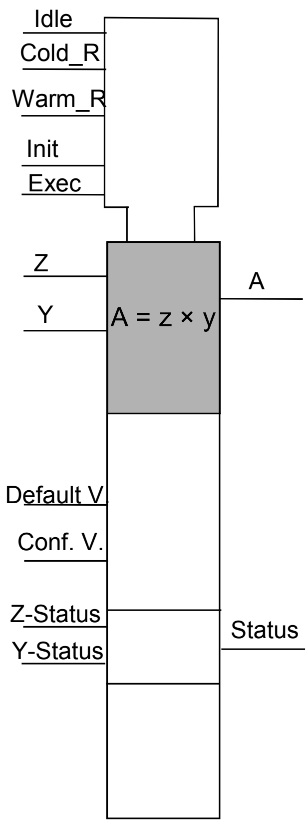


Figure C.8 – Full structure of an EFB (example)

C.2.7 Relationship between IEC 61499-1, IEC 61804 and other standardization activities

The purpose of a FB differs according to its application. The FB classification aims at structuring them in a hierarchy. During the specification, there should be an inheritance of defined functionality along one path of the tree. For classification, an object-oriented approach should be used. The classification reflects several viewpoints, for example, the existence of interfaces to the application periphery (to communication, process, operation system or not), the manipulation of either the data or the event flow or both and the functional purpose regarding the process. Each element of the hierarchy is an entity in terms of the specification. The properties of the entities are specified by attributes. There is a specification line from the IEC 61499-1 type specification without special variables and attributes, multi-purpose FBs with a certain set of attributes to specific dedicated FBs with their specialized set of variables and attributes (see Figure C.9).

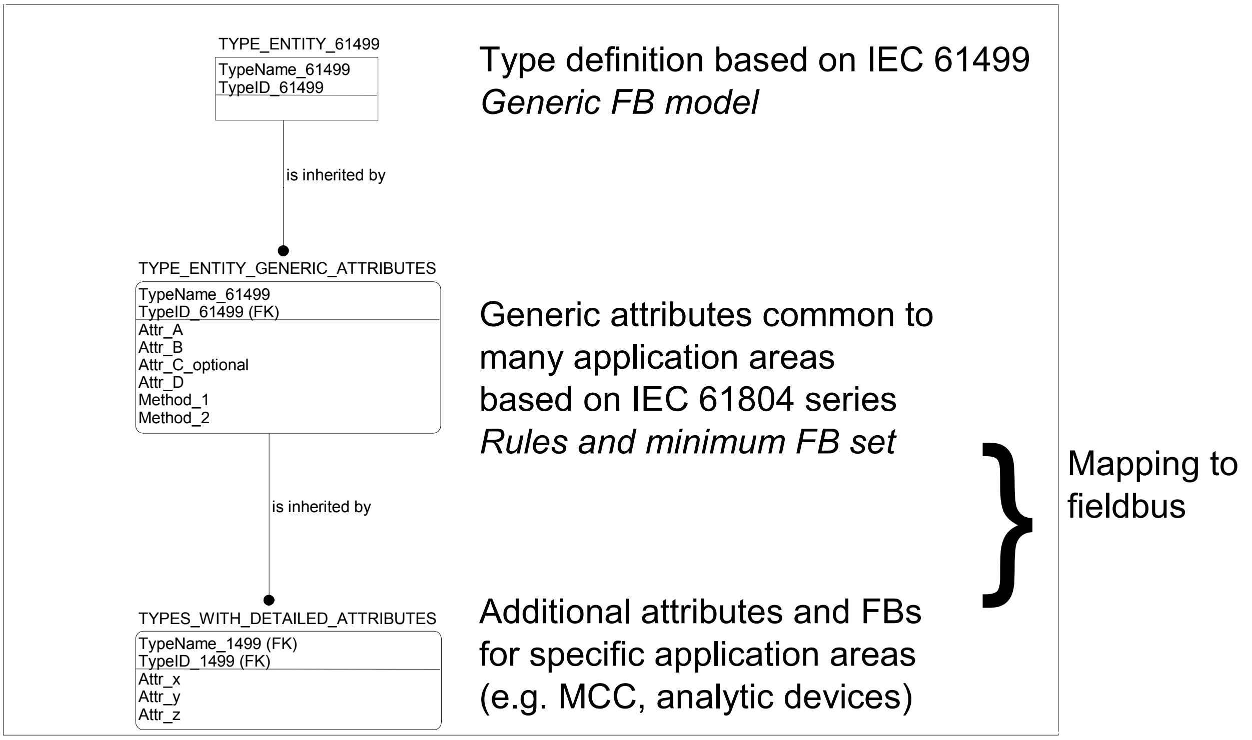


Figure C.9 – Relationship between IEC 61499-1, IEC 61804 series and other standardization activities

C.3 Mapping of IEC 61499-1 device and resource model to IEC 61804-1

C.3.1 General

The following subclauses describe the IEC 61804 device model as a result of the requirement specification. This clause closes with a reference overview between the device models of both publications.

C.3.2 IEC 61804-1 device model components overview

In a real device or resource, there are only data and function (program) code as well as interfaces to the communication, process and HMI. Only via the communication system, by a configuration or design tool, these data and program code are seen as FBs, management agents or data connections. The specification of the device model is an external view to the device implementation. The device model describes details of the overall system architecture. The device model is an abstraction of the real device.

The basic components of a device performing FB applications are (see Figure C.5):

- FBs composed of data and event inputs and outputs, internal data, contained data and algorithms should have the following general requirements:
  - FB environment

    FBs are embedded in an operating environment shown as FB environment. The FB environment performs special process control functions which are *a priori* available application portions without any programming design activities, for example trending. These FB environment functions are used by external applications independent of, i.e. parallel to, the FB application.

– AME

Many process control devices need adoption to the used hardware configuration from the functional point of view. For instance, the available FBs in a modular device have to be configured or the model version with or without an additional input for measurement compensation. This causes to add or delete FBs. These functions will be initiated by the so-called AME, i.e. addition functions in a device modifying their application software. IEC 61804 series does not define the AME. This should be done within the framework of the IEC 61158 series.

– Communication

The data transfer between FBs, the interactions between FBs/FB environment and maintenance, HMI monitoring and the interaction between FB environment/AME and configuration functions have to be supported by the communication services and protocols. The specification of the FBs and the FB environment is independent of the communication system. A mapping sublayer between FB environment and communication system will adapt the application and the communication.

Parameters, blocks, objects and functions in the FB application process and the system and network management application process should map efficiently to the underlying communications protocols and services. Communications services and protocols used with the FB application process and the system and network management application process should be specifically designed for use with distributed application processes and should provide the services required by those applications.

There should be three different types of communication requirements:

- Time-critical communications

  Communications services and protocols used with the FB application process should support the unique requirements of time critical communications, including

  – deterministic transfers,

  – spatial consistency of transfers,

  – temporal consistency of transfers.

- Demand communications

  Communications services and protocols used with the FB application process should support the unique requirements of time-available communications, including segmented transfers of large data blocks. Communications services and protocols used with the FB application process should also support the unique requirements of “report by exception” communications.

- Event communications

  Communications services and protocols used with the FB application process should support the unique requirements for communicating events, and in particular be designed to

  – minimize loading during quiescent periods;

  – prevent overloading during high-activity periods.

  – Resources

A resource is considered to be a logical subdivision within the software (and possibly hardware) structure of a device. Resources have independent control of their operation. The definition of a resource may be modified without affecting other resources within a device. A resource accepts and processes data and/or events from the process and/or communication interfaces and returns data and/or events to the process and/or communication interfaces, as specified by the applications utilizing the resource. An interoperable network view of applications is provided through device resources. Each resource specifies the network visible aspects of one or more local applications (or parts of distributed applications).

C.3.3 Classifications of FBs

C.3.3.1 Overview

The components of the FB application architecture that may be accessed through its associated resources are described below (see Figure C.6). They are

a) blocks
   - resource blocks;
   - technology blocks;
   - FBs;
b) alert block;
c) trend block;
d) view blocks.

