User Manual

Self-description Data Documentation and Guidelines

ITER plant systems have to be configured and controlled by the ITER central I&C systems (CODAC, interlocks and safety). Configuration settings of plant systems, though being plant specific, can be described in generic terms. This set of configuration data is called plant system self-description data. This document describes the data model of the self-description data and methods to create and maintain these data.

---

Approval Process

<table>
<thead>
<tr>
<th>Name</th>
<th>Action</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Stepanov D.</td>
<td>11-Feb-2011:signed</td>
</tr>
<tr>
<td>CoAuthor</td>
<td></td>
<td>IO/DG/DIP/CHD/CODAC/CDC</td>
</tr>
<tr>
<td>Reviewers</td>
<td>Klotz W.- D.</td>
<td>15-Mar-2011:recommended</td>
</tr>
<tr>
<td></td>
<td>Wallander A.</td>
<td>11-Feb-2011:recommended</td>
</tr>
<tr>
<td>Approver</td>
<td>Bora D.</td>
<td>18-Mar-2011:approved</td>
</tr>
<tr>
<td>Document Security: level 1 (IO unclassified)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RO: Stepanov Denis

Read Access: AD: ITER, AD: External Collaborators, AD: Section - CODAC, project administrator, RO
<table>
<thead>
<tr>
<th>Version</th>
<th>Latest Status</th>
<th>Date</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2.1</td>
<td>Approved</td>
<td>11 Feb 2011</td>
<td>Reacted on comments from the external review. Done proofreading. Updated fig. 3-26 and 3-46.</td>
</tr>
<tr>
<td>v2.0</td>
<td>Signed</td>
<td>10 Jan 2011</td>
<td>Update to reflect 2010 developments and in preparation for the PCDH v6.</td>
</tr>
<tr>
<td>v1.1</td>
<td>Approved</td>
<td>01 Feb 2010</td>
<td>Updated the model in accordance with the latest developments and described it in more detail. Incorporated reactions to comments received.</td>
</tr>
<tr>
<td>v1.0</td>
<td>Approved</td>
<td>07 Dec 2009</td>
<td></td>
</tr>
</tbody>
</table>
# Table of contents

1 INTRODUCTION .................................................................................................................................. 3
   1.1 Purpose ........................................................................................................................................ 3
   1.2 PCDH context ................................................................................................................................. 3
   1.3 Scope ........................................................................................................................................... 3
   1.4 Definitions and Abbreviations ....................................................................................................... 4
   1.5 Reference documents ................................................................................................................... 5

2 SELF-DESCRIPTION DATA METHODOLOGY .................................................................................. 6
   2.1 Self-description concept .................................................................................................................. 6
   2.2 Self-description database ................................................................................................................ 6
   2.3 Self-description data life cycle ....................................................................................................... 7

3 DATA MODEL DESCRIPTION ........................................................................................................ 9
   3.1 CODAC markup language ............................................................................................................. 9
   3.2 I&C naming conventions ................................................................................................................. 9
      3.2.1 Plant breakdown structure (PBS) ............................................................................................ 9
      3.2.2 Part numbering convention .................................................................................................... 10
      3.2.3 Signal naming convention ..................................................................................................... 11
      3.2.4 Functional breakdown structure (FBS) ................................................................................... 12
      3.2.5 Function naming convention .................................................................................................. 13
      3.2.6 Variable naming convention ................................................................................................. 13
   3.3 Plant system .................................................................................................................................. 14
      3.3.1 General structure ..................................................................................................................... 14
      3.3.2 Plant system parts ................................................................................................................... 15
      3.3.3 Plant system signals ................................................................................................................ 15
      3.3.4 Signal general properties ....................................................................................................... 16
      3.3.5 Signal physics properties ....................................................................................................... 16
      3.3.6 Signal I&C properties ............................................................................................................ 17
      3.3.7 Signal importance properties ............................................................................................... 17
      3.3.8 Signal display properties ....................................................................................................... 18
   3.4 Plant system I&C ......................................................................................................................... 19
      3.4.1 General structure ..................................................................................................................... 19
      3.4.2 Control unit ............................................................................................................................. 20
      3.4.3 Controller ................................................................................................................................. 21
      3.4.4 Input / output board .................................................................................................................. 21
      3.4.5 Input / output board channel ................................................................................................... 22
      3.4.6 Slow controller ......................................................................................................................... 22
      3.4.7 Fast controller .......................................................................................................................... 23
      3.4.8 Plant system host ..................................................................................................................... 23
      3.4.9 EPICS input / output controller ............................................................................................... 24
      3.4.10 EPICS database .................................................................................................................... 25
      3.4.11 EPICS records ....................................................................................................................... 25
      3.4.12 EPICS channel access settings ............................................................................................. 25
      3.4.13 EPICS drivers ......................................................................................................................... 27
      3.4.14 EPICS S7PLC driver ............................................................................................................... 27
      3.4.15 EPICS modules ....................................................................................................................... 28
      3.4.16 Variables ................................................................................................................................ 28
      3.4.17 Variable's data type ............................................................................................................... 29
      3.4.18 Array data type ....................................................................................................................... 30
      3.4.19 Enumeration data type .......................................................................................................... 31
      3.4.20 Bit pattern data type .............................................................................................................. 31
      3.4.21 Variable’s archiving info ...................................................................................................... 31
3.4.22 Variable deployment information ................................................................................................ 32
3.4.23 Plant control system .................................................................................................................. 32

3.5 CODAC data structures relevant for plant systems ........................................................................ 32

