

Requirements management support for the ITER Plasma Control System in view of first plasma operations

M. Cinque^{a,b}, F. Fucci^c, G. De Tommasi^{a,b}, P.C. de Vries^d, I. Gomez^f, B. Sammuli^e, J. Snipes^d,
W. Treutterer^f, M. Walker^e, A. Winter^g, L. Zabeo^d

^a*Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione,
Università degli Studi di Napoli Federico II, via Claudio 21, 80125, Napoli, Italy*

^b*Consorzio CREATE, via Claudio 21, 80125, Napoli, Italy*

^c*Critiware, c/o Centro Meridiana, Torre Antares, 80013, Casalnuovo di Napoli, Italy*

^d*ITER Organization, Route de Vinon sur Verdon, CS 90 046, 13067 St Paul Lez Durance Cedex, France*

^e*General Atomics PO Box 85608, San Diego, California 92186-5608, USA*

^f*Max Planck Institute for Plasma Physics Boltzmannstr 2, 85748 Garching, Germany*

^g*Max Planck Institute for Plasma Physics, Teilinstitut Greifswald, Wendelsteinstrasse 1, Greifswald, Germany*

Abstract

In a complex project as the ITER one, it is essential to rely on a *system-engineering oriented* approach for requirements management, to both track the development of the various systems and to assess their compliance with the requirements set expressed at each operational phase. Such an approach is currently adopted for the ITER Plasma Control System (PCS), where a database implemented using the Enterprise Architect tool has been setup to act as the central collaboration tool for all the PCS system design.

This paper describes the current structure of the PCS DB, and presents examples on the interfacing of the PCS and diagnostics, relevant for first plasma operations.

Keywords: System Engineering, ITER, Plasma Control System

1. Introduction

The Plasma Control System (PCS, [1] and [2]) is one of the main ITER systems. It is in charge of running the plasma discharge, by receiving data from the real-time diagnostics, and by computing the commands to be processed by various plant systems to act on the plasma (e.g., the power supplies of the poloidal field coil circuits, the gas injection valves, and the additional heating systems). Moreover, as in all existing tokamaks, the use of the PCS is also envisaged for the commissioning of single diagnostics and plant systems, as well as for integrated commissioning, with and without plasma [3].

To this aim, the PCS will implement several functions, which will be not limited to control algorithms, but they will also include support functions that process the inputs acquired from different diagnostics, and exception handling [4]. There are many different types of requirements for the ITER PCS

(functional, architectural, operational, etc.), which come from a heterogeneous set of sources. Moreover, the design of some plant systems the PCS has to interface is not yet finalized, and the way ITER will be operated is also under development.

In such a complex context, it is essential to rely on a system-engineering oriented approach for requirements management, to both track the development of the PCS and to assess the compliance of the product with the requirements set expressed at each operational phase (first plasma operation, pre-fusion plasma operations, etc.). The current ITER choice is to archive all the models and artifacts produced in the PCS design activities through the use of a database, called the PCS Database (PCS DB), which acts as the central collaboration tool for all the PCS system design, and it is deployed using the Enterprise Architect® (EA) system design software, and SysML [5] as modeling language. EA includes several tools for collaborative work (such as, package locking, baselining, and team review) and facilitates the input of requirements as text-like

Email address: macinque@unina.it (M. Cinque)

documents with the Specification Manager view.

The PCS DB is devised to be flexible enough not only to trace system requirements to higher level ones (i.e., stakeholder requirements and others coming from ITER project high level documents), but also to map PCS functions to the architectural components. The PCS DB is also envisaged to document the set of all the PCS functionalities, by linking system requirements to both their assessments (which will be run in the PCS simulation environment [6]), and to the documents that describe the corresponding commissioning procedures. Hence, the PCS DB will be the tool to trace all the information relevant to the PCS design, assessment, implementation, and commissioning.

In this work, the overall methodology for the PCS requirements management process, and the current structure of the PCS DB is described. In particular, the paper is structured as follows. Section 2 describes the PCS database methodology and structure. Section 3 provides some examples of the current contents of the DB, focusing on the use of the DB for linking the PCS design with interface requirements from diagnostics, while Section 4 ends the paper with final remarks.

2. The PCS Database

2.1. Methodology

In the PCS DB, the information about system design and testing is organized according to the classical V-Model, depicted in Fig. 1. Each step of the requirements management process (descending phases, left part of the “V”) can provide support information to one of the steps of the V&V process (ascending phases, right part of the “V”). For instance, stakeholder requirements define the results