C.3.3.2 Resource block

The characteristics of the physical subcomponent associated with a resource may be described by a set of resource block contained variables. The resource block may also contain variables that are common to FBs and technology blocks, for example set fail-safe. These variables are defined in the resource block.

C.3.3.3 Technology block

Technology blocks insulate FBs from the specifics of I/O devices, such as sensors, actuators, and switches. Technology blocks control access to I/O devices through a device-independent interface defined for use by FBs. Technology blocks also perform functions, such as calibration and linearization, on I/O data to convert them to a device-independent representation. Their interface to FBs is defined as one or more implementation-independent I/O channels.

C.3.3.4 FB

The FB is the primary means of defining monitoring and control in a FB application. FBs represent the basic automation functions performed by an application that is as independent as possible of the specifics of I/O devices and the network. Each FB processes input variable and technology block input according to a specified algorithm and an internal set of contained variables. They produce output variable and output to technology blocks.

Based on the processing algorithm, a desired monitoring, calculation or control function may be provided. The results from FB execution may be reflected in contained variable for operation or diagnostic information. In addition, processing results may be reflected in the output to a technology block or to one or more output variables that may be linked to other FBs.

C.3.3.5 View block

The FB environment includes data structure definitions to allow access to related block parameters as a group, called view blocks. View blocks facilitate fast operator display response when viewing FB data.

For each FB type, view blocks be defined for each of the following block parameter groupings:

- operations dynamic parameters;
- operations static parameters;
- all dynamic parameters;
- other static parameters.

C.3.3.6 Trend block

FBs should include data structure definitions to allow access to multiple time-stamped samples of a single block parameter as a group, called trend blocks. Trend blocks eliminate the communications and system processor overhead required for scanning parameters at a fast rate for trending.

Trend object definitions should include the definition of standard sample types and their associated sampling functions (for example, spot sample, integrated average, minimum, maximum, etc.).

C.3.3.7 Alert block

The FB environment should include data structure definitions and associated resource and FB functions to allow the controlled transfer of alarm and event information within the system, called alert blocks. Alert blocks predictably and efficiently route alarms and events to a selected destination (or destinations) within the system.

Alert block definitions include the definition of standard exchange protocols for initiating, sending, and confirming alert block reports, acknowledging alerts, interpreting standard and custom reason codes, and configuring alert functions such as alert key, alert priority, alert auto-acknowledge, etc.

Alerts are used by resource, technology, and FBs to communicate notification messages when alarms are detected. An alarm is the detection of a block leaving a particular state and when it returns back to that state. The time at which the alarm state was detected is included as a time stamp in the alert message. Also, the priority is included to indicate whether this is an advisory or critical alarm.

C.4 Mapping of IEC 61499-1 management FBs

Management is within the scope of the IEC 61158 series and not reflected in this Technical Specification.

C.5 IEC 61499-1 textual language

The IEC 61499-1 textual language is used to describe examples of IEC 61804 FBs to illustrate the FB definitions.

C.6 References between IEC 61499-1 and IEC 61804 model components

Table C.2 summarizes the above clauses. The references are not formal ones; they should visualize the common specifications and differences in both concepts.

Table C.2 – Reference between IEC 61499-1 and IEC 61804 components

IEC 61499-1	IEC 61804	Comments
System model	System overview	The IEC 61804 system overview shown in Figure 1 reflects that there are non-FB applications interaction with the FB application. IEC 61499-1 knows FB application only
Application model	FB application	The IEC 61804 FB application shown in Figure 1 reflects that there are non-FB applications interaction with the FB application. IEC 61499-1 knows FB application only
Device model	Device	No difference
Resource model	Resource	No difference
Scheduling functions (partly)	FB environment	Today IPMCS process control common functionalities are implicitly available, such as alert handling, use of specific quality of services (QoS) of the communication system, access right management, and others. During the preparation of IEC 61804-2 it should be decided which of these functionalities will become explicit (for example, become FBs) and which remain hidden. IEC 61499-1 specifies all explicit (nothing is magic)
FB model	FB	No difference except that event connections as an explicit part of the application design is not mandatory
-	Resource block	Not reflected in IEC 61499-1
FB model	FB	No difference except that event connections as an explicit part of the application design is not mandatory
Service interface FB to the process	Technology block	No difference
Special collection FB which provides the inputs to service interface FB to the communication	View block	IEC 61804 has an implicit mapping to communication
Special collection FB which provides the inputs to service interface FB to the communication	Alert block	IEC 61804 has an implicit mapping to communication
Special collection FB which provides the inputs to service interface FB to the communication	Trend block	IEC 61804 has an implicit mapping to communication
Management FB	-	This is part of IEC 61158 system management

Annex D
(informative)

Mapping of an analogue input FB to IEC 61499

D.1 Overview

This annex proposes, as an example, the mapping of an analogue input block on the model provided by IEC 61499-1.

D.2 Introduction

D.2.1 General

The mapping uses parts of the IEC/PAS 61499-1 models to describe a process control FB (FB) as it is specified by ISP, PROFIBUS and FF. In particular, the underlying example is the analogue input FB (AI FB) as specified by PROFIBUS-PA.

IEC/PAS 61499-1 contains a textual syntax of the models in Annex B, which is used to describe the AI FB in addition to the graphical representation. This is the starting point for the use of the IEC 61499 models for the following reasons:

- to apply the IEC 61499 series for process control FB systems, i.e. gain experience with the IEC 61499 models;
- to validate (test the suitability) of the IEC 61499 models for process control;
- to find problems with the textual representations;
- to present the IEC 61804-1 requirements to IEC 61499 series and other users;
- to investigate if the IEC 61499 textual representation is suitable for a formalized description of the IEC 61804 series output;
- to help clarify misunderstandings between IEC 61804 series and IEC 61499 series.

D.2.2 Modifications of the mapping version 1.1

Version 1.1 (Date:1999, July 7th) lists the following modifications:

a) *Modification: update new version of mapping to IEC 61499-1 with separate event to control the MODE algorithm;*

b) *Modification: use only AI FB mapping with MODE EVENT, modify INPUT and OUTPUT variables and events;*

c) *Integration of parameter description.*

D.3 Analogue input block

D.3.1 Analogue input block overview

Analogue input blocks represent transmitters. The parameters are shown in Figure D.1.

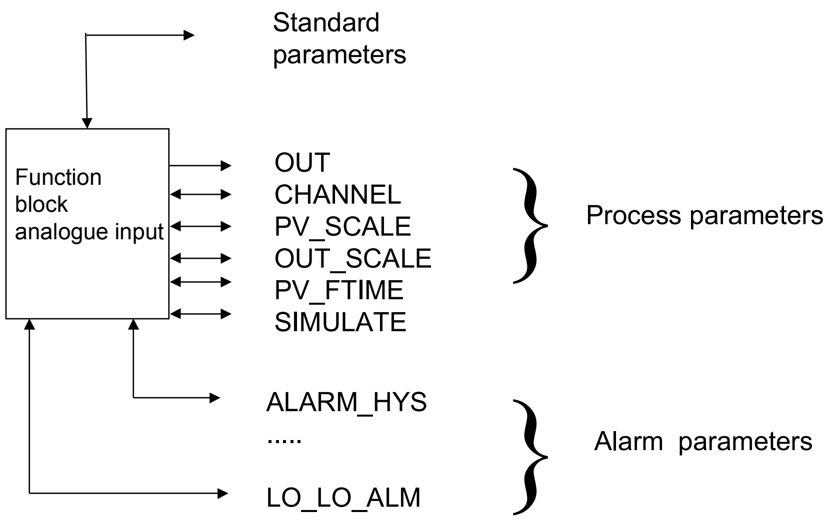


Figure D.1 – Summary of the parameter of analogue input blocks

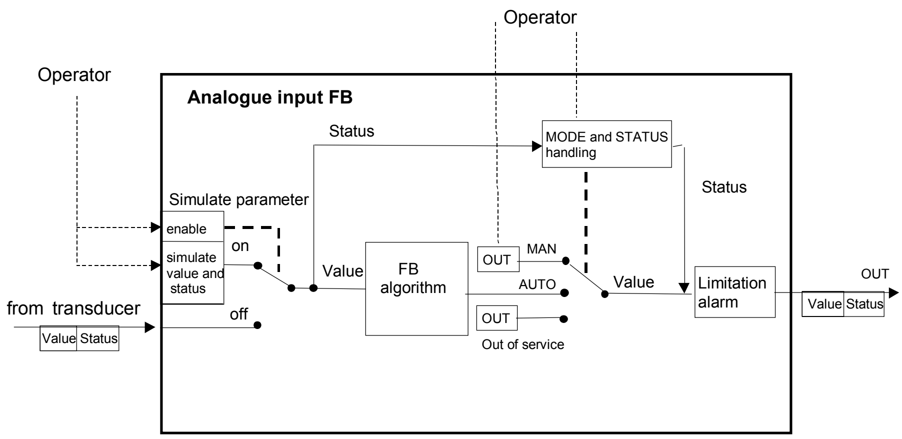


Figure D.2 – Simulation, mode and status diagram of analogue input block

The structure of the AI with simulation, mode and status is shown in Figure D.2.

