A W7-X experiment program editor – a usage driven development

Anett Spring[a,*], Marc Lewerentz[a],
Torsten Bluhm[a], Peter Heimann[b], Christine Hennig[a], Georg Kühner[a], Hugo Kroiss[b],
Johannes G. Krom[a], Heike Laqua[a], Josef Maier[b], Heike Riemann[a], Jörg Schacht[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.: +49-3834-88-2757; Fax: +49-3834-88-2509; E-mail address: anett.spring@ipp.mpg.de*

Abstract

The set-up of experiment programs for the complex fusion device Wendelstein 7-X has to define a multitude of parameters which have to obey large number of rules arising from physics and technical constraints. Since this is hard to automate as long as the dependencies are not known sufficiently, the W7-X CoDaC team decided to implement an editor following a constructive approach: Starting from an established experiment program the user is able to modify parts of it – thus complying the usual workflow of experimenters.

Already the very first implementation has been deployed at the W7-X CoDaC prototype, the WEGA stellarator. Driven by agile programming principles the weighting of the requirements has been influenced by the editor usage in the daily experiment routine, thus ensuring client-oriented development steps and short release cycles. At present, a stable program editor implementation with graphical preview, immediate feedback on user actions and instantaneous warnings about incorrect settings is under continuous operation at the CoDaC prototype. It has potential to improve together with growing knowledge about the physical and technical constraints. The experiences gained give certainty that the editor is suitable for future use during the start-up phase and the first years of W7-X operation.

1 The structure of W7-X experiment programs

Wendelstein 7-X is currently built as a highly complex fusion experiment capable of running long-term discharges. Planning a W7-X experiment means to define a multitude of parameters and their intended sequential change, while observing the physical and technical dependencies and constraints.

To assist the user with this complex task, three basic concepts have been introduced by the W7-X CoDaC team: the hierarchical component layout, the time segmentation and the abstraction of high-level parameters [1], [2], [3]. This leads to a structured experiment program representation (Fig 1), reflecting both the arbitrary time slice segmentation and the component hierarchy: each time segment for the W7-X project as a whole consists of a set of component tasks, one for each component (where "components" imply both technical machine components and diagnostics). Consecutive segments may be aggregated to scenarios to structure a program into logical parts and to assist reusing such parts in new programs.

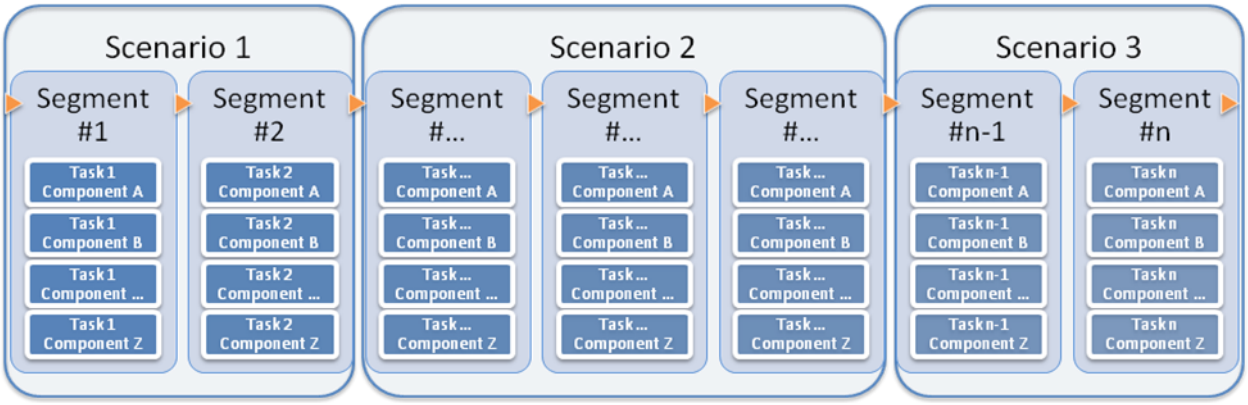

Fig 1 Schematic structure of W7-X experiment programs

The behaviour of each component within its task is defined by a set of high level parameters and their low level counterpart. The high level parameters reflect a more abstract view on the experiment settings. They will be transformed to low level values for the technical parameters, namely the set values to be provided for the control modules of all involved ControlStations as the PC based interfaces to the actors and sensors in the W7-X segment control processes. To give an example: Physicists want to define the magnetic configuration by physics parameters as the value of the plasma radius, the magnetic strength on axis, and the rotational transformation. These values have to be transformed to values for the current for each of the W7-X coils, and – in the end – to the setpoints of the related actuators for the power supplies. See [3] for more details.