3.5.1 Catalogs ..................................................................................................................................... 33
3.5.2 Supported I/O board descriptions .............................................................................................. 33
3.5.3 I/O board technical data ............................................................................................................. 34
3.5.4 I/O board channel parameters ................................................................................................... 35
3.5.5 I/O board physical requirements ............................................................................................... 36
3.5.6 I/O board power requirements .................................................................................................. 37
3.5.7 Connectors ................................................................................................................................. 37
3.5.8 CODAC servers ......................................................................................................................... 38
3.5.9 EPICS database model ............................................................................................................... 39
3.5.10 EPICS predefined records ....................................................................................................... 41
3.5.11 Linux settings ........................................................................................................................... 41
3.5.12 Linux drivers ............................................................................................................................ 41

3.6 Static data sets ............................................................................................................................. 42

3.7 Examples ...................................................................................................................................... 43

3.7.1 Plant system with a slow controller .......................................................................................... 43
1 INTRODUCTION

1.1 Purpose

ITER plant systems have to be configured and controlled by the ITER central I&C systems (CODAC, interlocks and safety). Configuration settings of plant systems, though being plant specific, can be described in generic terms. This set of configuration data is called plant system self-description data. This document describes the data model of the self-description data and methods to create and maintain these data.

1.2 PCDH context

The Plant Control Design Handbook (PCDH) [RD1] defines methodology, standards, specifications and interfaces applicable to ITER plant systems instrumentation and control (I&C) system life cycle. I&C standards are essential for ITER to:

- Integrate all plant systems into one integrated control system;
- Maintain all plant systems after delivery acceptance;
- Contain cost by economy of scale.

PCDH comprises a core document which presents the plant system I&C life cycle and recaps the main rules to be applied to the plant system I&Cs for conventional controls, interlocks and safety controls. Some I&C topics will be explained in greater detail in dedicated documents associated with PCDH as presented on Figure 1-1. This document is one of them. Its objective is to describe the self-description concept and implementation in more detail.

![PCDH documents structure](image)

Figure 1-1: PCDH documents structure

1.3 Scope

This document does not describe the self-description data in its entirety, but rather focuses on key elements which serve as the building blocks of the ITER I&C systems. The detailed information can be obtained from the iter-schemas package provided as a part of the CODAC Core System software distribution.
### 1.4 Definitions and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D</td>
<td>Three Dimensional</td>
</tr>
<tr>
<td>AAAA</td>
<td>ISA class</td>
</tr>
<tr>
<td>AC</td>
<td>Alternating Current</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard for Code Information Interchange</td>
</tr>
<tr>
<td>ASL</td>
<td>Access Security Level</td>
</tr>
<tr>
<td>CA</td>
<td>Channel Access</td>
</tr>
<tr>
<td>CATIA</td>
<td>Computer Aided Three-dimensional Interactive Application</td>
</tr>
<tr>
<td>CIS</td>
<td>Central Interlock System</td>
</tr>
<tr>
<td>CODAC</td>
<td>Control, Data Access and Communication</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off-The-Shelf</td>
</tr>
<tr>
<td>CWS</td>
<td>Cooling Water System</td>
</tr>
<tr>
<td>DA</td>
<td>Domestic Agency</td>
</tr>
<tr>
<td>DB</td>
<td>DataBase</td>
</tr>
<tr>
<td>DBD</td>
<td>DataBase Definition</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>DMA</td>
<td>Direct Memory Access</td>
</tr>
<tr>
<td>EPICS</td>
<td>Experimental Physics and Industrial Control System</td>
</tr>
<tr>
<td>FBS</td>
<td>Functional Breakdown Structure</td>
</tr>
<tr>
<td>FFF</td>
<td>FBS node</td>
</tr>
<tr>
<td>GMT</td>
<td>Greenwich Mean Time</td>
</tr>
<tr>
<td>HMI</td>
<td>Human-Machine Interface</td>
</tr>
<tr>
<td>HPN</td>
<td>High Performance Networks</td>
</tr>
<tr>
<td>I&amp;C</td>
<td>Instrumentation and Control</td>
</tr>
<tr>
<td>I/O</td>
<td>Input / Output</td>
</tr>
<tr>
<td>ICH</td>
<td>Ion Cyclotron Heating</td>
</tr>
<tr>
<td>ICP</td>
<td>ITER Collaboration Platform</td>
</tr>
<tr>
<td>ID</td>
<td>IDentifier</td>
</tr>
<tr>
<td>IDM</td>
<td>ITER Document Management</td>
</tr>
<tr>
<td>IO</td>
<td>ITER Organization</td>
</tr>
<tr>
<td>IOC</td>
<td>Input / Output Controller</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IRQ</td>
<td>Interrupt ReQuest</td>
</tr>
<tr>
<td>ISA</td>
<td>International Society of Automation</td>
</tr>
<tr>
<td>ITER</td>
<td>International Thermonuclear Experimental Reactor</td>
</tr>
<tr>
<td>LCC</td>
<td>Local Control Cubicle</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>NI</td>
<td>National Instruments</td>
</tr>
<tr>
<td>NNN</td>
<td>sequential number</td>
</tr>
<tr>
<td>OLC</td>
<td>Operating Limits and Conditions</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PBS</td>
<td>Plant Breakdown Structure</td>
</tr>
<tr>
<td>PCDH</td>
<td>Plant Control Design Handbook</td>
</tr>
<tr>
<td>PCF</td>
<td>Plant system Controller, Fast controls</td>
</tr>
<tr>
<td>PCI</td>
<td>Peripheral Component Interconnect</td>
</tr>
<tr>
<td>PFI</td>
<td>Programmable Function Interface</td>
</tr>
<tr>
<td>PIS</td>
<td>Plant Interlock System</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>PNG</td>
<td>Portable Network Graphics</td>
</tr>
<tr>
<td>PPP</td>
<td>PBS level 3 identifier</td>
</tr>
</tbody>
</table>
1.5 Reference documents

The following documents are referenced in this document:

[RD1] Plant Control Design Handbook (27LH2V v6.1)
[RD5] ITER PBS database (2FBMWF)
[RD6] ITER Numbering System for Parts/Components (28QDBS v1.3), 04 Sep 2008
[RD7] ITER Function Category and Type for ITER Numbering System (2FJMPY v1.4), 17 Dec 2010
[RD8] I&C signal and process variable naming convention (2UT8SH v7.3), 08 Feb 2011
[RD9] ITER Functional Breakdown Structure (43RRBR v1.1), 16 Dec 2010
2 SELF-DESCRIPTION DATA METHODOLOGY

2.1 Self-description concept

ITER plant systems have to be configured and controlled by ITER central I&C systems (CODAC, interlocks and safety). The overall configuration data can be broken down into static and dynamic parts. The part of the data which does not change during plant system operation is called static configuration; it is subject to change through dedicated maintenance procedures. On the other hand, the dynamic configuration data does change during plant system operation.