Figure D.3 presents a summary of the inputs and outputs of the mode and status generation.

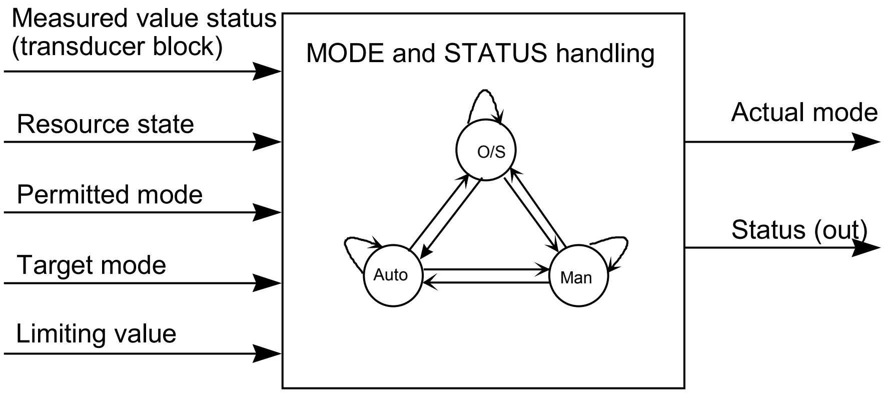


Figure D.3 – Conditions of mode and status generation

The measurement value/status delivered by the transducer block to the FB across the channel is one of the input of the MODE calculation. The resource state describing the health of the device in general is not explicitly defined in the profile, i.e. it is device-specific and not presented in a parameter, but the resource state has to distinguish between *ok* and *not ok* as a minimum. Permitted, actual and normal mode are attributes from the FB-Parameter MODE_BLK. The target mode is set by the operator and the permitted mode from the block designer. Also the high and low limiting value (HI_LIM, HI_HI_LIM, LO_LIM, LO_LO_LIM), regarding the output value, influence the status of the output.

Actual mode is an attribute of the FB-Parameter MODE_BLK and the result of the mode calculation. The status (OUT) is coupled with the out parameter (DS 33) of the block.

D.3.2 AI_State machine

The permitted modes of the AI are Out of Service (O/S), Man (Manual) and Auto (Automatic). The possible transitions are illustrated in Figure D.4.

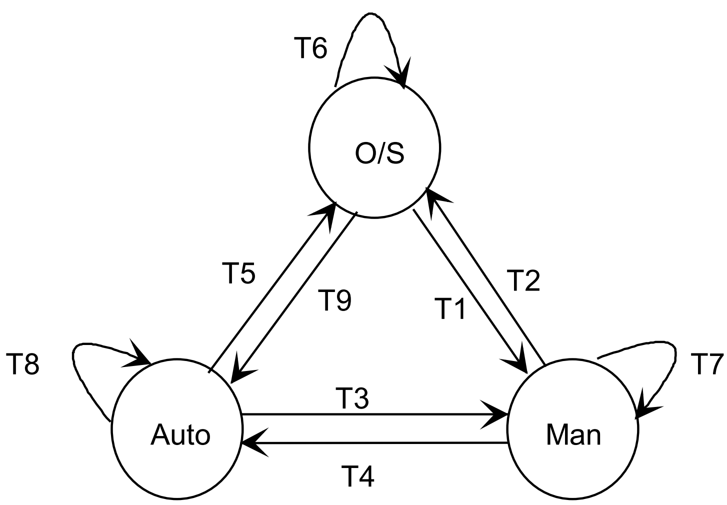


Figure D.4 – State machine of the analogue input block

D.3.3 Conditions in which the actual mode is calculated and the target mode is changed

Table D.1 contains on the left all the conditions which demand a mode change from the actual mode (last execution) to the new actual and target mode of the AI-FB. The results of the calculation are illustrated on the right .

In Table D.1, the first column contains the number of the transition of the state machine. A general condition is that permitted modes are O/S (state after a device or block configuration error), Man (OUT value provided by the operator), Auto (OUT value provided by the device).

Table D.1 – Conditions and results of the actual mode calculation

Conditions					Results	
Transition	**Target mode (operator)**	**Actual mode (previous execution)**	**Resource state**	**Status (transducer input)**	**Target mode (calculated)**	**Actual mode (calculated)**
T2,T5,T6	*	*	<>ok	*	O/S	O/S
T2,T5,T6	O/S	*	*	*	O/S	O/S
T1	Man	O/S	ok	*	Unchanged	Man
T1	Auto	O/S	ok	*	Unchanged	Man
T4	Auto	Man	ok	Good (NC)	Unchanged	Auto
T7	Auto	Man	ok	<>Good (NC)	Unchanged	Man
T3	Auto	Auto	ok	<>Good (NC)	Unchanged	Man
* No influence						

D.3.4 Conditions on which the output status are generated

Table D.2 shows which conditions influence the status of the output parameter. The conditions are illustrated on the left and the results of the calculation on the right.

Table D.2 – Conditions and results of the status calculation from the output parameter

Conditions				Result status (out)
Actual mode	**Simulation**	**Output limitation**	**Status (transducer input)**	
O/S	*	*	*	Bad-out of service
Man	*	*	*	Uncertain – ok, high limit, low limit
Auto	Inactive	*	Good (NC)- *	Good (NC) – ok
Auto	Inactive	*	High limit	High limit
Auto	Inactive	*	Low limit	Low limit
Auto	Active	*	*	From simulate parameter
*	*	High	*	High limit
*	*	Low	*	Low limit
* No influence.				

Table D.3 presents an overview of all additional (related to the general definitions) parameters and their attributes of analogue input blocks.

Table D.3 – Parameter attributes for the analogue input block

	Parameter name	Object type	Data type	Store	Size	Access	Para-meter use/type of transport	Default values	Mandatory optional (Class A, B)
... Standard parameter									
Additional parameter for analogue input block									
10	OUT	Record	DS-33	D	5	r	O/cyc	Measured of the variable state	m (A,B)
11	PV_SCALE	Record	DS-36	S	11	r/w	C/a	-	m (A,B)
12	OUT_SCALE	Record	DS-36	S	11	r/w	C/a	-	m (B)
14	CHANNEL	Simple	Unsigned16	S	2	r,w	C/a	-	m (B)
16	PV_FTIME	Simple	Float	N	4	r,w	C/a	0	m (A,B)
19	ALARM_HYS	Simple	Float	S	4	r,w	C/a	0.5 % of range	m (A,B)
21	HI_HI_LIM	Simple	Float	S	4	r,w	C/a	Max. value	m (A,B)
23	HI_LIM	Simple	Float	S	4	r,w	C/a	Max. value	m (A,B)
25	LO_LIM	Simple	Float	S	4	r,w	C/a	Min. value	m (A,B)
27	LO_LO_LIM	Simple	Float	S	4	r,w	C/a	Min. value	m (A,B)
30	HI_HI_ALM	Record	DS-39	D	14	r	C/a	0	m (A,B)
31	HI_ALM	Record	DS-39	D	14	r	C/a	0	m (A,B)
32	LO_ALM	Record	DS-39	D	14	r	C/a	0	m (A,B)
33	LO_LO_ALM	Record	DS-39	D	14	r	C/a	0	m (A,B)
34	SIMULATE	Record	DS-50	N	6	r,w	C/a	Disable	m (B)

D.3.5 Analogue input block parameter descriptions

Tables D.4 and D.5 contain the verbal descriptions of the parameters of Table D.3. Figure D.5 describes graphically the relation between the limit-checking parameters .

