From a physics discharge program to device control – linking the scientific and technical world at Wendelstein 7-X

Heike Riemann[a,*], Torsten Bluhm[a], Peter Heimann[b], Christine Hennig[a], Georg Kühner[a], Hugo Kroiss[b], Heike Laqua[a], Marc Lewerentz[a], Josef Maier[b], Jörg Schacht[a], Anett Spring[a], Andreas Werner[a], Manfred Zilker[b]

[a] *Max-Planck-Institut für Plasmaphysik, EURATOM Association, Teilinstitut Greifswald, Wendelsteinstraße 1, D-17491 Greifswald, Germany*

[b] *Max-Planck-Institut für Plasmaphysik, EURATOM Association, Boltzmannstraße 2, D-85748 Garching, Germany*

[*] *Corresponding author; Tel.: +49-3834-88-2225; Fax: +49-3834-88-2509; E-mail address: Heike.Riemann@ipp.mpg.de*

Wendelstein 7-X is designed to be a steady state experiment. One discharge may last up to 30 min and will be divided into many segments (elementary experimental program parts). Physics programs need to be pre-planned and technical parameters need to be adjusted in the long run. The technical parameters are a complete set of parameters for all participating components, which describe their behavior within a segment.

To enable the diagnosticians to focus on physical experiment parameters only, an abstraction layer above the technical parameters has been developed, the so-called high level parameters. The high level parameters specify a subset of the technical parameters on a physics level and are therefore used to hide technical details. During ordinary program planning, the diagnosticians only see and work with these physics parameters. Having finished editing, the high level parameters will be mapped by a user defined transformation function onto technical parameters. For all technical parameters which are not defined by the high level parameters, the default values will be taken from one particular predefined pattern. Hence a complete segment is defined by a set of high level parameters including a pattern.

In comparison to shot-based and other existing experiments, this is an entirely new, scientifically oriented approach to plan experiments. The motivation and concept of the high level parameters will be discussed. The successful implementation of this concept at the W7-X CoDaC prototype – the small stellarator experiment WEGA – will be presented.

Keywords: high level parameter, abstraction layer, scientifically-oriented, physics parameters, planning experiments, long discharge duration

1. Introduction

Wendelstein 7-X is designed for continuous operation with a discharge duration of about 30 min. Since different physics objectives are to be investigated within this time frame, one discharge is divided into many elementary program parts which are responsible for experiment control during a certain period of time, the so-called segments. Each segment describes the state and behavior of the technical and diagnostic components during the execution of the segment. A set of chained segments is called a scenario, and a few scenarios make up the whole experimental program. [1]

At W7-X about 60 diagnostic systems with many different hardware devices will be implemented. The software representation of an individual device or system component is called a module. In addition, modules may also represent other components, processes etc. (see Refs. [2,3] for more details). Each module has a set of parameters which describes its behavior within a segment. [4] A complete set of parameters for all participating modules within one segment is called a segment description. In order to execute several physics objectives, parameter changes are necessary. The large number of parameters and the overall complexity require automatical interaction with the hardware during a discharge, while manual interaction should be avoided. Therefore the configuration of components needs to be planned well before a discharge and segment parameters need to be defined in advance.

All information about system configuration and experiment program specification is kept in the W7-X configuration database [1].

The system component parameters are technical parameters. However, physicists would prefer a more physics-oriented point of view onto the segment and discharge description. In order to focus on physics parameters and thereby filtering too many details, an abstraction layer above the technical parameters has been developed, the so-called high level parameters.

2. Purpose

Segment descriptions are basically lists of parameter sets for the system components of one segment. The level of detail in a segment descriptor is naturally very high. Each parameter down to the lowest level in a module has to be specified and declared. This is in principle necessary but not what experimentalists want to care about. They usually aim at specifying the behavior and state of the plasma and machine in terms of physics. [1] Moreover, many parameters have to be set, but do not change during experiments. It appears useful to hide them from diagnosticians. Therefore the high level parameters have been introduced, which specify a subset and abstraction of technical parameters on a higher level, see figure 1.

For tests of simple systems or subsystems, the requirement for abstraction is not very high. In this case, the high level parameter values can be used directly for the low level parameter values and are therefore basically used as a filter function for low level parameters. On the other hand, the level of abstraction may be high and one single high level parameter may determine several low level parameters, leading to complex mapping procedures.