A data model for the static configuration can be extended to describe the kind of data that can be configured at run-time, thus providing a full image of a plant system to a central control system. Using unified data schema for these data facilitates a fully data driven approach in which central systems are prepared to work with a particular plant system by just reading its description. This approach is called a self-description of plant systems. These static configuration data of plant systems are then called self-description data (SDD) and the data schema for them are called self-description schema. Both the self-description data and the schema for it are subject to rigorous change control.

As stated in the PCDH ([RD1]), the I&C self-description data consist of:

- Plant system I&C unique identification;
- Command list;
- Alarms list;
- Set-points list;
- Plant system I&C design limits;
- Physical (raw) signals list (I/O);
- Processed / converted signals list;
- Data streams list;
- Logging messages list;
- Definition of the plant system I&C state machine in accordance with the defined plant system operating states;
- Definitions of plant system I&C HMI;
- Plant system I&C constant values;
- Default values (“factory settings”) for run-time configuration used for plant system I&C start-up;
- Identification of source codes and binary packages of the plant system I&C specific software;
- Documentation.

This list is not exhaustive and will be further refined in future. Some of these data domains (e.g., state machines) are not yet supported by the SDD software; this is work in progress.

The self-description data is also supposed to help with integration of non-standard I&C solutions, such as COTS devices. At least the interface with the CODAC system could be defined in such a way.

The self-description data are important deliverables of both the I&C design and I&C manufacturing phases (see the PCDH [RD1]).

2.2 Self-description database

ITER currently provides a prototype database to store self-description data as a part of the CODAC Core System distribution. These databases are local to a particular I&C development. A similar instance is planned to be run at ITER premises; it will collect data and coordinate the I&C development across the ITER members.

Some data, relevant for self-description, are stored in other project databases. For example, the plant breakdown structure is managed in the ICP database; I&C diagrams (including parts and signals) are part of the SEE System Design database; geographical and 3D information is available in Enovia (CATIA database). To facilitate the interface between the central and local SDD databases, the I&C relevant data from other databases will be linked to the central SDD database via internal mechanisms and provided to the PS I&C developers as if it was a part of the SDD database. The master copy of the data stays with the original databases and shall be modified directly there.
2.3 Self-description data life cycle

The CODAC Core System and its tools (the SDD editor and the SDD database) are the main instruments to create and store the self-description data. The following diagram (Figure 2-1) illustrates the process of the data creation.

1) The local SDD database is initialized with the design data from the central database (e.g., a list of plant systems, plant parts or plant signals);
2) Plant system &C designer uses the SDD editor to define key &C components, such as control units or &C variables;
3) The information defined is then saved in the local SDD database and translated into the tool-specific format for particular software products (such as SCADA tools);
4) A plant system &C programmer uses these data as input and completes them with the processing algorithms and other tool-specific information;
5) The &C control units are programmed and/or configured with the resulting data. The &C system is then ready for testing;
6) A plant system operator runs the desired control sequences and receives reactions from the plant system;
7) In the case of problems related to the &C, the operator communicates with the PS &C designer and programmer in order to pinpoint and fix the problem. The steps 2→7 are then repeated as needed;
8) When the PS I&C is running smoothly (or at any other moment, if needed), its configuration data can be submitted to the central I&C database;
9) Syndicated self-description data from different plant systems can be used in a number of ways, such as estimating the progress on the PS I&C development, or to providing test data and PS requirements to the CODAC designers and programmers.

This procedure ensures that by the time that the plant system is delivered, CODAC will possess a stable tested snapshot of the PS configuration data. During the plant system commissioning the CODAC system and its tools can be used directly in the same procedure as described above.

When the system is commissioned and is operating, Figure 2-1 is reduced to the following (Figure 2-2).

In this case the SDD database does not directly participate in the process; it is used for service and consultation purposes.
3 DATA MODEL DESCRIPTION

This chapter describes the I&C data model and its mapping to the self-description data schema. The I&C data model consists of two main parts: plant systems and plant system I&Cs. Plant systems implement the required technical functions of the ITER plant, while plant system I&Cs fulfill the control function on the respective plant systems. A single plant system I&C can have control loops in several different plant systems.

The next section gives a short insight into the data modeling technology. The following sections go into details on the topics of I&C naming conventions, definitions of plant systems, plant system I&Cs and selected CODAC data structures.

3.1 CODAC markup language

An XML technology [RD2] was chosen to express self-description data, as it is application independent and is purely data centred. There is a vast set of XML processing tools available on the market and more and more software products are becoming XML-aware. This makes the transition from a pure system description to the actual software implementation easier.

The data model for XML documents can be described using XML Schema notation, which is a well recognised standard, developed by the W3C group ([RD3]). XML Schema definitions (XSD) allow the construction of very complex data structures with strict data validation, whilst remaining application independent. Many data domains possess data descriptions written in XSD. Technologies exist which allow application of these data descriptions to processing software and data storage (e.g., databases).