Table D.4 – Process parameter description

Parameter	Description
CHANNEL	Reference to the active transducer block which provides the measurement value to the FB, The number is the same as the one of this transducer block in the Composite_Directory_Entry in the directory of the device management
OUT	Process variable The FB parameter OUT contains the value of the current process variable in a vendor-specific or configuration-adjusted engineering unit and the relevant state
PV_SCALE	Conversion of the process variable into per cent using the high- and low-scale values, engineering units code and number of digits to the right of the decimal point. The engineering unit of PV_SCALE has to be the same as that of the related transducer block
OUT_SCALE	Scale of the process variable The FB parameter OUT_SCALE contains the values of the lower limit and upper limit effective range, the code number of the engineering unit of process variable and the number of digits on the the right-hand side of the decimal point
PV_FTIME	Filter time of the process variable The FB parameter PV_FTIME contains the time constant for the rise time of the FB output up to a value of 63,21 % resulting from a jump on the input. The engineering unit of the parameter is second

Table D.5 – Alarm parameter description

Parameter	Description
ALARM_HYS	Hysteresis Within the scope of the PROFIBUS-PA specification for transmitters, there are functions for the monitoring of limit violation (off-limit conditions) of adjustable limits. If the value of one process variable is just the same as the value of a limit and the variable fluctuates around the limit, many limit violations will occur. This triggers off a lot of messages; so it has to be possible to trigger messages only after crossing an adjustable hysteresis. The sensitivity of triggering of the alarm messages is adjustable. The value of the hysteresis is fixed in ALARM_HYS and is the same for the parameters HI_HI_LIM, HI_LIM, LO_LIM and LO_LO_LIM. The hysteresis is expressed as value of the span which is equal above and below the limit, in the engineering unit of xx_LIM
HI_ALM	State of the upper limit of warnings This parameter contains the state of the upper limit of a warning and the related time stamp. The time stamp expresses the time the measured variable has been equal to, or higher than, the upper limit of the warning. Devices without a clock use the beginning of the PROFIBUS-PA time (1st January 1992) as a time stamp*
HI_HI_ALM	State of the upper limit of alarms This parameter contains the state of the upper limit of an alarm and the related time stamp. The time stamp expresses the time the measured variable has been equal to, or higher than, the upper limit of the alarm. Devices without a clock use the beginning of the PROFIBUS-PA time (1st January 1992) as a time stamp*
HI_HI_LIM	Value for upper limit of alarms Upper limit value for alarms with engineering unit. If the measured variable is equal to, or higher than, the upper limit value, the state bit in the state byte of OUT and in the FB parameter ALARM_SUM have to be changed to 1
HI_LIM	Value for upper limit of warnings Upper limit value for warnings with engineering unit. If the measured variable is equal to, or higher than, the upper limit value, the state bit in the state byte of OUT and in the FB parameter ALARM_SUM have to be changed to 1
LO_ALM	State of the lower limit of warnings This parameter contains the state of the lower limit of a warning and the related time stamp. The time stamp expresses the time at which the measured variable has been equal to, or higher than, the lower limit of the warning. Devices without a clock use the beginning of the PROFIBUS-PA time (1st January 1992) as a time stamp*
LO_LIM	Value for lower limit of warnings Lower limit value for warnings with engineering unit. If the measured variable is equal to, or lower than, the lower limit value, the state bit in the state byte of OUT and in the FB parameter ALARM_SUM have to be changed to 1
LO_LO_ALM	State of the lower limit of alarms This parameter contains the state of the lower limit of an alarm and the related time stamp. The time stamp expresses the time at which the measured variable has been equal to, or higher than, the lower limit of the alarm. Devices without a clock use the beginning of the PROFIBUS-PA time (1st January 1992) as a time stamp*
LO_LO_LIM	Value for the lower limit of alarms Lower limit value for alarms with engineering unit. If the measured variable is equal to, or lower than, the lower limit value, the state bit in the state byte of OUT and in the FB parameter ALARM_SUM have to be changed to 1
*See ALARM_FLOAT_STRUCTURE.	

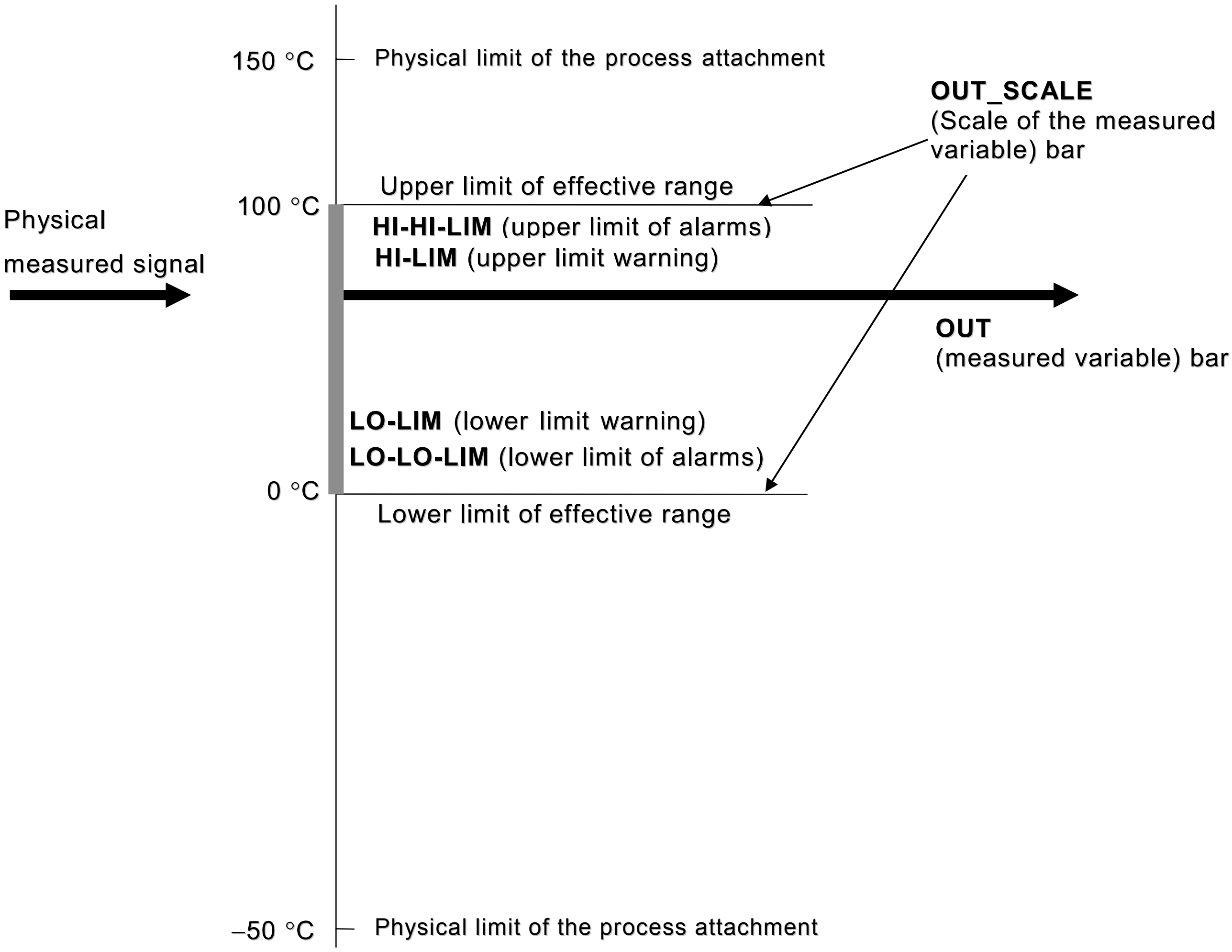


Figure D.5 – Example of the analogue input block parameter

Table D.6 – Simulation

Parameter	Description
Simulate	For commissioning and test purposes the input value from the transducer block in the analogue input block AI-FB can be modified. This means that the transducer and AI-FB will be disconnected