Structure and content of a program are persistent in the segment database: the sequence of segments, the aggregation to scenarios, and the high level and the low level parameters of the components' tasks.

2 Editor implementation approach

The very first set-up of the complex program structure is a difficult job and hard to automate as long as the physical and technical dependencies of W7-X are not known sufficiently. On the other hand, the experimenter's usual way is to adjust experiment settings step by step – an approach legitimating the development of a 'constructive' editor: Starting from an established experiment program the user is able to modify it as requested.

From these considerations, two years ago we started to implement a first prototype editor [4] which resolves and displays the elements of a program into a table-like pattern (fig 2 – the middle part) and provides the possibility to edit the values of high level parameters, check simple parameter limits and transform them to low level parameters. Details of the workflow from parameter editing to persistence have been described in [4].

Positive feedback from the prototype users encouraged us to start a detailed requirements analysis for a program editor suitable for the first years of W7-X operation. Three main requests have been determined: (1) the graphical preview of typical physical and/or technical values and their variation in time within an experiment program, (2) the option to make structural changes on a program by adding, removing and re-arranging its elements, and (3) the possibility to define and evaluate more complex constraints regarding component- and time-spanning limits and operation ranges. The latter is to ensure the technical feasibility of the resulting program as far as possible. While (2) is mainly a challenging GUI programming task, for the topics (1) and (3) the segment description concept had to be enhanced to cope

with dependencies between the elements of an experiment program. For this purpose the new *component model* framework [5] has been developed.

Component models are unique for each component and specify dependencies between its parameters. This may include conversion factors, operation limits, restrictions or even the back and forth transformation of different ways to define the same physical values. The aggregation of component models to a component-spanning project model allows defining dependencies between different components. Both component and project models are persistent in form of Java code and configurable by additional XML based descriptions in the database. They will be instantiated at runtime, offering two main missions: (a) Filled with a segment description (including its aggregated components' tasks), the project model instance fully describes the behaviour of the involved components as well as the inner and inter-components' constraints. And (b): While segment descriptions in the database are static and independently from their position within an experiment program, the component model framework allows to arrange model instances in a way to let them know their predecessors and successors. Both leads to a program-spanning experiment model matrix – a counterpart to the table-like program aggregation out of segments and component tasks (Fig 1), but with one main difference: the experiment model additionally describes the relations between all the program parts. See [5] for more details.

This fundamental approach facilitates different *views* on an experiment program, e.g. for computing complex dependencies within a program or computing parameter curves over time. With growing knowledge of the machine's behaviour and its operation ranges, the component models can be enhanced step by step – as soon as the constraints can be described in a way suitable for automatic processing – giving certainty of a future-proof implementation.

3 Usage driven development

Already at the first development stage, the editor has been deployed at the W7-X CoDaC prototype, the WEGA stellarator [6]. Very soon it has been used by the operation team in the daily experiment routine what in turn significantly influenced the weighting of the requirements.

The Java/Swing implementation has proven to be a suitable and flexible framework for a highly responsible user interface. The application development is embedded in the CoDaC-wide quality assurance system [7] including code versioning, code checking, continuous integration tests and release versioning. The provisioning via Java Webstart allows automatic application updates in short release cycles. A ticket system for users and developers helps to track bugs and feature requests.

The project planning based on detailed discussions of the intended workflows (as end-to-end scenarios) with the users, both the WEGA operators and other physicists with fusion experiment experience. The development milestones have been determined and prioritized according to user needs and the estimated development efforts. At monthly iteration meetings the next steps are defined, milestones monitored and experiences of the users discussed. Short daily reporting sessions of the development team assure problems to be identified and solved promptly. New features are presented to the users 'on-the-job', sometimes even as development snapshots, to get immediate feedback. All this ensures agile and client-oriented development steps – resulting in a well-tested editor application with the following features:

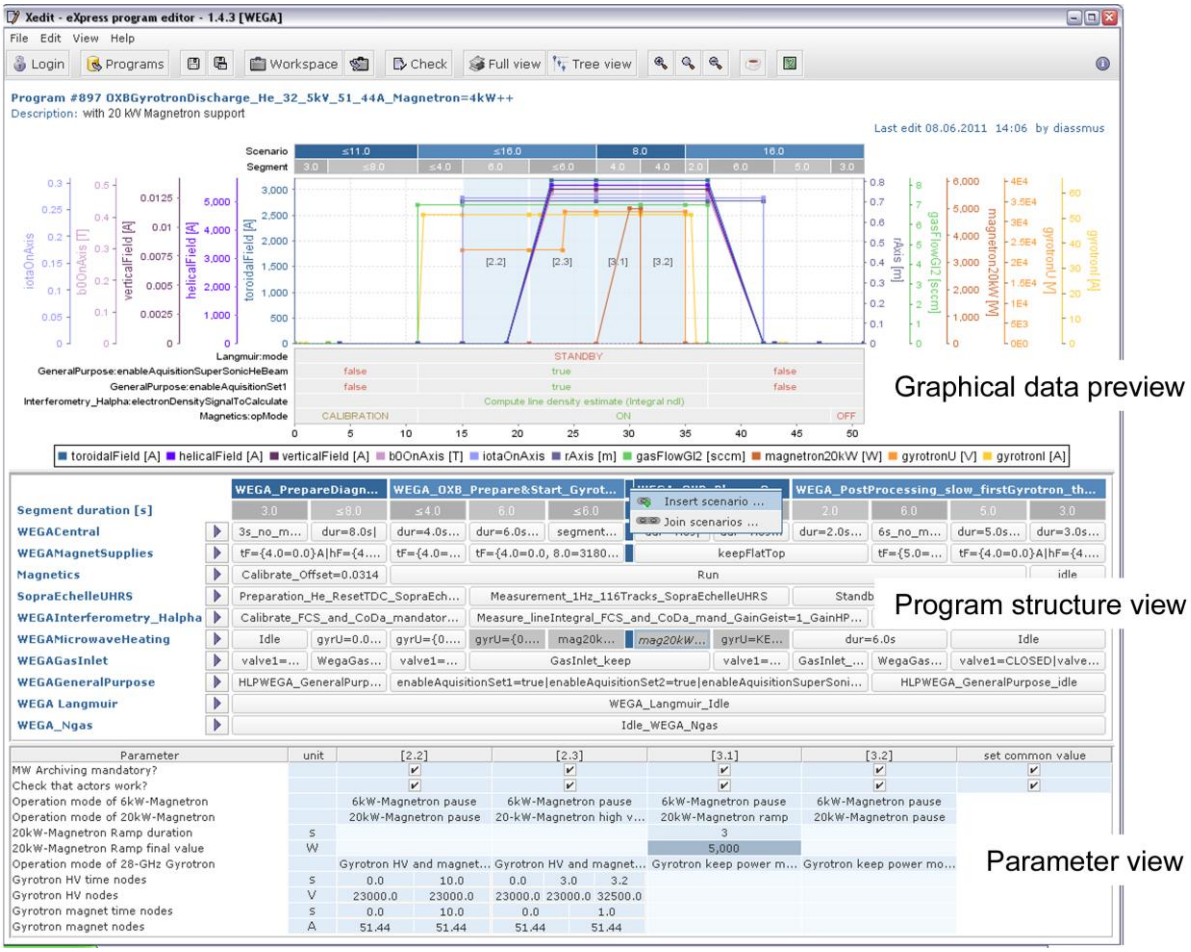

Fig 2 Screenshot while editing a WEGA program: Four tasks of the component 'MicrowaveHeating' have been selected in the program structure view to show their high-level parameters in the parameter view. The time span of the related segments is marked in the graphical data preview (with darker background). One of the parameters has been changed in the parameter view and is highlighted (with darker background).
The user started to insert a scenario via mouse operation in the program structure view.

Program editing assistance

One principle of the editor implementation is to assist the operator during the experiment planning as far as possible and to ensure the creation of programs, which obey all the currently known rules and will be most likely feasible at runtime.

The program structure view in the editor user interface (fig 2 – the middle part) reflects the fact that the behaviour of a specific component may not change on every segment switch; for clarity reasons identical subsequent component tasks are combined to one 'box'. This allows to edit such subsequent tasks in one go.

Clicking a program element opens the parameter view (fig 2 – the lower part), the place to edit values. A clean parameter view will help to concentrate on the main editing tasks, thus parameters not intended to be changed often may be hidden. A full, structured view of all parameters can be shown on demand. While editing, the user will be warned about violations of parameter ranges or missing parameter settings, e.g. after switching from one high level parameter choice to another. If the allowed values for a parameter are restricted to discrete values, they are provided via drop-down boxes. Limited parameter ranges will be shown on tooltips. With the new component model framework all those restrictions are given as a special view of the experiment model. Because they are calculated program-wide, they may

vary with the actually set parameters and will be adapted immediately by announcing changes to the experiment model instance.

It was requested to have an automatic name generation out of the main parameter settings to enhance the program settings overview. This has been implemented as another view of the component model framework to be configured in the database. Besides this the user may name new or changed program elements manually, e.g. to mark standard tasks.

Graphical data preview

Absolutely essential for the operator is a graphical preview of an experiment program, i.e. a presentation of typical physics and/or technical values over time. This data preview (fig 2 – the upper part) complements the schematic program structure view. It has been implemented using the JFreeChart [8] library with the underlying data sets computed as a view of the related experiment model. Generally, the preview may contain both data value curves and status data. In cases where the high-level parameter configuration offers different choices to define technical (low level) parameters from physics (high level) parameters, the component model provides suitable transformations to get the appropriate data preview. The available data curves in the preview chart are configured with the component and project model setup in the database. Out of this fund each user can select the actually displayed curves.

In contrast to the structure view, the data preview displays the relative duration of each segment. The program structure view and the data preview are connected visually: on mouse hovering over program elements the related preview area will be highlighted. According to the principle "What you see is what you get", the data preview is adapted instantaneously while editing parameters. This has been implemented by immediately announcing parameter changes to the experiment model instance (with its underlying project and component model instances). The response time has proven to be adequate for smooth user interaction.

Structural changes on a program

A crucial demand of the editor users is the possibility to change the structure of a program, e.g. to prolong a program or to change the behaviour of one component or diagnostic within a program. In the program structure view the user is able to insert, remove, join, or split scenarios, to insert or remove segments, and to replace tasks within segments. The user guidance has been investigated together with the users. It is entirely mouse assisted: the mouse has to be moved over the location where one of the mentioned actions shall occur; via context menus the available actions at this point are provided. The affected program elements are highlighted to visually assist the user during the editing process.

Even here the data preview will be updated instantaneously by re-computing the experiment model instance.

Good practice

The editor application provides a number of standard user assisting features, such as online user manual, undo/redo operations, user settings restoration on application start and information about available application updates. Summaries of the settings are shown on tooltips on all the program elements. Time consuming processes inform the user about the progress and provide an option to cancel where reasonable. When closing the editor after a program has been changed but not yet saved, the user will be warned.

Additionally, a program explorer has been implemented. It allows listing all available programs, sorting and filtering by name and description, marking favourites, and locking to avoid unwanted editing of standard programs.

4 Status and outlook

Besides the WEGA operators, the program editor is routinely used by physicists in laboratories or at in-house test facilities, amongst them the W7-X ECRH team during the gyrotron tests and commissioning. The experiences gained from the close co-operation between developers and users give certainty that the editor implementation is suitable and upgradeable for the future use during the start-up phase and the first years of W7-X operation. It is applicable both for project (W7-X) programs as well as for component programs for autonomous operation during commissioning or in laboratories.

For the future there are still several features and enhancements that have been deferred for now. This concerns the handling of longer programs (up to 30 min at W7-X) with a growing number of components, which demands to rework the program display, e.g. implement options to group components, hide and zoom program parts. Another point is the enhancement of the program explorer to manage the increasing number of W7-X experiment programs.

Acknowledgement

The developers would like to thank the WEGA team for the close co-operation, intense testing and many fruitful discussions.

References

[1] Jörg Schacht, Heike Laqua, Marc Lewerentz, Ina Müller, Steffen Pingel, Anett Spring, Andreas Wölk, Overview and status of the control system of WENDELSTEIN 7-X, Fusion Engineering and Design, Volume 82, Issues 5-14, Proceedings of the 24th Symposium on Fusion Technology - SOFT-24, October 2007, Pages 988-994, ISSN 0920-3796, DOI: 10.1016/j.fusengdes.2007.02.009.

[2] Heike Laqua, Helmut Niedermeyer, Jörg Schacht, Anett Spring, Real-time software for the fusion experiment WENDELSTEIN 7-X, Fusion Engineering and Design, Volume 81, Issues 15-17, 5th IAEA TM on Control, Data Acquisition, and Remote Participation for Fusion Research - 5th IAEA TM, July 2006, Pages 1807-1811, ISSN 0920-3796, DOI: 10.1016/j.fusengdes.2006.04.005.

[3] Heike Riemann, Torsten Bluhm, Peter Heimann, Christine Hennig, Georg Kühner, Hugo Kroiss, Heike Laqua, Marc Lewerentz, Josef Maier, Jörg Schacht, Anett Spring, Andreas Werner, Manfred Zilker, Experiment planning using the high level parameter concept, Fusion Engineering and Design, Volume 85, Issues 3-4, Proceedings of the 7th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, July 2010, Pages 478-481, ISSN 0920-3796, DOI: 10.1016/j.fusengdes.2009.10.009.

[4] Anett Spring, Marc Lewerentz, Torsten Bluhm, Peter Heimann, Christine Hennig, Georg Kühner, Hugo Kroiss, Heike Laqua, Josef Maier, Heike Riemann, Jörg Schacht, Andreas Werner, Manfred Zilker, A first W7-X experiment program editor, Fusion Engineering and Design, Volume 85, Issues 3-4, Proceedings of the 7th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, July 2010, Pages 525-528, ISSN 0920-3796, DOI: 10.1016/j.fusengdes.2009.10.002.

[5] Marc Lewerentz, Anett Spring, Torsten Bluhm, Peter Heimann, Christine Hennig, Georg Kühner, Hugo Kroiss, Johannes G. Krom, Heike Laqua, Josef Maier, Heike Riemann, Jörg Schacht, Andreas Werner, Manfred Zilker, Experiment planning using high-level component models at W7-X, this issue

[6] Heike Laqua, Dieter Aßmus, Torsten Bluhm, Peter Heimann, Stefan Heinrich, Christine Hennig, Uwe Herbst, Eric Köster, Hugo Kroiss, Georg Kühner, Josef Maier, Ina Müller, Marc Lewerentz, Matthias Otte, Steffen Pingel, Heike Riemann, Jurgen Sachtleben, Jörg Schacht, Anett Spring, Andreas Wölk, Andreas Werner, Manfred Zilker, Test of the steady state W7-X control and data acquisition system at the WEGA stellarator, Fusion Engineering and Design, Volume 85, Issues 3-4, Proceedings of the 7th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, July 2010, Pages 520-524, ISSN 0920-3796, DOI: 10.1016/j.fusengdes.2009.10.005.

[7] Georg Kühner, Torsten Bluhm, Peter Heimann, Christine Hennig, Hugo Kroiss, Alexander Kruger, Heike Laqua, Marc Lewerentz, Josef Maier, Heike Riemann, Jörg Schacht, Anett Spring, Andreas Werner, Manfred Zilker, Employing industrial standards in software engineering for W7X, Fusion Engineering and Design, Volume 84, Issues 7-11, Proceeding of the 25th Symposium on Fusion Technology - (SOFT-25), June 2009, Pages 1130-1135, ISSN 0920-3796, DOI: 10.1016/j.fusengdes.2008.12.019.

[8] JFreeChart, free (LGPL) chart library for the Java(tm) platform, http://www.jfree.org/jfreechart/, http://sourceforge.net/projects/jfreechart/