The XML Schema is used as a generic data modeling approach for I&C data. When it comes to the actual implementation, this data model can be re-expressed by means which are more suitable to a particular application or a tool (e.g., a relational data model, or Java or Python classes). Nevertheless, when the need to exchange data using XML arises, the current model provides a ready format for the files.

The I&C data structure of ITER is described using a set of XSD files. These files, in fact, define a CODAC markup language – a universal means of preparing and maintaining plant system configurations.

The XML Schema data models are managed using the ITER version control repository and are distributed as a part of the CODAC Core System.

3.2 I&C naming conventions

Management of I&C objects is not possible without rigid naming conventions applied and adhered to uniformly throughout the project. Some of them are already defined, others are under consideration. This section describes several naming schemes which are already established.

3.2.1 Plant breakdown structure (PBS)

The ITER facility is broken into functional parts in a hierarchical manner. The breakdown structure and rules are defined in the document [RD4]. The self-description schema implements this breakdown as shown on Figure 3-1 (three levels of the PBS are shown).
The schema defines seven PBS levels with a possibility of further extension. Each PBS node is assigned an identifier (two characters) and a description. In accordance with the current PBS database implementation ([RD5]) the PBS is versioned at level 1 only (i.e. the whole sub-tree of the PBS level 1 has the same version).

An example of the PBS level 3 identifier would be 51HV11, which represents the ICH power supply.

The PBS identifiers are the basis of other naming schemes, including those which are I&C related.

3.2.2 Part numbering convention

ITER I&Cs have to reference physical objects which are to be controlled. These functional names (called part numbers) are defined in the document [RD6]. The structure of the part number is shown on Figure 3-2.
The first three levels of the PBS (called PPPPPP) are supplemented with a function category designator (TTT) and a sequential number (NNNN). The plant-wide registry of the TTT codes is managed centrally in the document [RD7].

An example of a part number would be 51HV11-EW-0001, which represents the ICH system’s power supply source № A1. The current ITER breakdown can be consulted in the online PBS database [RD5].

### 3.2.3 Signal naming convention

Signals are cornerstone objects of I&C; they are used to name and represent data coming from the plant as well as control information going back to the plant from central systems. The naming convention for signals is defined in the document [RD8] and implemented as shown on Figure 3-3.
Thus, the signal name includes the name of the component the signal is related to, as well as a number of additional properties such as an ISA class (AAAA), a designator of particular element of the class (RRRR) and other signal qualifiers (SSS). The SSS code is broken down into the importance specifier (CODAC, interlocks, or safety) and an SS code, which lists some particular signal properties.

The list of AAAA and SS codes is defined in the document [RD8].

An example of a signal name would be 26CCC2-PV-0010:PCVZ10-CRC, which represents a valve actuator in a non-radioactive component cooling water system.

### 3.2.4 Functional breakdown structure (FBS)

While the PBS is convenient for definition of the physical objects of the ITER plant, their operation and control requires a different view of the plant. This “functional” breakdown is defined in [RD9]. Its structure (very similar to the PBS one) is shown on the Figure 3-4 (three levels of the FBS are shown).
The schema defines four FBS levels with a possibility of further extension. Each FBS node is assigned an identifier (one to four characters) and a description. The FBS level 1 also provides references to the PBS level 1 nodes it covers, in order to facilitate understanding of the high-level link between the two hierarchies.

3.2.5 Function naming convention

ITER plant functions are just nodes in the FBS tree; their naming is defined in the document [RD8] and is shown on Figure 3-5.

A maximum of three levels of the FBS are used for I&C purposes. An example of a level 2 function would be RF-ICH-RFIH, meaning delivering the ICH and current drive.

There is no database for functions established yet; some high level functions are listed in [RD9].

3.2.6 Variable naming convention

Variables are representations of the process-important data in I&C controllers. Their naming convention is defined in the document [RD8] and is shown on Figure 3-6.
Thus, the variable name includes the name of the function the variable is related to, plus an identifier depending on the nature of the variable. For variables representing signals it is formed from the parts of the signal name (TTT, NNNN, AAAA, and RRRR; see section 3.2.3). For other variables it is a free-format identifier of maximum 16 characters (V16).

An example of a variable name would be **CWS-CWCT:PV0010-PCVZ**, which represents a valve actuator in a non-radioactive component cooling water system.

### 3.3 Plant system

A plant system is an autonomous part of the ITER plant which implements and is responsible for a given technical function. A self-description schema for a plant system does not list all possible plant system properties, but focuses on those interesting from the controls point of view.

#### 3.3.1 General structure

The structure of a plant system is shown on Figure 3-7.
A plant system is identified by a PBS level 1 code and a name. It consists of the list of plant components (parts) and of the list of signals produced by the plant system.

Each description of a plant system is supplemented with a version number, which allows tracking evolution of the plant description in time.

### 3.3.2 Plant system parts

A plant system component is basically any physical part of the plant system identified by a part number. It could be either I&C related (e.g., a controller) or not (e.g., a valve). The definition of the plant system component is shown on Figure 3-8.

Each part has a name, which is a part number in accordance with the part numbering convention (see section 3.2.2) and an optional description.

### 3.3.3 Plant system signals

A signal is a piece of information about the state of the plant. The definition of the plant system signal is shown on Figure 3-9.
Each signal has a name in accordance with the signal naming convention (see section 3.2.3) and an optional description.

A signal possesses a number of properties, which are described below. They represent a piece of the plant system design information, which helps I&C designers to determine the proper approach for signal handling in the plant system I&Cs. These properties are also utilised by the central I&C systems during signal processing and presentation.

### 3.3.4 Signal general properties

General properties are basic properties of the signal; they are essential for the signal description. The list of general properties is shown on Figure 3-10.

![Figure 3-10: Signal general properties](image)

The source indicates the origin of the signal (a reference to a plant part). The direction of the signal is defined from the point of view of the central I&C systems (input, output or bidirectional).