D.4 Mapping

D.4.1 Analogue input block specification in terms of IEC 61499 graphical representation with MODE-Event

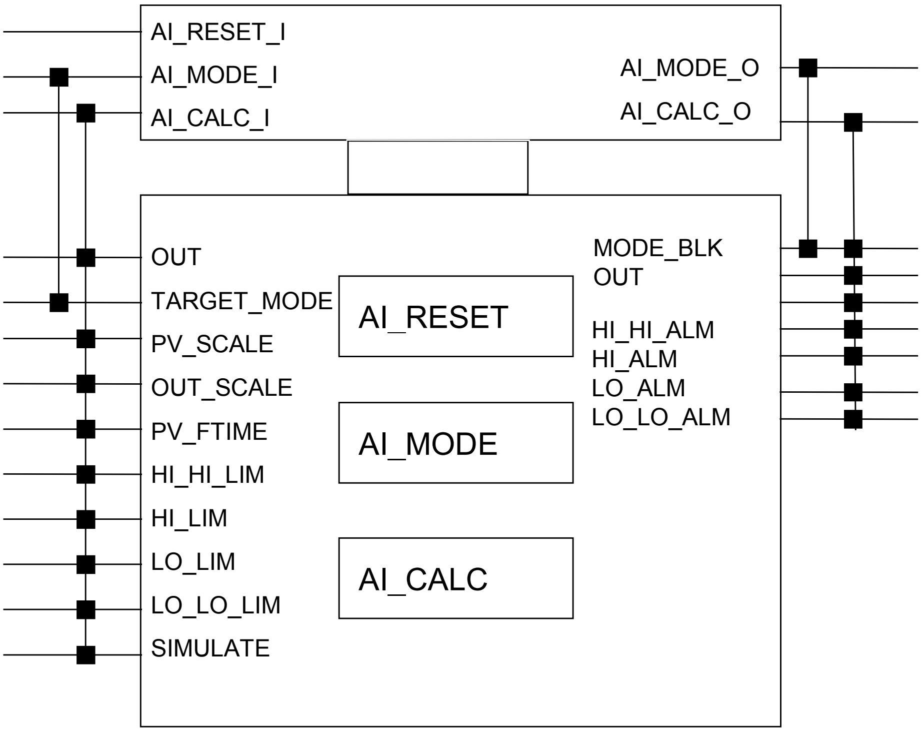


Figure D.6 – AI FB graphical representation

In Figure D.6, it is evident that the so-called contained parameters of a FB is used as INPUT only. The FB does not change any of the contained parameters; therefore, they do not update before the output event generation. The AI_RESET_I does not force the FB to update the INPUT Variables from the external data connections. This AI_RESET function is a reset to the default values.

D.4.2 Analogue input block ECC specification in terms of IEC 61499

The AI FB has four states, the START, RESET, MODE and MAIN (see Figure D.7). The idle state is START. After one event occurs, the transition fires to the related state executing the connected algorithm. The output event is generated after the algorithm is finished. Then the state turns back to START without any addition event (TRUE transitions).