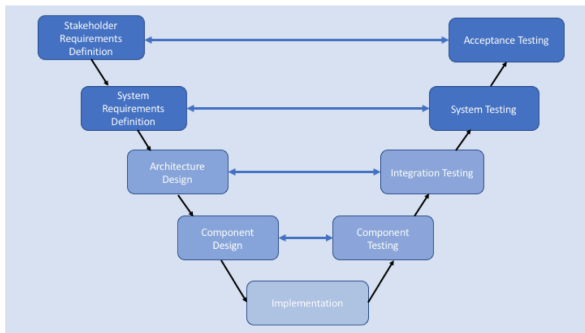


Figure 1: The V-Model

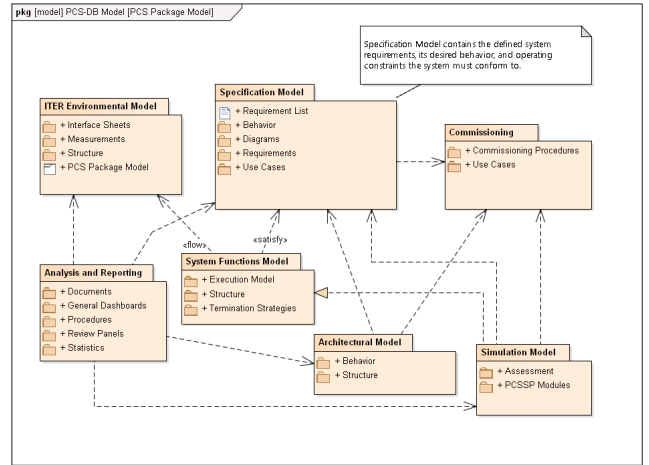


Figure 2: The package diagram of the PCS model

that users expect from the system; hence, they can be used to validate the product in the final acceptance test. Similarly, system requirements define what the system must do, and so they can be used to verify the system during the system test (as part of commissioning procedures).

The approach is defined with reference to the Enterprise Architect (EA) [7] software for the automated management of the requirement database, and it is articulated in the following steps: (i) define the entire PCSDB structure (see 2.2); (ii) populate the database; (iii) introduce traceability (see 2.3); (iv) manage change, iteratively; and (v) generate reports and views.

2.2. Structure

SysML provides a formalism to structure the model in hierarchical entities, named package diagram. A package is a model artifact that acts as a container for other artifacts stored in the PCS DB. By *artifact* we generally refer to a modelling element stored in the PCS DB.

Fig. 2 shows the implementation of the PCS package diagram and the dependency links between packages, which aim at capturing the logical connections among all the components involved in the design. The rationale behind the subdivision of packages relates to the V-model so that root-level packages are containers of artifacts produced at the various stage of the design flow.

The package structure of the PCS model includes six packages: (a) *Specification Model*, containing all the artifacts concerning the description of the re-

109 requirements and the use cases of the Plasma Control System; (b) *ITER Environmental Model* that includes all the elements concerning the interfacing of the PCS with other plant systems, such as, interface sheets and measurements; (c) *System Functions Model*, including the functional decomposition of the system in control functions, support functions, and the related execution model; system functions are designed to satisfy the requirements in the specification and to be compliant with the interface sheets; (d) *Architectural Model*, containing all the artifacts concerning the design of the PCS architecture, including SysML behavioral and the structural diagrams; (e) *Simulation Model*, which includes all the artifacts concerning the simulation of PCS control algorithms; (f) *Commissioning*, including all the artifacts about the PCS commissioning use cases and procedures; and (g) *Analysis and Reporting*, which provides a summarized view of the PCSDB content status and and scripts to generate documentation from the model. Packages (a)-(e) contain design elements related to the left side of the V design process, whereas, package (f) includes elements relevant for the right side of the V.

115 Modeled artifacts can link, as sources, external documents stored in the ITER document management system. To keep the coherence between artifacts and documents, we always refer to a precise version of the a document. If the document is updated, relevant changes have to be manually reflected in the PCSDB.

140 2.3. Traceability

141 Traceability relationships (i.e., basically links between modelling elements) are the bulk of the whole design process and are stored in the PCS DB. Initially, the collected artifacts do not have explicit relationships. Therefore, it is required, starting from requirements, to define links both vertically (i.e., from the requirement up to the high level requirement(s) it refers to, or down to the design artifact that satisfies the requirement), and horizontally (i.e., within the same layer).

151 A traceability link identifies the dependency relationship between requirements. Changes on those might have mutual effects to be traced automatically. Once requirement statements have been imported, links can be easily created in EA, graphically or using traceability matrices. The presence of links permits to perform several automated types of analysis on the database, such as:

159 **Impact Analysis:** useful to assess the impact of changes of a requirement, e.g., counting how many requirements (and consequently system functions and components) need to be changed if a requirement changes.

164 **Derivation Analysis:** to derive the high level requirements that give rise to a functionality. This is useful to justify the presence of low level artifacts, or the absence of a requirement.

169 **Coverage Analysis:** designers desire to constantly assess their models. Coverage analysis helps them in understanding if their model (architectural model and component model) is correctly linked with all the requirements and use cases, to help them justify their choices.

174 2.4. Staged approach

175 To conform to the ITER staged approach, the design of the PCS, and therefore the elements in the PCSDB, shall be properly distributed over the 4 stages (First Plasma - FP, Pre-Fusion Plasma Operation 1 - PFPO1, Pre-Fusion Plasma Operation 2 - PFPO2, and Fusion Plasma Operation - FPO). To each of the PCSDB components requirement, system block, use case etc., an identifier (field in the EA terminology) *phase* is added. In this way it will be possible, during the design lifecycle of the PCS, to output from the PCS DB only the artifacts relevant for a given operation phase. For instance, considering requirements, we can group them depending on the phase, and the checking of their fulfillment can be done with respect to the current design stage.