### 3.3.5 Signal physics properties

Physics properties tell more about the physics process going on behind the signal. The list of physics properties is shown on Figure 3-11.
A physical value and its unit of measurement indicate the physical value being measured or controlled. Maximum signal frequency indicates the characteristic speed of the process. This value is used to select the proper technology to implement this signal (e.g., a slow or fast controller) and to collect information on processing demands of the plant systems.

### 3.3.6 Signal I&C properties

I&C properties go into more details on information relevant for I&C systems. The list of I&C properties is shown on Figure 3-12.

The signals can be of analog or digital nature. The signal type suggests its typical use within the I&C systems (e.g., an oscillogram or a command). Digital resolution tells how many bits are needed to properly represent the signal value in memory. Fixed signal lines can provide their constant value here. Finally, there is a way to indicate that the signal needs to be archived.

### 3.3.7 Signal importance properties

The importance properties deal with the signal significance in the system. The list of importance properties is shown on Figure 3-13.
These properties allow specification of whether the signal belongs to the conventional control, plasma control, interlocks or safety and if it requires any redundancy (is duplicated elsewhere in the system). For safety signals there is an indication of a safety class (SIC-1, SIC-2 or safety related). Finally, there are fields to specify signal's operating limits and alarm limits.

3.3.8  Signal display properties

Display properties provide guidance on building plant system mimic screens and on presentation of the signals to operators. The list of display properties is shown on Figure 3-14.
The properties allow the definition of a display name for the signal and its boundaries, as well as its increment value and frequency of the screen update. There is also a way to tell if the signal might require operator’s attention and if the signal value can be altered by the operator.

### 3.4 Plant system I&C

Plant system I&C is a synthesis of hardware and software applied to a plant system’s technical process in order to attain the process’ objective.

#### 3.4.1 General structure

Whilst a plant system descriptor talks about a plant system in general, the plant system I&C deals specifically with I&C components and their properties. The structure of the plant system I&C is shown on Figure 3-15.
Figure 3-15: Structure of a plant system I&C

Plant system I&C is identified by an identifier and a description. The format of the identifier is not constrained in this version of schema, but it is recommended to use a function name (see section 3.2.5) from level 2 or 3. Each plant system I&C has only one plant system host (PSH) which serves for configuration and control purposes.

The plant system I&C is further split into the conventional control, interlock and safety parts, with interlocks and safety being optional. In addition to that, the plant system I&C, as a whole, possesses a list of I&C variables (see section 3.4.16). For the definition of the plant control system section 3.4.23 should be consulted; the definitions for the plant interlock and safety systems will be elaborated in future versions of the schema.

Like the case of a plant system, the description of a plant system I&C is supplemented with a version number, which allows tracking of the plant description evolution.

3.4.2 Control unit

A control unit is, basically, any computer used in the I&C system and connected to one or more control networks. This is shown on Figure 3-16.
Because it is a physical part of a plant, the control unit inherits its properties, such as a part number and an optional description. In addition, one or more host names could be specified for a control unit.

Plant system I&C can contain three types of control units which are a slow controller, a fast controller and a plant system host. CODAC system adds a fourth type, which is a CODAC server (see section 3.5.8).

### 3.4.3 Controller

A controller is a control unit which is capable of interfacing and processing signals coming to and from the plant. The structure of the controller is shown on Figure 3-17.

A controller is normally equipped with one or more input / output (I/O) boards. This generic definition is not used in the plant system I&C by itself, but serves as a basis for definition of slow and fast controllers (see sections 3.4.6 and 3.4.7).

### 3.4.4 Input / output board

An I/O board is an interface from computer systems to the plant signals. Its structure is shown on Figure 3-18.
I/O boards are also treated like the plant parts and thus have part numbers assigned to them. Each board also has a type identifier, which points into the list of supported I/O boards (see section 3.5.2). This identifier helps to determine the correct driver to work with this board.

Each I/O board has one or more channels, which are described in the next section.

Note: these definitions of the I/O boards should not be confused with the definitions in section 3.5.2. The latter define characteristics of a class of boards as a whole, while the former define properties of individual boards.

### 3.4.5 Input / output board channel

An I/O board channel descriptor is a set of settings to configure this channel properly. Its structure is shown on Figure 3-19.

Each channel has a name and, optionally, a signal associated with it. If the channel allows multiple voltage settings, the one that is required is recorded here.

Other channel settings are under definition and will be added later on.

### 3.4.6 Slow controller

A slow controller is a controller used to implement slow control loops. In the ITER I&C systems slow controllers are implemented with the help of Siemens Simatic PLCs (see [RD1]). The structure of the slow controller is the same as for a generic controller; it is shown on Figure 3-20.
3.4.7 Fast controller

A fast controller is a Linux-based computer with one or many EPICS input / output controllers running. Its structure is shown on Figure 3-21.

Figure 3-21: Fast controller

Here the OS settings represent the Linux settings (see section 3.5.11). The EPICS settings are described in section 3.4.9.

3.4.8 Plant system host

The plant system host (PSH) is again a Linux-based computer capable of running EPICS and other software.
for plant system I&C configuration and control. It is not equipped with the I/O boards. The definition of the plant system host is shown on Figure 3-22.

**3.4.9 EPICS input / output controller**

The EPICS SCADA software is a key element of the ITER I&C systems. Instances of EPICS running on the plant system I&C control units are called EPICS input / output controllers (IOCs). The data structure of a typical IOC is described in [RD10] and is shown on Figure 3-23.

The IOC consists of the reference to the database model description (the DBD file, see section 3.5.9), as well as a set of channel access parameters, list of drivers and EPICS modules used. The core of the IOC is
an EPICS database (the DB files), which defines the data acquisition and processing logic. The IOC can have a name in order to identify it in the I&C control unit. A version of its configuration data can equally be specified.