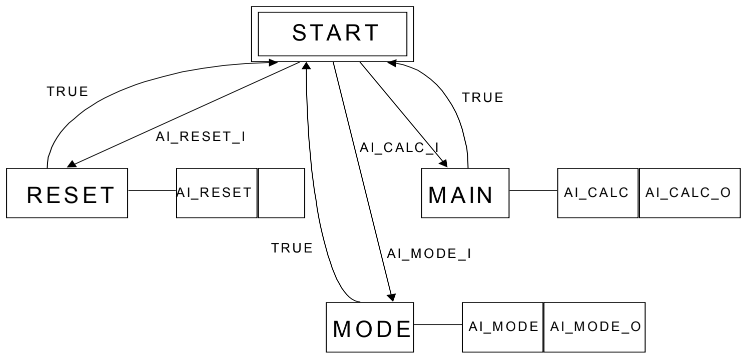


Figure D.7 – AI FB ECC

D.4.3 Analogue input block specification in terms of IEC 61499 textual representation

```
(* #define STATUS constants for example BAD_OUT_OF_SERVICE, NC_GOOD, ... *)
(* #define MODE constants for example OUT_OF_SERVICE, MAN, AUTO *)
(* declare data structures DS_33, DS_37, .... *)


FUNCTION_BLOCK ANALOGUE_INPUT

EVENT_INPUT
        AI_RESET_I;

        AI_MODE_I WITH      TARGET_MODE ;


        AI_CALC_I WITH      OUT
                    PV_SCALE ,
                            OUT_SCALE ,
                            PV_FTIME ,
                            HI_HI_LIM ,
                            HI_LIM ,
                            LO_LIM ,
                            LO_LO_LIM,
                            SIMULATE;
END_EVENT

EVENT_OUTPUT
        AI_MODE_O WITH      MODE_BLK ,

        AI_CALC_O WITH      MODE_BLK ,
                            OUT ,
                            HI_HI_ALM ,
                            HI_ALM ,
                            LO_ALM ,
                            LO_LO_ALM ,
END_EVENT
```

```
EC_STATES

        START
        RESET :        AI_RESET;
        MODE :         AI_MODE; AI_MODE -> AI_MODE_O;
        MAIN :         AI_MODE, AI_CALC -> AI_CALC_O;

                       (* Is it necessary that every ec_action generate an output event? *)

END_STATES

EC_TRANSITION

        START TO RESET    := AI_RESET_I;
        START TO MAIN     := AI_CALC_I;
        START TO MODE     := AI_MODE_I;
        RESET TO START    := TRUE;
        MODE TO START     := TRUE;
        MAIN TO START     := TRUE;

END_TRANSITION


(*  ---------------------------------------------------------------------------------------------------------------
*)


VAR_OUTPUT

        MODE_BLK :        DS_37;

        OUT :             DS_33;

        HI_HI_ALM :       DS_39;
        HI_ALM :          DS_39;
        LO_ALM :          DS_39;
        LO_LO_ALM :       DS_39;

END_VAR

VAR_IN                                      (* INPUT variables are read/write *)

        TARGET_MODE       Unsigned8;

        OUT :             DS_33;            (* for MANUAL MODE *)

        PV_SCALE :        DS_36;
        OUT_SCALE :       DS_36;
        PV_FTIME :        FLOAT;
        HI_HI_LIM :       FLOAT;
        HI_LIM :          FLOAT;
        LO_LIM :          FLOAT;
        LO_LO_LIM :       FLOAT;

        SIMULATION :      DS_50;


END_VAR
```

```
VAR
        TB_MEASURED_VALUE : DS_33;              (* input value from the transducer FB *)
        PV: DS_33;
        TEMP : FLOAT;

END_VAR


(*    ----------------------------------------------------------------------------------------------------------------
*)
*NOTE   The internals of the algorithms are described with the IEC 61131-3 structured text.*

ALGORITHM AI_RESET_IN_ST

        OUT :=                          0;

        PV_SCALE.EU0 :=                 0;
        PV_SCALE.EU100 :=               100;
        PV_SCALE.EU :=                  35;      (* code for ... *)
        PV_SCALE.DP :=                  2; (* valid decimal point *)

        OUT_SCALE.EU0 :=                0;
        OUT_SCALE.EU100 :=                      100;
        OUT_SCALE.EU :=                 35;      (* code for ... *)
        OUT_SCALE.DP :=                 2; (* valid decimal point *)

        PV_FTIME :=                     1;

        HI_HI_LIM :=                    OUT_SCALE.EU100   ;
        HI_LIM :=                       OUT_SCALE.EU100   ;
        LO_LIM :=                       OUT_SCALE.EU0       ;
        LO_LO_LIM :=                            OUT_SCALE.EU0       ;


END_ALGORITHM
(*    ----------------------------------------------------------------------------------------------------------------
*)

ALGORITHM AI_MODE_IN_ST

(* calculation of the actual MODE *)

        IF RESOURCE_STATE <> OK
                THEN MODE_BLK.ACTUAL = OUT_OF_SERVICE;
        END_IF

        IF TARGET_MODE = OUT_OF_SERVICE
                THEN MODE_BLK.ACTUAL = OUT_OF_SERVICE;
        END_IF

        IF TARGET_MODE = MAN     & MODE_BLK.ACTUAL = OUT_OF_SERVICE
                                 & RESOURCE_STATE = OK
                THEN MODE_BLK.ACTUAL = MAN;
        END_IF

        IF TARGET_MODE = AUTO    & MODE_BLK.ACTUAL = OUT_OF_SERVICE
                                 & RESOURCE_STATE = OK
                THEN MODE_BLK.ACTUAL = MAN;
        END_IF
```

```
        IF TARGET_MODE = AUTO   & MODE_BLK.ACTUAL = MAN
                                & RESOURCE_STATE = OK
                                & TB_MEASURED_VALUE.STATUS = NC_GOOD
                THEN MODE_BLK.ACTUAL = AUTO;
        END_IF

                (* TB_MEASURED_VALUE is a call by reference in the PROFIBUS/FF FB
model *)

        IF TARGET_MODE = AUTO   & MODE_BLK.ACTUAL = MAN
                                & RESOURCE_STATE = OK
                                & TB_MEASURED_VALUE.STATUS <> NC_GOOD
                THEN MODE_BLK.ACTUAL = MAN;
        END_IF

        IF TARGET_MODE = AUTO   & MODE_BLK.ACTUAL = AUTO
                                & RESOURCE_STATE = OK
                                & TB_MEASURED_VALUE.STATUS <> NC_GOOD
                THEN MODE_BLK.ACTUAL = MAN;
        END_IF


(* calculation of the OUT status *)

        IF MODE_BLK.ACTUAL = OUT_OF_SERVICE
                THEN OUT.STATUS := BAD_ OUT_OF_SERVICE;
        END_IF

        IF MODE_BLK.ACTUAL = MAN
                THEN OUT.STATUS := UNCERTAIN_NON_SPECIFIC;
        END_IF

        IF MODE_BLK.ACTUAL = AUTO
                        & TB_MEASURED_VALUE.STATUS = NC_GOOD
                        & SIMULATE.ENABLE = FALSE
                THEN OUT.STATUS := NC_GOOD_NON_SPECIFIC;
        END_IF

        IF MODE_BLK.ACTUAL = AUTO
                        & TB_MEASURED_VALUE.STATUS = HIGH_LIMIT
                        & SIMULATE.ENABLE = FALSE
                THEN OUT.STATUS := HIGH_LIMIT;
        END_IF

        IF MODE_BLK.ACTUAL = AUTO
                        & TB_MEASURED_VALUE.STATUS = LO_LIMIT
                        & SIMULATE.ENABLE = FALSE
                THEN OUT.STATUS := LO_LIMIT;
        END_IF
END_ALGORITHM


(*   -------------------------------------------------------------------------------------------------------------------
*)

ALGORITHM AI_CALC_IN_ST

VAR     FIELD_VALUE : FLOAT; END_VAR


(* FB simulation *)
```

```
    IF SIMULATION.ENABLE = TRUE
        THEN  TB_MEASURED_VALUE.STATUS := SIMULATION.STATUS;
              TB_MEASURED_VALUE.VALUE := SIMULATION.VALUE;
    END_IF;


(* calculation of MEAS_VALUE in % *)

    FIELD_VALUE := (TB_MEASURED_VALUE.VALUE – PV_SCALE-EU0) /
                   (PV_SCALE-EU100 – PV_SCALE-EU0)

    PV.VALUE := FIELD_VALUE * (OUT_SCALE-EU100 – OUT_SCALE-EU0)
                   + OUT_SCALE.EU0

(* calculation of OUT in Engineering Unit *)

    IF MODE_BLK.ACTUAL = AUTO

    OUT.VALUE := PV.VALUE
    END_IF


(* limit check and Alarm generation *)

    IF OUT.VALUE > HI_LIM THEN        OUT.STATUS := HIGH_LIMIT;
                                      HI_ALM.VALUE := OUT.VALUE;
                                      HI_ALM.ALARM_STATE := 1; (* ... *)
                                      HI_ALM.SUBCODE := 1; (* ... *)
    END_IF;

    IF OUT.VALUE > HI_HI_LIM THEN     OUT.STATUS := HIGH_LIMIT;
                                      HI_HI_ALM.VALUE := OUT.VALUE;
                                      HI_ALM.ALARM_STATE := 1; (* ... *)
                                      HI_ALM.SUBCODE := 1; (* ... *)
    END_IF;

    IF OUT.VALUE < LO_LIM THEN        OUT.STATUS := LO_LIMIT;
                                      LO_ALM.VALUE := OUT.VALUE;
                                      HI_ALM.ALARM_STATE := 1; (* ... *)
                                      HI_ALM.SUBCODE := 1; (* ... *)
    END_IF;

    IF OUT.VALUE < LO_LO_LIM THEN OUT.STATUS := LO_LO_LIMIT;
                                      LO_LO_ALM.VALUE := OUT.VALUE;
                                      HI_ALM.ALARM_STATE := 1; (* ... *)
                                      HI_ALM.SUBCODE := 1; (* ... *)
    END_IF;

END_ALGORITHM


END_FUNCTION_BLOCK
```

Annex E
(informative)

FB requirement overview support

E.1 General

The life-cycle requirements and the end-user requirements do not reflect the structure of the FB architecture consisting of FB, FB environment, communication and system management.

E.2 The FB requirement overview support.

Table E.1 allocates the components of the FB architecture.

Table E.1 – Allocation of requirements to architecture components

Requirement subclause bullets	FB	FB environment	Communication	Device and system management
2.1 – 1, 2, 5, 8	X	-	-	-
2.1 – 3	X	-	-	X
2.1 – 6	-	-	-	X
2.1 – 4, 7, 9, 10	-	-	-	-
3.1 – 1 to 7	X	-	-	-
3.2.1 to 3.2.3	X	-	-	-
3.3.1	-	X	-	X
3.3.2	X	-	-	-
3.3.3 – 1 to 3	X	-	-	-
3.4 – 1 to 3	X	-	-	X
3.5.1 – 1 to 4	X	-	-	-
3.5.2 – 1 to 7	-	-	-	X
3.5.3 – 1 to 3	X	X	X	-
3.5.4 – 1 to 12	-	-	-	X
3.5.5 – 1 to 3	-	-	-	X
3.5.6	-	x	x	X
3.5.7 – 1 to 2	-	x	-	-
3.5.8 1 – to 10	x	x	-	-
3.5.9	x	x	-	-

The above subclauses present requirements from the end-user, the device-vendor and the engineering point of view. Some requirements overlap. For better understanding, requirements are in the wording of the according point of view. Table E.2 provides an overview concerning requirements related to each other. Requirements with no relationship to one another are not listed in Table E.2.

Table E.2 – Overview of related requirements

Requirement subclause bullets	Related req. 1 subclause-bullets	Related req. 2 subclause bullets
2.1 – 3	4.6.1.1	4.6.9 –1,4
2.1 – 6	4.4.4	
2.1 – 8,9,10	4.3.2	
3.1	4.1.2	4.3.3
3.2.1, 3.2.2	4.2.3 –2	
3.2.3	4.2.3	
3.3	4.6.7	
3.4	4.2.3 – 8	
3.5.1	2.1 – 1 to 3	4.3.4
3.5.2	4.4.1	
3.5.3	4.3.4.	
3.5.4	4.6.5.	
3.5.5	4.6.5.	5.1.1.6 a) to c)
3.5.6	4.3.4.7	
3.5.7	4.6.7. – 1 to 6 , 10	
3.5.8	4.2.3. – 9	
3.5.9	4.2.3. – 9	

Annex F
(informative)

AME requirements

F.1 General

The specification of system and network management is outwith the scope of this part.