191 3. Example: ITER Environmental Model

192 PCS functions depend on a large number of input parameters from the plasma and plant systems to perform their actions. Diagnostics and operational instruments are the main source for the relevant input information to the PCS.

198 Generally a single plasma parameter is the combination of several diagnostics that might also be distributed and available differently accordingly to the staged approach. For example, during First Plasma operation, the *line-averaged electron density* is provided only by a reduced version (only one line of sight instead of two) of the Density Interferometer Polarimeter (DIP). From PFPO1 the DIP will be fully functional and additional diagnostics will be allocated (the Toroidal Interferometer/Polarimeter

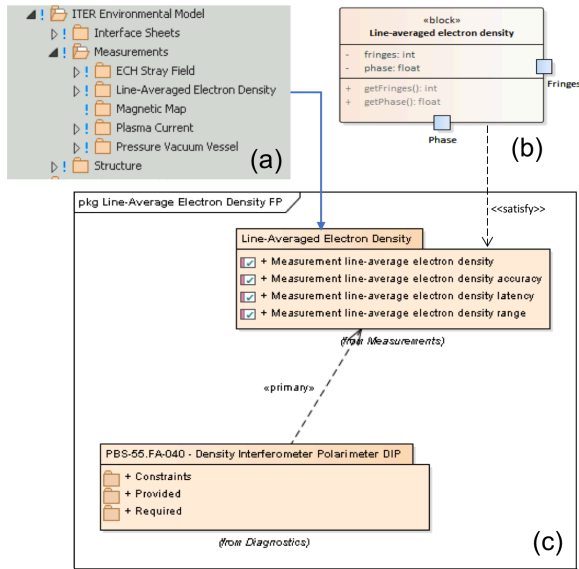


Figure 3: ITER Environmental Model Example: (a) list of measurements added for first plasma operation; (b) structure of the line-averaged electron density measurement; (c) line-averaged electron density measurement requirements and related diagnostics at First Plasma.

- TIP, and the Core Thomson Scattering diagnostic) aiming at providing a more robust measurement.

To this aim, the ITER Environmental Model includes a *Measurement* package, which decouples parameters and diagnostics, and contains requirements about the accuracy, the time resolution and constraints about the real-time measurements provided from the diagnostics to the PCS. Measurements are associated with all the interface sheets already present in the PCS DB.

Fig. 3.(a) shows the list of the measurements added for First Plasma operation. Fig. 3.(c) provides the details of the line-averaged electron density package, with associated measurement requirements, and how they link with diagnostics at First Plasma operations (at PFPO1 the TIP and the Core Thomson Scattering would be added; this way, it is possible to navigate the design depending on the operation phase). Finally, measurement blocks provide information to the architecture and to the PCS functions about the values received from diagnostics. For instance, for the line-averaged electron density block, depicted in Fig. 3.(b), it contains the fringes and the phase properties; the phase is a value in radians while the fringes is an integer value. The design specifies that the last updated values

of the fringes and phase properties are accessible via the `getFringes()` and `getPhase()` operations. These properties can then be used for the modeling of control functions and the architecture.

Traceability links help to perform impact or coverage analysis, e.g., to check how many control functions or architectural elements are affected by a change in a measurement requirement, in terms of accuracy or time resolution.

4. Conclusions

The design of a complex system as the ITER Plasma Control System, involving a large team with different expertise and perspectives, requires a systematic way to organize and trace the design information, following system engineering practices. This paper presented The PCS DB tool, used at ITER to allow the collaborative design of PCS control functions and architecture, and to trace design artifacts to requirements, commissioning procedures, and other plants systems, e.g., diagnostics, as shown in the provided example.

Acknowledgements

ITER is a Nuclear Facility INB-174. The views and opinions expressed herein do not necessarily reflect those of the ITER Organization. This work is partially supported by the RT-CASE project, funded by the Dept. of Electrical Engineering and Information Technology of the University of Naples Federico II, Italy.

References

- [1] J. Snipes, et al., Overview of the preliminary design of the ITER plasma control system, *Nucl. Fus.* 57 (2017) 125001.
- [2] P. de Vries, et al., Preparing the Plasma Control System final design for ITER first plasma operations, *Fus. Eng. Des.* 129 (2018) 334–340.
- [3] A. Neto, et al., From use cases of the Joint European Torus towards integrated commissioning requirements of the ITER tokamak, *Fus. Eng. Des.* 96 (2015) 672–675.
- [4] G. Raupp, et al., Preliminary exception handling analysis for the ITER plasma control system, *Fus. Eng. De.* 123 (2017) 541–545.
- [5] Systems modeling language, <http://sysml.org> - Accessed: 2018-07-28 (2018).
- [6] M. Walker, et al., The ITER plasma control system simulation platform, *Fus. Eng. Des.* 96 (2015) 716–719.
- [7] Sparx Systems Enterprise Architect, <http://sparxsystems.com/products/ea/> - Accessed: 2018-07-28 (2018).