3.4.10 EPICS database

The EPICS database is a set of records of a well defined structure. It is shown on Figure 3-24.

![Figure 3-24: EPICS database](image)

The database can have a name (normally the name of a corresponding DB file), as well as a version number.

3.4.11 EPICS records

The EPICS record is the implementation of an I&C variable in the EPICS software. The structure of a record is shown on Figure 3-25.

![Figure 3-25: EPICS record](image)

Each record has a name and a type and consists of a number of fields, which define properties of the variable. Every field has a name and a value. The available record types and their possible fields are defined in the EPICS database model (DBD, see section 3.5.9).

3.4.12 EPICS channel access settings

Each IOC setup can adjust parameters of its communication protocol, Channel Access (CA). Available settings are shown on Figure 3-26.
The explanation of these settings is given in [RD11]; most of them have reasonable defaults. When running several IOCs on the same control unit, at least the port number has to be corrected for all but one IOC in order to avoid port binding conflicts on startup.

3.4.13 EPICS drivers

The records bound to hardware are normally supported by a set of EPICS drivers. The drivers currently in use in the ITER I&C are shown on Figure 3-27.

![Figure 3-27: Currently supported EPICS drivers](image)

The “asyn” and “NI6259” drivers are used for the fast controller I/O boards. Their settings will be detailed in the future releases of the schema.

The “S7PLC” driver is used to communicate with the Siemens PLCs.

3.4.14 EPICS S7PLC driver

The PLC driver requires specific configuration before use. Its structure is shown on Figure 3-28.

![Figure 3-28: S7PLC driver settings](image)

The driver is configured with one or more communication channels. Each communication channel represents a connection to a PLC. The settings of the communication channels are documented in [RD12].
3.4.15 EPICS modules

The EPICS modules represent add-ons to the basic IOC functionality. The incomplete list of supported modules is shown on Figure 3-29.

EPICS modules include “autoSave” (support for variable persistency), sequencer (support for state machines) and logger (logging facilities). For the “autoSave” there is a possibility to specify names of the records to be saved on persistent media.

The list of modules and their settings will be elaborated in future versions of the schema.

3.4.16 Variables

Each plant system I&C possesses a number of variables which reflect processes in the plant. The structure of a variable is shown on Figure 3-30.
## 3.4.17 Variable’s data type

The variable’s value section specifies the type of data held in this variable and, optionally, the value itself (for static variables). The list of data types is not implementation specific; it should be mapped to the particular software types (e.g., EPICS) during the data handling. The list of currently defined types is shown on Figure 3-31.

<table>
<thead>
<tr>
<th>psic:direction</th>
<th>“State” (read from the CODAC point of view) or “configuration” (written from the CODAC point of view)</th>
</tr>
</thead>
<tbody>
<tr>
<td>psic:kind</td>
<td>Discrete (two- or multi-state), continuous or string</td>
</tr>
<tr>
<td>psic:value</td>
<td>Variable data type and (optionally) its default value</td>
</tr>
<tr>
<td>psic:implementsSignal</td>
<td>Name of the signal in the case the variable implements it</td>
</tr>
<tr>
<td>psic:raisesAlarm</td>
<td>True if the variable can raise an alarm (false by default)</td>
</tr>
<tr>
<td>psic:archivingInfo</td>
<td>Configuration for the archiver (if needed)</td>
</tr>
<tr>
<td>psic:storage</td>
<td>Storage specifier (static, persistent or volatile). Volatile is a default</td>
</tr>
<tr>
<td>psic:deployedOn</td>
<td>Reference to the control unit where this variable will be deployed</td>
</tr>
</tbody>
</table>

**Figure 3-30: A structured view of a variable**

The variable has a name in accordance with the variable naming convention (see section 3.2.6) and an optional description. The variable’s direction tells whether the variable presents status of the plant (“state”), or drives its state (“configuration”). The variable’s kind represents the nature of the variable; it could be one of the “discrete two-state”, “discrete multi-state”, “continuous”, or “string”. The “value” field defines the variable’s data type and, optionally, a default value (used for static variables). In the case when the variable implements some signal, its name is recorded in the corresponding field. There are also means to tell whether the variable may raise an alarm, or to be archived. The storage specifier tells how exactly the variable’s value should be treated during a restart of the control unit it resides on. Finally, the deployment information records exactly where this variable is located in a real system.
A generic data type

dataType

Collection of the elementary data types
data:elementaryDataTypes

- Boolean value
  - psic:bool
- One byte signed integer
  - psic:int8
- One byte unsigned integer
  - psic:uint8
- Two bytes signed integer
  - psic:int16
- Two bytes unsigned integer
  - psic:uint16
- Four bytes signed integer
  - psic:int32
- Four bytes unsigned integer
  - psic:uint32
- Four bytes real number
  - psic:float32
- Eight bytes real number
  - psic:float64
- The UTF-8 string type (unlimited length)
  - psic:string
- Restricted string suitable for identifiers (ASCII, 40 chars max)
  - psic:nameString

Collection of the derived data types
data:derivedDataTypes

1-dimensional array
- psic:array
- Array length
  - psic:length
- Any sort of enumerated entities
  - psic:enum
- Enumeration with fixed values (powers of two)
  - psic:bitPattern

An array data type represents a simple one-dimensional array. It is shown on Figure 3-32.

Array length
Array's data

Array element

3.4.18 Array data type

The list of data types is very common; it includes types to represent numbers and strings. One restricted string type ("nameString") is included to represent name identifiers in the I&C controllers. Several compound types – an array, an enumeration and a bit pattern – are provided as well. They are described in the following sections. There could be more data types introduced in future releases of the schema.
Each item in the array represents a value of one of the elementary types defined in the schema.

### 3.4.19 Enumeration data type

An enumeration data type defines a list of name-value pairs. Each value may be of one of the elementary types defined in the schema. This is shown on Figure 3-33.

![Figure 3-33: Enumeration data type](image)

Each name-value pair can have an indication of being selected, which is used to represent static discrete values in the self-description data. An enumeration can also have an exclusiveness flag, which indicates that only one of the defined states can be selected at any given time.

### 3.4.20 Bit pattern data type

A bit pattern is a particular case of an enumeration in which values are fixed as powers of two. In this case they are not provided in the schema but determined on the basis of order of the name labels in the collection, as shown on Figure 3-34.

![Figure 3-34: Bit pattern data type](image)

The maximum number of names in a bit pattern is fixed at 16. The rest of the properties are the same as for the enumeration type.

### 3.4.21 Variable’s archiving info

An archiving info section describes parameters for the archiver, as shown on Figure 3-35.
Configuration for the archiver (if needed)

Sampling frequency (Hz)

The sampling mode could be one of “scan” or “monitor” and the sampling frequency is specified in Hz. If the entire section is missing from the variable’s description, the variable does not need to be archived.

3.4.22 Variable deployment information

This section describes variable mapping to the physical control unit and its software. It is shown on Figure 3-36.

The first part is a reference to a particular control unit (an I&C component), which is done via its part number. The type of the unit (a slow or fast controller, a PSH or a CODAC server) determines further details. For the slow controllers, which are PLC based, the name of the communication channel (see section 3.4.14) and the variable’s slot in it are specified. For other types of control units, which are all EPICS-based, the name of the IOC and the appropriate name of the record are specified.

3.4.23 Plant control system

It can be simply stated that the plant conventional control system is composed of slow and fast controllers, as shown on Figure 3-37.

There could be an arbitrary number of slow or fast controllers in the plant control system.
3.5 CODAC data structures relevant for plant systems

Sometimes data structures may be shared by central I&C systems and plant systems. In some other cases CODAC provides reference information, which is useful for the plant system design and operation. Several such data structures are described in this chapter.

3.5.1 Catalogs

A catalog, in a general sense, is a collection of items of the same origin. The PCDH collection includes several catalogs, such as the catalogs of slow and fast controllers. To store this information in a structured fashion, the following model is defined (Figure 3-38).

Figure 3-38: Catalog structure

The catalog has a name, a version, and, optionally, a description and a name of its maintainer. Catalog items
include reference information, such as reference number, link to a data sheet, a list of standards the item adheres to, or a picture.

3.5.2 Supported I/O board descriptions

CODAC provides detailed descriptions of the officially supported I/O boards, or, in other words, descriptions of the boards of a certain type. They can be used by the SCADA software in order to know what the possible settings or limits in operating this particular board are. The descriptor of a supported board is shown below (Figure 3-39).

![Figure 3-39: Description of a supported I/O board](image)

The I/O board class has a name, a specification of its usage (in a slow or fast data processing chain), the names of the corresponding Linux and EPICS drivers and a TTT code (see section 3.2.2) for this type of board. The catalog reference holds the information about commercial aspects as well as documentation, as described in the previous section. The technical data part goes into the technical details of the board.

3.5.3 I/O board technical data

The I/O board technical information is an exhaustive set of information about board parameters. It is shown on Figure 3-40.
The technical information includes the controller’s interface bus type, the timing resolution, the scanning rate and amount of memory available, the lists of board channels, programmable interfaces and counters. Also included are some design and / or service-oriented information such as physical and power requirements, as well as pinouts of the board connectors.

3.5.4 I/O board channel parameters

The I/O board channels and programmable interfaces have a similar descriptor which is shown on Figure 3-41.
Channel description lists typical settings which can be set on an individual channel. They include the size of the memory buffer available for digitizing as well as the list of programmable voltage levels. Also included is the information on the channel mapping to the pin in the board’s connector.

3.5.5 I/O board physical requirements

Physical requirements give information about mechanical and environmental characteristics of the board. They are shown on Figure 3-42.
3.5.6 I/O board power requirements

Power requirements give information about power demands of the board. They are shown on Figure 3-43.

3.5.7 Connectors

Connectors are used not only in the I/O boards, but virtually everywhere. Knowing connectors’ pinouts is essential for the construction and service of plant systems. The data description of a connector is shown on Figure 3-44.

Figure 3-42: I/O board physical requirements

The fields in the description are self-explanatory.

Figure 3-43: I/O board power requirements

These properties represent a list of power bus lines with the indication of voltage and corresponding current draw from the line.
Each connector descriptor provides for a particular type (an industry-recognized name), a name (if any) and, optionally, a picture. Each pin has information on its name, description, and also an indication if the pin is connected or shared by several channels.

3.5.8 CODAC servers

As it was stated previously (see section 3.4.2), the CODAC system has its own type of control unit, which is a CODAC server. This type of unit may be of interest for the plant system I&C designers, because they may designate part of the plant system I&C data processing to be done on the CODAC side, in which case the deployment unit for their I&C variables will be one of the CODAC servers.

As of today, the data structure of the CODAC server is not very different to the plant system host; it is shown on Figure 3-45.
3.5.9 EPICS database model

The EPICS database model defines what kind of records can exist in the EPICS database and what properties (fields) they are allowed to have. Normally, this model should be the same across the entire installation; however, EPICS allows different IOCs to have different database models (DBD files).

The structure of the database model is shown on Figure 3-46.
The detailed description of the EPICS database model is available in [RD10].
3.5.10 EPICS predefined records

In order to have a uniform approach to programming of the EPICS IOCs, CODAC imposes a set of predefined EPICS records to be used to perform routine tasks, such as connectivity or health monitoring. The data structure of these predefined records is shown on Figure 3-47.

![Figure 3-47: EPICS predefined records](image)

The predefined record is a regular EPICS record supplemented with a qualifier of the designated control unit type (combinations are possible). It is also possible to indicate that the record has to be present on every IOC in the system.

3.5.11 Linux settings

Linux is a base operating system (OS) for I&C control units. Normally, its configuration should be uniform across the system as much as possible; however, deviations are inevitable and tolerable. To record these configurations, the following model is used (Figure 3-48).

![Figure 3-48: Linux description](image)

The essential part is the OS identification (distribution version, hardware platform, etc). The list of installed drivers is important for the controllers and may vary depending on the control unit type.

This model is far from being complete and will be extended in future versions of the schema.

3.5.12 Linux drivers

Sometimes having an EPICS-level driver is not enough, and support from the OS itself is required (e.g., in the case of PCI-like devices). This is where the Linux drivers are used. Very often they require specific configuration in order to function properly. This information is recorded as shown on Figure 3-49.
The driver descriptor holds identification information as well as a list of special nodes created in the /dev file system, and information on used resources (PCI bus, interrupts, DMA). Some documentation, such as information about the implemented system calls, is also provided.

### 3.6 Static data sets

Apart from the data model, the self-description schema distribution includes quasi-static project data which are important for the I&C design and implementation. They are called “quasi-static” because they are updated quite slowly (e.g., once or twice per year) and are supposed to be frozen by the end of the design phase. The data are delivered in the form of XML files compliant with the self-description data model. There is ongoing work to make the same information available from the central ITER databases.

At the time of writing, the list of data includes:

<table>
<thead>
<tr>
<th>№</th>
<th>File name</th>
<th>Description</th>
<th>See section</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PBS.xml</td>
<td>Full PBS tree</td>
<td>3.2.1</td>
</tr>
<tr>
<td>2</td>
<td>FBS.xml</td>
<td>Full FBS tree</td>
<td>3.2.4</td>
</tr>
<tr>
<td>3</td>
<td>PlantParts.xml</td>
<td>The list of the ITER plant parts defined so far</td>
<td>3.3.2</td>
</tr>
<tr>
<td>4</td>
<td>TTT.xml</td>
<td>Registry of the TTT codes</td>
<td>3.2.2</td>
</tr>
<tr>
<td>5</td>
<td>AAAA.xml</td>
<td>Registry of the AAAA codes</td>
<td>3.2.3</td>
</tr>
<tr>
<td>6</td>
<td>SS.xml</td>
<td>Registry of the SS codes</td>
<td>3.2.3</td>
</tr>
</tbody>
</table>
Table 3-1: List of static project data supplied as a part of the SDD

### 3.7 Examples

#### 3.7.1 Plant system with a slow controller

As an example take a simple plant system (51 – ICH) which consists of three components (parts):

1. a power supply (51HV11-EW-0001) with an on/off actuator and a corresponding sensor;
2. a slow controller (PLC) (51HV1B-PCS-0001) used to monitor and control this power supply;
3. a plant system host (51CICT-PSH-0001) used for software configuration and service functions of this plant system.

The actuator and sensor of a power supply form two I&C signals (51HV11-EW-0001:CZ-CRC for an actuator and 51HV11-EW-0001:CY-CRC for a sensor) which have their process variables allocated on the slow controller 51HV1B-PCS-0001. The type of these variables is a discrete two state variable represented by a bit pattern with the ON and OFF states. The state of the power supply, read from the sensor, is saved to an archive 10 times per second; and its OFF state raises a minor alarm in the CODAC system.

The slow controller is configured to have one communication channel hosting the two process variables.

The PSH hosts one variable 51-ICH:COUNTDOWN which does not represent any particular equipment but serves as a counter for the pulse sequence. It has a type of 4 bytes floating point value.

The XML files reflecting such a configuration are shown below.

**Listing 1: PlantSystem.xml**

```xml
<plantSystem ID="51" name="ICH & CD system" version="20110106"
  <parts>
    <part name="51HV11-EW-0001" description="Power supply"/>
    <part name="51HV1B-PCS-0001" description="Controller"/>
    <part name="51CICT-PSH-0001" description="Plant System Host"/>
  </parts>
  <signals>
    <signal name="51HV11-EW-0001:CY-CRC" description="Power supply sensor">
      <source>51HV11-EW-0001</source>
      <direction>input</direction>
      <nature>digital</nature>
      <IandCType>state</IandCType>
      <archive>true</archive>
      <importance>regular</importance>
      <safetyClass>non-SIC</safetyClass>
      <displayName>PS1</displayName>
    </signal>
    <signal name="51HV11-EW-0001:CZ-CRC" description="Power supply actuator">
      <source>51HV11-EW-0001</source>
      <direction>output</direction>
      <nature>digital</nature>
      <IandCType>command</IandCType>
      <archive>true</archive>
      <importance>regular</importance>
      <safetyClass>non-SIC</safetyClass>
      <displayName>PS1 OR</displayName>
      <operatorCanModify>true</operatorCanModify>
    </signal>
  </signals>
</plantSystem>
```

**Listing 2: PlantSystemIandC.xml**

```xml
<signal name="51HV11-EW-0001:CY-CRC" description="Power supply sensor">
  <source>51HV11-EW-0001</source>
  <direction>input</direction>
  <nature>digital</nature>
  <IandCType>state</IandCType>
  <archive>true</archive>
  <importance>regular</importance>
  <safetyClass>non-SIC</safetyClass>
  <displayName>PS1</displayName>
</signal>
<signal name="51HV11-EW-0001:CZ-CRC" description="Power supply actuator">
  <source>51HV11-EW-0001</source>
  <direction>output</direction>
  <nature>digital</nature>
  <IandCType>command</IandCType>
  <archive>true</archive>
  <importance>regular</importance>
  <safetyClass>non-SIC</safetyClass>
  <displayName>PS1 OR</displayName>
  <operatorCanModify>true</operatorCanModify>
</signal>
```

Note: the ICH system identifiers are used here as an example. The description above does not represent a set-up of an actual ITER ICH system.