The following information is not normative for application of this standard. However, the AME requirements described in this annex are anticipated for any practical realization of a system of distributed FBs as described in IEC 61804 -1 and IEC 61804-2.

F.2 System management requirements

F.2.1 Tags

A device needs at least one unique identifier. This can be achieved by the following four resource parameters:

a) Manufacturer’s name;

b) type or model number:

c) revision number;

d) serial number.

These parameters are required to be fixed in the device by the manufacturer.

- In normal operation, a device is required to be named by a single unique device parameter called the physical tag. The physical tag is required to be used to communicate with the device during normal system operation. The physical tag is required to be retained by a device (i.e. non-volatile) when the device is removed from the system or when power is lost and then restored.
- System and network management is required to have functions to determine that a device has no assigned physical tag (i.e. has a default or null tag), to identify the device by manufacturer/model number/serial number, and to assign a physical tag to it. These functions are required to operate without interfering with communications between other devices in the system.
- System and network management is required to have a function to determine that a tag (physical tag or block tag) is unique in the system (i.e. unique across both physical and block tags). System and network management is required to execute this function whenever a new tag is assigned, a tag is changed, or a new device with pre-configured tag (or tags) is added to a system. This function is required to operate without interfering with communications between other devices in the system.
- In devices having a single FB and FB resource, the physical tag, resource block tag, and FB tag are required to be the same tag.
- In devices having a single FB resource, the physical tag, and resource block tag are required to be the same tag.

F.3 Network addresses

a) System and network management are required to have functions to allow the addition or removal of devices from the system without interfering with the network addresses or operation of other devices in the system.

b) System and network management are required to support addresses for three types of devices:

   1) temporary devices;

   2) new devices (without system-assigned addresses);

   3) system devices (with system-assigned addresses).

c) System and network management are required to support three methods of address assignment:

   1) pre-assigned;

   2) system assigned – fixed;

   3) system assigned – dynamic.

d) Pre-assigned node addresses are fixed in a device via hardware jumpers or switches. Fixed system-assigned node addresses are retained in a device (i.e. are non-volatile) when the device is removed from the system or power is lost and then restored. Dynamic node addresses are not retained by the device (i.e. are volatile) when the device is removed from the system or power is lost.

e) System and network management is required to have functions to allow the system to determine the node-address assignment method supported by a device.

f) Devices that support system-assigned node address assignment are required to support both the fixed and the dynamic methods. Either the fixed or the dynamic system-assigned node addressing method are required to be user-configured in the device by means of a hardware jumper or switch.

g) There is a one-to-one correspondence between the physical tag and the node address of a device. A device supporting a system-assigned node address is required to have an assigned physical tag before the system will assign a node address to that device. Devices with pre-assigned node addresses are required to have an assigned physical tag before they are permitted to operate normally in the system using their pre-configured node address (i.e. only system management operations are allowed for a device with a pre-configured node address and no assigned physical tag).

h) System and network management is required to have functions to allow the system to determine that the retained node-address in a new device (either pre-configured or fixed) is unique in the system. System and network management is required to execute this function whenever a new a new device is added to a system. This function is required to operate without interfering with the node addresses or operation of other devices in the system.

i) System and network management is required to prevent duplicate node addresses in systems with some devices having pre-assigned node addresses and other devices having system assigned node addresses. System and network management is required to also prevent duplicate node addresses in systems with some devices having fixed system-assigned node addresses and other devices having dynamic system-assigned node addresses.

   NOTE This implies that devices with either pre-configured or fixed (non-volatile) node addresses are required to support the same System and Network Management node address assignment protocol as devices with dynamic node addresses.

j) System and network management is required to prevent the introduction of a new device containing a tag that duplicates a tag already operating in the system. This is required to be achieved by preventing operation of such new device if it supports pre-assigned node addresses, and by not assigning a node address to such new device if it supports system-assigned node addresses (either fixed or dynamic).

F.4 Data object naming convention

The addressing/naming of parameters in a DFBAP has different views, the FB application view, the communication configuration view, the device identification view and possibly others. The scope of identifiers differs in a specified part of the system, i.e. identifiers are unique in a defined area. This is required to be precisely defined in a specification.

A possible approach is shown in Figure F.1.

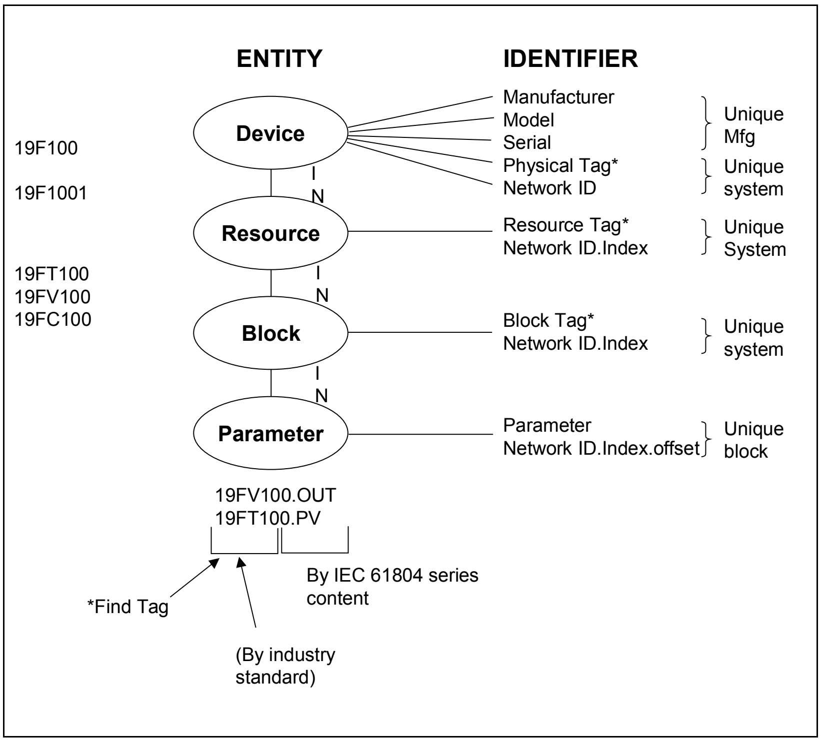


Figure F.1 – Naming and addressing approach

F.5 Device communications relationships

a) System and network management is required to have functions to establish and to terminate communications relationships between functions in different devices. These functions are required to operate without interfering with the existing communications relationships or the operation of other devices and functions in the system.

b) Communications relationships between functions in different devices are required to be made by reference to “tag parameter”. System and network management is required to have functions to determine the “node address and index” reference in a device, given the “tag parameter” reference, and vice versa, to support the configuration of communications relationships in the FB environment.

c) Established communications relationships between functions in different devices are required to be retained when power is lost and then restored.

F.6 Time synchronization

F.6.1 Time object

System and network management is required to include data structure definitions and associated functions to correlate FB Environment time to data link time, called a time object. This is required to support the coordination of data communications and FB execution.

F.6.2 Time publisher

a) System and network management is required to include data structure definition functions and protocols needed to send application time synchronization to distributed FB environments (called a time publisher). This is required to support the coordination of data time-stamping and function execution in distributed applications.

b) System and network management is required to include the functions and protocols needed to support redundant time publishers (primary and multiple secondaries).

F.6.3 Time subscriber

System and network management is required to include data structure definitions and associated functions to receive application time synchronization for distributed FB environments (called a time subscriber). This is required to operate in conjunction with the time publisher(s).

F.6.4 Local time

System and network management time distribution methods are required to support the distribution of both local and universal time.

F.6.5 Network (link) time

a) System and network management is required to include functions and protocols to synchronize network (link) time across multiple bus segments. This is required to support the synchronization of time-critical communications between multiple segments.
b) System and network management is required to also include functions and protocols to synchronize network (link) time between a bus segment and a higher-level system (for example a DCS or PLC controller). This is required to support the synchronization of time-critical communications between the segment and the higher-level system (i.e. synchronize FB execution and communication in cascade structures between segment devices and higher-level system devices).