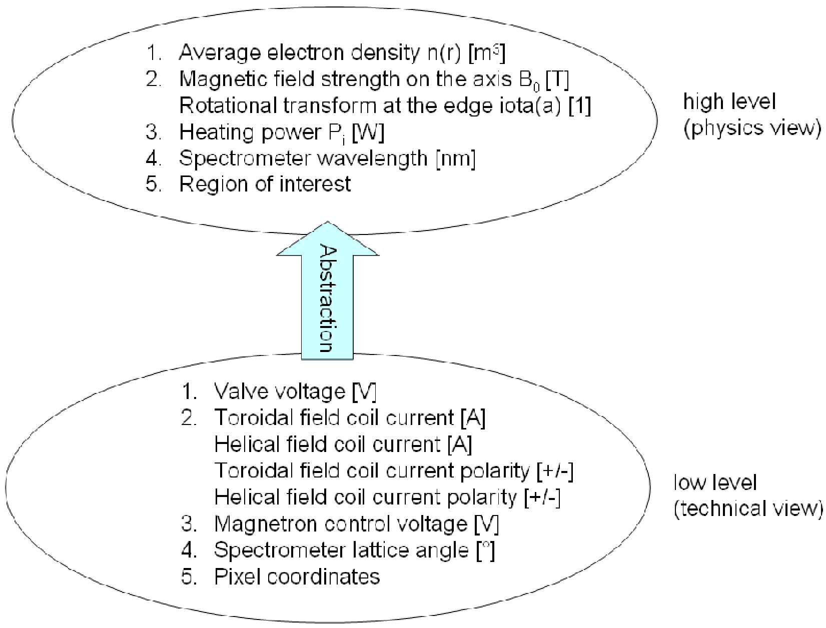

Fig. 1: Mapping low level parameters onto high level parameters, e.g. (3) a magnetron control voltage corresponds to a particular heating power.

During ordinary program planning, experimentalists only see and work with these physics parameters. After having edited the high level parameters, they will be mapped onto low level parameters by an adequate function.

Using high level parameters, the information presented to the user is reduced considerably, physicists can focus on physics parameters and thus an appropriate and convenient handling of the existing complexity is achieved.

3. Concept and Implementation

The 'High level parameter' concept is implemented in Software written in Java and represented by the classes `FlexibleParamDesc` and `FlexibleParam`.

Since many high level parameters exist for each segment, they are structured in order to reduce complexity. The structure determines the view on the high level parameters and is used to construct a graphical user interface for editing high level parameters. Structuring is done by a hierarchy of sequences, choices and parameters. A sequence is a collection of high level parameters and makes sure that parameters are clearly arranged by context or other conditions. A choice provides alternatives for defining high level parameters. For example, the implementation of the magnetic field can be done by either direct definition of the coil currents or defining the magnetic field values, which correspond to certain coil currents. According to the selected choice, the corresponding parameters will be available. Parameters of choices being unnecessary in this context are hidden from the user. Arranging information in groups by context and showing only relevant parameters finally leads to a reduction of complexity.

The hierarchical structure of high level parameter descriptors (`FlexibleParamDesc`), being similar to the composite design pattern of [5], as well as the corresponding parameters (`FlexibleParam`) are shown in figure 2. Each high level parameter descriptor contains an array of parameter descriptors (`ParameterDesc`), which are used to build up the structure and contain descriptive information. Parameter descriptors can be of type simple value (numbers, e.g. float, double and integer) or of type `ParameterTransDetailDesc`. The latter are the nodes in the structure and represent the choices or sequences (determined by the attribute `type`). The attribute `functions` bears descriptors of functions responsible for transformation from high level to low level parameters. The parameter descriptors of type simple value, e.g. `ParameterFloatDesc`, represent leaves within the hierarchical tree structure and contain meta information for the parameters, like range and allowed values.

In contrast to class `ParameterDesc`, the class `Parameter` is not structured and contains only the actual values of high level parameters. Thereby a separation of description (configuration and context) and content (concrete parameters for a segment) is guaranteed. The connection is realized by the unique identifier attribute `id` in both classes, which is used by the software but invisible to the user. Since the view on the high level parameters may change and restructuring become necessary, this separation enables restructuring without affecting parameters already used.

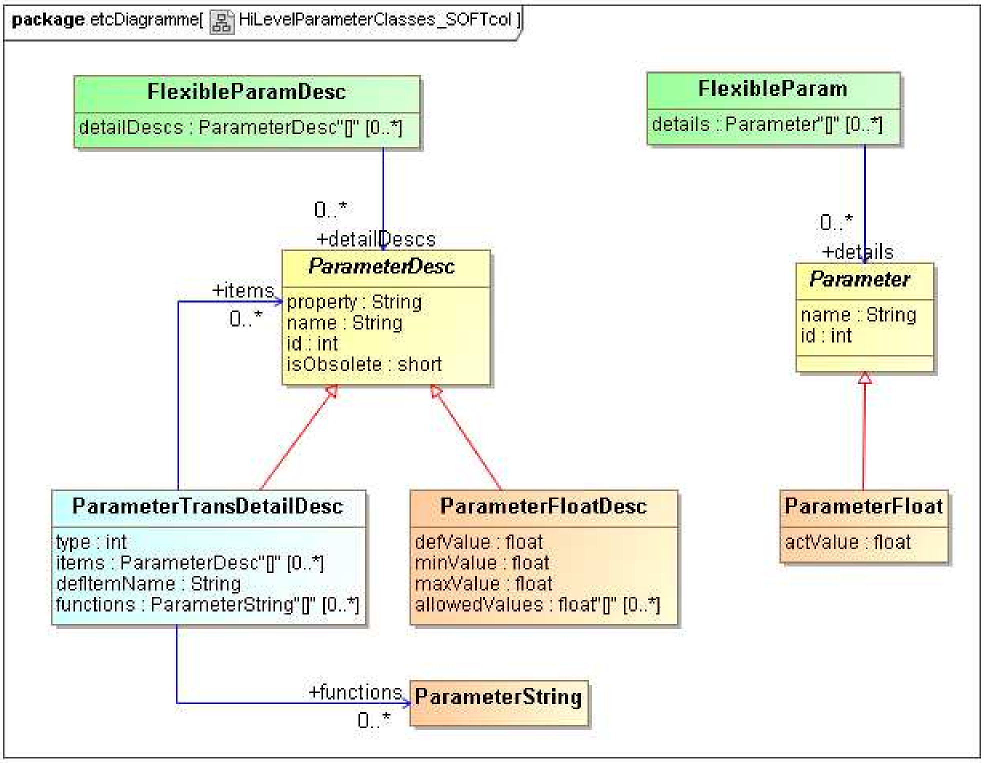

Fig. 2: Structure of the high level parameter descriptors and the corresponding parameters as UML (Unified Modelling Language) diagram

Even though high level parameters can be anything technically or physically meaningful, system components need their complete low level parameter set to operate correctly. This is guaranteed by a user-defined transformation function mapping high level parameters onto low level parameters. Low level parameters not being specified by high level parameters will be set to default values provided by the corresponding pattern. Here, a pattern is a complete set of default low level parameters and has to be defined by the user as a reference. Hence a complete segment is defined by a set of high level parameters and a pattern.

3.1. Transformation

When all high level parameters have been specified, the user can initiate the start of the transformation. In this process, the default low level parameters from the specified pattern will be partially overwritten with the calculated low level parameters from the transformation functions. In order to do so, the transformation function as well as the target path to the corresponding low level parameter must have been specified (in the attribute `functions` of class `ParameterTransDetailDesc`). For every low level parameter target the corresponding transformation function is called and subsequently calculates the corresponding low level parameter from the complete set of high level parameters within a sequence or choice:

$LLP_j = f\,(HLP_0, \ldots, HLP_n)$ [LLP = low level parameter, HLP = high level parameter]

The result of the calculation is assigned to the associated target path and thereby overwrites the default setting from the pattern.

Transformation functions can range from very simple to highly complex, depending on the level of abstraction and interdependencies. For example, a simple one-to-one transformation function would be the direct definition of individual coil currents. A highly complex transformation function would be the calculation of individual coil currents from a given magnetic field strength and rotational transform.

Transformation functions have to be implemented by users, i.e. experimental physicists and engineers, with the assistance of the CoDaC (Control, Data acquisition and Communication) group. Additionally a frame generator is used to interpret the high level parameter descriptor structure and to generate code stubs for the integration of user-defined model functions.

3.2. Editors

There are two specialized editors to create and edit the high level parameters, the group and segment editor [1]. The group editor is an expert tool used to create the structure of high level parameters and to specify the transformation functions as well as the corresponding target paths to low level parameters. The segment editor is a user tool to edit the content of high level parameters and initiate the transformation.

4. Example implementation at WEGA

The small stellarator experiment WEGA is used as a test bed for the control and data acquisition system of W7-X and diagnostics prototypes. Therefore the WEGA is also used to prove the high level parameter concept. This includes testing and using the group and segment editors, which are already available with their basic functionality. Experiences with high level parameters at WEGA should help to improve and extend this functionality.

The segment duration is one example for a high level parameter. At WEGA, segments of specified durations are already in use. One segment can be terminated by different conditions, e.g. by an operator action, after a specified duration or when the next segment gets ready for execution (i.e. waiting for some technical feedback to continue). The latter termination condition can, for example be used for microwave heating systems to wait until the high voltage needed to drive emission is available. In the segment editor the user can select when the segment should end and accordingly edit the values of the high level parameters for the selected choice. In case of a specified duration being chosen, the length of the segment has to be given in seconds. The transformation function used in this case is a one-to-one function.

The operation mode, e.g. stand-by, calibration or data acquisition, is another example for a high level parameter choice. Other examples for WEGA high level parameters that use the one-to-one transformation function are the calibration, spectrometer wavelength, exposure time, gain of phase meter and enabling of data acquisition. User-defined transformation functions are used for high level parameters like magnetic field strength, rotational transform, plasma radius, gas type, heating power, selection of the activated magnetron and others.

With its seven components and currently 2.000 segment parameters, WEGA is complex enough to test the high level parameter concept. Until now the WEGA has been operated without high level parameters and only experts are able to define the complex segment descriptions. Experiences show that the use of high level parameters is essential. Currently it is possible to create and edit high level parameters, and one-to-one transformations have been successfully executed. User-defined transformation functions have already been specified, but still need to be tested.

5. Summary

The preparation and description of long discharges like in W7-X using low level parameters is a very complex process. A reduction of complexity is achieved by using high level parameters, an abstraction layer that shields the user from the technical parameter world and filters it. It allows physicists to focus on physics parameters. In comparison to shot-based and other existing experiments, this is an entirely new, scientifically-oriented approach to support planning of experiments.

The high level parameter concept is designed to be generic and flexible so that even different levels of abstraction are possible. Moreover, the high level parameters cannot only be used to plan experiments, but also to observe discharges and to start later data retrieval. This could be done by displaying high level parameters for observation purposes and by searching for them in the logs. The abstraction layer is essential for continuous operation experiments and can be used for W7-X as well as for laboratory and other experiments.

References

[1] P. Heimann, T. Bluhm, Ch. Hennig, H. Kroiss, G. Kuhner, J. Maier, H. Riemann, M. Zilker, Database structures and interfaces for W7-X, in: Proceedings of the 6th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, April 2008, Fusion Engineering and Design 83 (2–3) (2008) 393–396.

[2] G. Kühner, P. Heimann, S. Heinzel, Ch. Hennig, H. Kühntopf, H. Kroiss, J. Maier, J. Reetz, M. Zilker, Editor for system configuration and experiment program specification, in: Proceedings of the Fourth IAEA Technical Meeting on Control, Data Acquisition and Remote Participation for Fusion Research, San Diego, USA, 2003, Fusion Engineering and Design 71 (2004) 225–230.

[3] P. Heimann, S. Heinzel, Ch. Hennig, H. Kühntopf, H. Kroiss, G. Kühner, J. Maier, J. Reetz, M. Zilker, Status report on the data acquisition system ofWendelstein 7-X, in: Proceedings of the Fourth IAEA Technical Meeting on Control, Data Acquisition and Remote Participation for Fusion Research, San Diego, USA, 2003, Fusion Engineering and Design 71 (2004) 219–224.

[4] H. Kühntopf, H. Kroiss, T. Bluhm, P. Heimann, Ch. Hennig, G. Kühner, J. Maier, M. Zilker, W7-X control group, Specialized editor for processing objects in a database to prepare discharges for WENDELSTEIN 7-X, in: 5th IAEA TM on Control, Data Acquisition, and Remote Participation for Fusion Research – 5th IAEA TM, July 2006. Fusion Engineering and Design 81 (15–17) (2006) 1741–1745.

[5] E.Gamma,R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-oriented Software, AddisonWesley, Reading, MA, 1995.