F.7 Redundant system components

a) System and network management is required to include functions and protocols to manage N-redundant system components, including N-redundant:
   1) media;
   2) physical layers;
   3) devices (including gateways, routers, bridges, repeaters, and end devices);
   4) application processes within a device.
b) System and network management is required to include data, functions and protocols to manage simplex-to-duplex bridges and routers.
c) System and network management is required to include data, functions and protocols to manage duplex media and physical layer in a single device.
d) System and network management is required to include data, functions and protocols to support channel selection for a device, segment, and network.
e) System and network management is required to include data, functions and protocols to support the redundancy aspects of certain application processes, such as:
   1) router and gateway application processes;
   2) voter application processes;
   3) application processes containing redundant application entities (i.e. redundant FBs).

F.8 Diagnostics

a) System and network management is required to include data structures, functions and protocols to manage the performance of system components, including:
   1) performing preventive maintenance;
   2) diagnosing intermittent errors;
   3) notifying of and locating faults;
   4) warning of incipient faults and failures;
   5) recovering from faults and failures.
b) System and network management diagnostic data structures, functions and protocols are required to service:
   1) media;
   2) devices;
   3) applications.
c) System and network management diagnostic data structures, functions and protocols are required to include at least the following diagnostic counters:
   1) total number of communications messages to or from this FB resource;
   2) total number of CRC errors to or from this FB resource;
   3) total number of framing errors to or from this FB resource;
   4) total number of "parameter not supported" errors to or from this FB resource;
   5) total number of receive buffer full errors to this FB resource;
   6) total number of alert buffer full errors to this FB resource;
   7) total number of FB Services Busy errors to or from this FB resource;
   8) total number of send buffer not emptied errors from this FB resource;
   9) total number of other communication errors (not listed above) to or from this FB resource.
d) These counters are required to be initialized to zero, and system and network management is required to provide services to reinitialize them. System and network management is required to initiate a FB environment rollover alert message if roll over occurs, containing the time and identity of the counter that rolled over.
e) System and network management diagnostic data structures, functions and protocols are required to include a diagnostic counter reset time parameter, which will contain the (local) time.
f) System and network management diagnostic data structures, functions and protocols are required to include an error rate parameter, defined as the sum of the number of CRC and framing errors per communications cycle.
g) System and network management diagnostic data structures, functions and protocols are required to include a configurable error rate limit parameter. If the error rate exceeds this error rate limit, then system and network management is required to initiate a FB environment error rate limit alert message, containing the time, and the current error rate.
h) System and network management diagnostic functions and protocols are required to include the capability to control and perform physical layer loopback communications tests, for devices that are capable of supporting loopback testing. Loopback testing of one end device in a system is required not to interfere with the time-critical communications of other end devices in the system.

F.9 Communications management

System and network management services are required to include data structures, functions and protocols to establish and maintain the communications configuration of all protocol layers in all devices (i.e. manage all layer management entities in all devices).

F.10 System and network management services

The following system and network management services are required to be provided:

a) set physical tag (with or without assigned node address);
b) set address (only with physical tag);
c) clear address (only with physical tag);
d) identify device (physical tag and node address);
e) find tag (physical or block);
f) FB start (with block tag);
g) management information base access (read and write);
h) publish time (application and link).

F.11 Integration of management in the device model

There is a need of an unambiguous specification of a management model within the device model. This is the prerequisite to specify the interface between the AME and the FB environment.

Bibliography

IEC 60050-351:1998, *International Electrotechnical Vocabulary (IEV) – Part 351: Automation control. Terms and definition*

IEC 60870-6-503:2002, *Telecontrol equipment and systems – Part 6-503: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Services and protocol*

IEC 60870-6-702:1998, *Telecontrol equipment and systems – Part 6-702: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – Functional profile for providing the TASE.2 application service in end systems*

IEC 61069-1:1991, *Industrial-process measurement and control – Evaluation of system properties for the purpose of system assessment – Part 1: General considerations and methodology*

IEC 61069-5:1994, *Industrial-process measurement and control – Evaluation of system properties for the purpose of system assessment – Part 5: Assessment of system dependability*

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

ISO/IEC 10000-1:1998, *Information technology – Framework and taxonomy of International Standardized Profiles – Part 1: General principles and documentation framework*

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

ISO 15745-1:2003, *Industrial automation systems and integration – Open systems application integration framework – Part 1: Generic reference description*

ISO Guide to the expression of uncertainty in measurement (1995)

ISA-TR50.02, Part 9-2000, *Fieldbus Standard for Use in Industrial Control Systems: User Layer Technical Report*

PRIAM:1995, *Prenormative Requirements for Intelligent Actuation and Measurement* (Esprit Project, 1992-1995)

NOAH: *Network Oriented Application Harmonisation* (EU Project [1998 to 2000], EP 26951)

VDI/VDE 3696:1995, *Herstellerneutrale Konfigurierung von Prozeßleitsystemen*

BSI 7986:2001, *Specification for Data Quality Metrics for Industrial Measurement and Control Systems*

EOQ (European Organisation for Quality): *Glossary of terms used in the management of quality* (Bern [1989] – 777 p.)

Afnor NFX 60-010: *Maintenance – Concepts and definitions of Maintenance activities*

Standards Survey

The IEC would like to offer you the best quality standards possible. To make sure that we continue to meet your needs, your feedback is essential. Would you please take a minute to answer the questions overleaf and fax them to us at +41 22 919 03 00 or mail them to the address below. Thank you!

Customer Service Centre (CSC)

International Electrotechnical Commission
3, rue de Varembé
1211 Genève 20
Switzerland

or

Fax to: **IEC**/CSC at +41 22 919 03 00

Thank you for your contribution to the standards-making process.

Nicht frankieren
Ne pas affranchir

Non affrancare
No stamp required

A Prioritaire

RÉPONSE PAYÉE

SUISSE

Customer Service Centre (CSC)
International Electrotechnical Commission
3, rue de Varembé
1211 GENEVA 20
Switzerland

Q1 Please report on **ONE STANDARD** and **ONE STANDARD ONLY**. Enter the exact number of the standard: *(e.g. 60601-1-1)*

....

Q2 Please tell us in what capacity(ies) you bought the standard *(tick all that apply).* I am the/a:

- purchasing agent ❑
- librarian ❑
- researcher ❑
- design engineer ❑
- safety engineer ❑
- testing engineer ❑
- marketing specialist ❑
- other...

Q3 I work for/in/as a: *(tick all that apply)*

- manufacturing ❑
- consultant ❑
- government ❑
- test/certification facility ❑
- public utility ❑
- education ❑
- military ❑
- other...

Q4 This standard will be used for: *(tick all that apply)*

- general reference ❑
- product research ❑
- product design/development ❑
- specifications ❑
- tenders ❑
- quality assessment ❑
- certification ❑
- technical documentation ❑
- thesis ❑
- manufacturing ❑
- other...

Q5 This standard meets my needs: *(tick one)*

- not at all ❑
- nearly ❑
- fairly well ❑
- exactly ❑

Q6 If you ticked NOT AT ALL in Question 5 the reason is: *(tick all that apply)*

- standard is out of date ❑
- standard is incomplete ❑
- standard is too academic ❑
- standard is too superficial ❑
- title is misleading ❑
- I made the wrong choice ❑
- other ....

Q7 Please assess the standard in the following categories, using the numbers:
(1) unacceptable,
(2) below average,
(3) average,
(4) above average,
(5) exceptional,
(6) not applicable

- timeliness ....
- quality of writing....................................
- technical contents.................................
- logic of arrangement of contents ....
- tables, charts, graphs, figures...............
- other ....

Q8 I read/use the: *(tick one)*

- French text only ❑
- English text only ❑
- both English and French texts ❑

Q9 Please share any comment on any aspect of the IEC that you would like us to know:

....
....
....
....
....
....
....
....
....
....
....
....

ISBN 2-8318-7212-X

9 782831 872124

ICS 35.200; 35.240.50

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND