Fusion Engineering and Design 214 (2025) 114880

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Distributed editing of experiment programs at Wendelstein 7-X

Anett Spring *, Heike Riemann, Marc Lewerentz, W7-X Team [1]

Max Planck Institute for Plasma Physics Greifswald, Germany

ARTICLE INFO

Keywords:
Wendelstein 7-X
Segment Control
Experiment program editor
Distributed editing
Xedit
TaskLinks

ABSTRACT

Configuring experiment programs at complex fusion experimental facilities such as Wendelstein 7-X is a challenge: the amount of parameters, the growing number of components, the required specialized knowledge of technical systems and diagnostics, the complex timing – all this demands an integrated solution for experiment leaders, engineers, and physicists.With the growing number of integrated components, the W7-X experiment program editor Xedit was extended to cope with *TaskLinks*: allowing externally configured program parts of individual components to be linked into the planned sequence of the central experiment program. The preparation of the specific *Tasks* is done by the component owners using a local Xedit instance – in the same way, as the users are already familiar with from the creation of local programs for commissioning or calibration runs. Centrally, as with all integrated components, the linked components' *Tasks* are then visualized and all parameters are checked for limit violations or other pre-defined rules before saving the complete planned program ready for execution.

1. Motivation

Wendelstein 7-X – as the world's largest stellarator-type fusion device – is equipped with an outstanding number of technical systems and diagnostic facilities compared to other fusion experiments and offers enormous flexibility: the coil system allows different magnetic field configurations, there are three heating systems, a pellet injector and various invasive diagnostic systems. Today, W7-X comprises more than 40 systems integrated into the Segment Control framework with far more than 200 ControlStations.

Configuring experiment programs at such complex fusion experimental facilities is a challenge: the amount of parameters for the growing number of involved components, the required specialized knowledge of technical systems and diagnostics, the complex timing that has to be aligned between dependent components – all this demands an integrated solution for experiment leaders, engineers, and physicists.

For the scientific evaluation of plasma experiments with different physics objectives, an individual configuration is required for a number of diagnostic systems, which in some cases has to be found during the experiment. The first experimental campaigns showed that the experts needed to make diagnostic-specific adjustments to the planned experiment program setup in a timely manner. These were to be communicated to the experiment leaders. Due to the tight session schedule at W7-X, this is a critical point for effective experiment planning and execution. Suitable software support is therefore urgently required, preferably in the familiar and commonly used graphical user interface.

2. W7-X segment control framework

2.1. Scope and benefits

W7-X has been going successfully through its first operation phases showing reliable operation assisted by the W7-X Segment Control and experiment-planning framework [1,2]. Both the central experiment operation and the stand-alone operation of integrated components and diagnostics benefit from the compound framework with

- complete program description of all component parameters in a "planned program" [3],
- pre-checking for reasonability of experiment program parameter settings [4],
- automated mapping of high-level physics parameters onto the technical machine parameters [5],
- standardized data exchange throughout the component hierarchy, enabling central event-based segment switching, straightforward

* Correspondence author.
E-mail address: anett.spring@ipp.mpg.de (A. Spring).

[1] See the authors list in [1].

https://doi.org/10.1016/j.fusengdes.2025.114880
Received 30 September 2024; Received in revised form 23 December 2024; Accepted 16 February 2025
Available online 27 February 2025

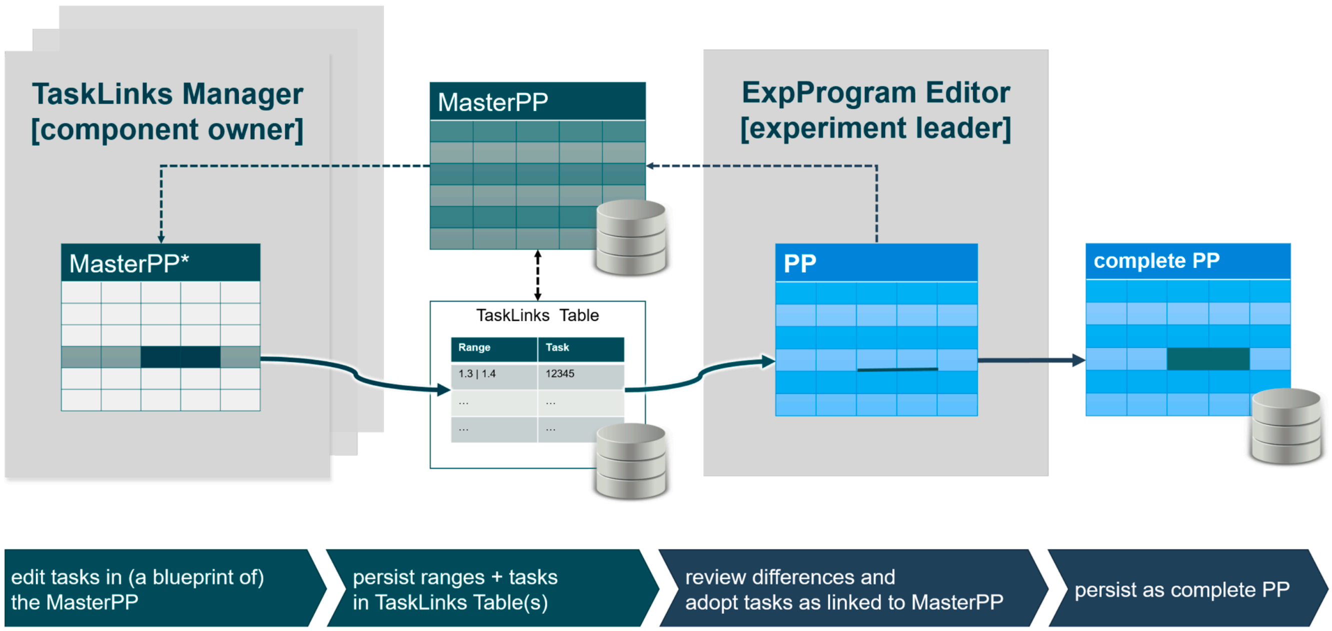

Fig. 1. Schematic overview of the TaskLinks framework design.

event handling and a standardized setup of complex feedback control [6],

- continuous data streaming and standardized parameter logging [7, 8],
- online monitoring of experiment data [9],
- automated meta data enrichment along the workflow: from session planning to experiment setup, runtime information and experiment evaluation [10,11]
- … and the use of common CoDaC[2] applications for experiment preparation, execution, monitoring, and data access [12]

2.2. W7-X segment control key characteristics

The basis of W7-X Segment Control is an arbitrary segmentation of experiment programs and the timing by centrally announced segment switches – triggered by events or e.g. (in the simplest case) after a fixed duration. *Segments* contain the complete parametrization of the planned behavior for this time interval in form of a *Task* for each of the integrated components (as technical systems or diagnostics). These *Tasks* represent a specific parameter set and can be re-used – both in other experiment programs and for stand-alone test operation of a component. To provide an overview, *Segments* are usually grouped in several *Scenarios* by content-related aspects such as preparation, plasma discharge, and post-processing phase. The resulting *Planned Program* (PP) fully describes the intended synchronized behavior of all involved components and is stored in a programs database.

Another important point and relevant difference to other so-called segment control systems [13,14] is that successive *Segments* can contain the identical *Task* for a component. In this way, the behavior of a component can be defined for a longer segment-spanning interval and only relevant *Segment* switches are associated with actual parameter changes by a component. Use cases are initialization phases of individual length per component, or e.g. uninterrupted modulation or feedback control processes.

The W7-X Segment Control Framework is basically designed to be universal, so that the tools and processes for editing, visualizing and constraints checking as well as the transformation and persistence of the technical parameters can be used both for the parameterization of (central) plasma discharge experiments and for (stand-alone) component operation during commissioning or calibration. The W7-X experiment program editor Xedit [3], as the currently only visual experiment editor for fusion devices, supports the special view on the structuring of *Scenarios, Segments* and *Tasks.* Xedit provides a complete overview of the (high-level) parameterization of the integrated components and persists the resulting technical parameters in a central database, which serves as the parameterization source for the control and data acquisition systems. It was implemented with regard to the W7-X Segment Control workflow. However, it has proven to be sufficiently flexible to also support the configuration of experiments at other fusion facilities such as WEST [15] or MPEX [16].

2.3. The challenge of complexity

While at other fusion experiments typically only the core feedback processes are set up centrally and diagnostics are only integrated via central triggers, W7-X takes a different approach: here all the component settings are configured within a central experiment program. This has the major advantage that the metadata of all involved components is recorded and processed together in order to store it in a central location - the W7-X Logbook [10]. While the W7-X Segment Control framework itself was implemented flexibly enough to cope with the growing number of integrated components to be controlled, the experiment program editor Xedit had to be adapted for efficient setting of the many experiment parameters.

The core requirement remains that the experiment planner always has the possibility to get a complete overview of the involved components and to intervene if necessary. However, with an increasing number of systems, he is neither able to enter every parameter value nor does he know about the reasonableness of internal parameters in detail. Experiment planning must become a joint but coordinated task of the involved technicians and physicists.

The challenge is to enable and support the joint creation of experiment programs, i.e. the distributed editing by experts. This requires (a) a common reference to the central program structure and (b) the implementation of an aggregation process to merge the component parameters into the central *Planned Program* (PP). The solution has to assist and force teamwork while retaining the visual overview. It has to focus on

[2] *The W7-X Control Data Acquisition and Communication group.*

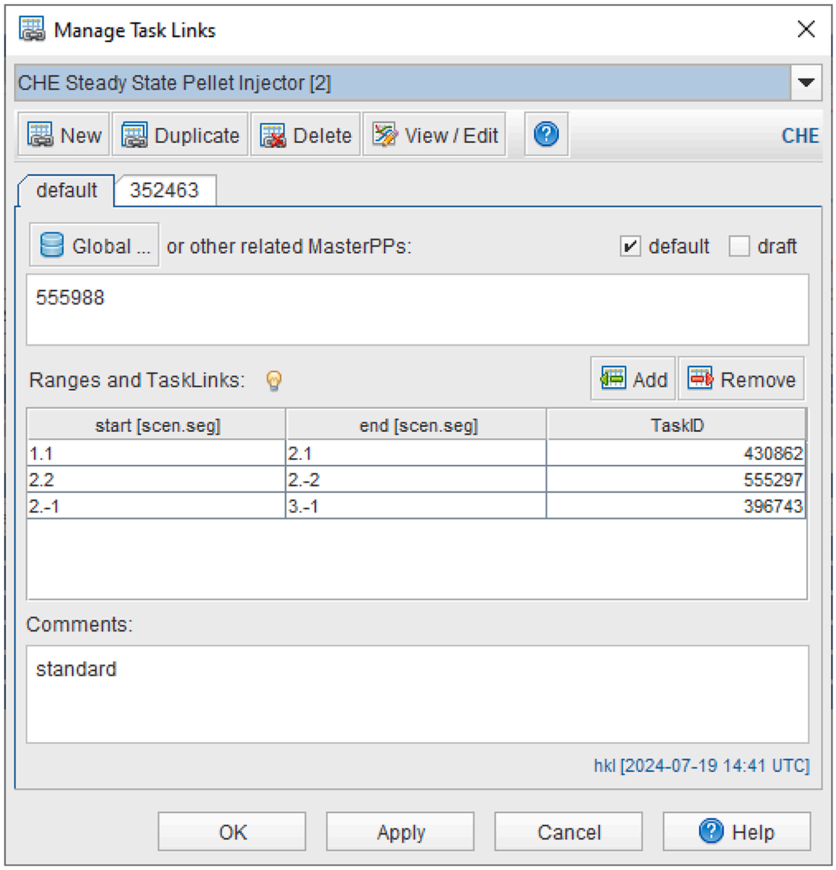

Fig. 2. *TaskLinks* Manager.

the users' requirements – both the experiment leaders and component owners.

3. Solution: Linking task parameters

The proposed solution to this co-editing challenge is to implement a linking mechanism between the central experiment program and *Tasks* to be prepared by the component owners for specific *Segment* ranges in the central PP.

3.1. Concept and requirements

The basic design idea of such *TaskLinks* is sketched in Fig 1:

(1) A common *MasterPP* serves as structural reference – both for the *Task* parameters to be provided by the component owner and the central PP to be setup by the experiment leader
(2) Component *Task* parameters are specified for freely selectable *Segments* in the *MasterPP*. The links to the resulting *Tasks* are persisted in a *TaskLinks* table with reference to the *MasterPP* and its structure.
(3) The *TaskLinks* provided per component are used to merge component *Tasks* into the planned sequence of the central experiment program at the referenced positions.
(4) The resulting central PP contains the complete set of *Task* parameters for all components and segment intervals. The PP can be visualized, checked for constraints and stored.

For practicability and general user acceptance, the implementation must fulfil additional requirements:

- *TaskLinks* may cover all or only a subset of *Segments*.
- The implementation must be sufficiently flexible to handle different *Segment* structures in central PPs.
- The component owner should have the option to define one default set of *TaskLinks* that can cover any other *MasterPP* as long as the structure fits.
- The component owner should have the option to prepare *TaskLinks* in advance (as draft), i.e. without making them effective.
- Misconfigured *TaskLinks* must be handled gracefully. They must not impair the functioning of other components *TaskLinks* or the configuration of the central program.
- The experiment leader must be informed about existing component *TaskLinks* to be applied onto the central PP.

In the workflow, it must remain the responsibility of the experiment leader to decide whether or not to accept the linked component *Tasks*, or even to override *Task* parameters.

3.2. TaskLinks implementation

TaskLinks carry information of (1) the component *Task* by its unique ID number, (2) the intended validity range in relation to the *Segment* structure of (3) the associated *MasterPP*.

Several *TaskLinks* entries are held in a *TaskLinks* table (with unique, not overlapping ranges). Several *TaskLinks* tables can be defined related to different *MasterPPs*. One of the table can be marked as default to be valid for any other (not explicitly referred *MasterPP*). To be accessible for both component owners and experiment leaders, these *TaskLinks* tables persist in the programs database.

The structuring in *Scenarios* and *Segments* is used for addressing positions and ranges within PPs. Starting with "1", counted from the

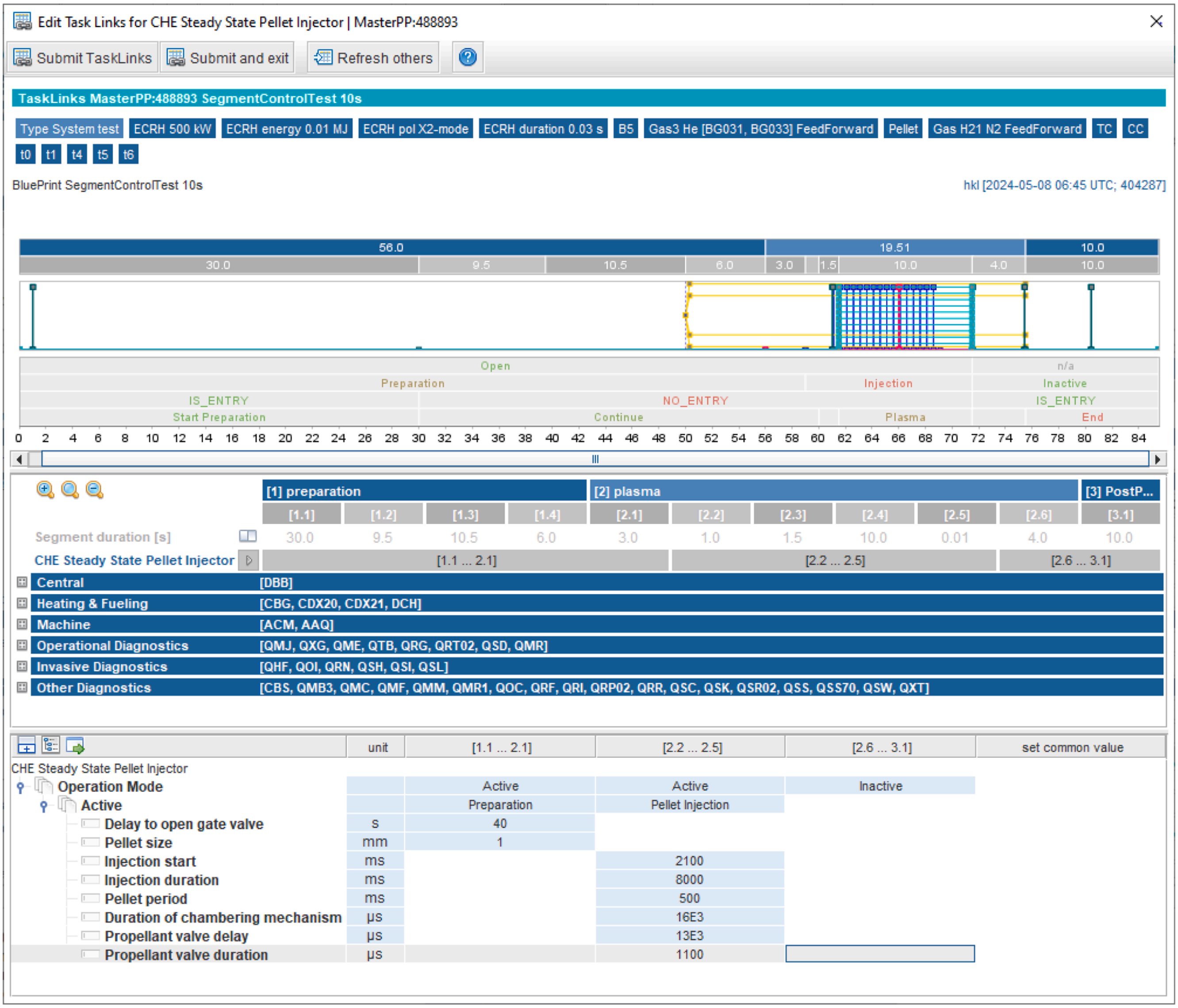

Fig. 3. Visual *TaskLinks* Editor.

beginning or from the end (with a negative sign, e.g. "-1″), positions are given by the combination of *Scenario* number and *Segment* number within a *Scenario*. With this, a range definition can even cover slightly changed timing in *MasterPPs*, e.g. additional *Segments* in a *Scenario*. Examples:

- [1.2] refers to the *Task* position in the second *Segment* within the first *Scenario*.
- [2.-1] refers to the *Task* position in the last *Segment* within the second *Scenario*.
- [-1.-1] refers to the very last *Segment* of a PP.

Ranges are given as start and end position (including):

- [2.3] to [2.-1] refers to the range from the third to the last *Segment* in the second *Scenario*.

In principle, every experiment program / PP can serve as a *MasterPP*. Depending on the physics' goal or certain use cases (as test, reference, standard programs, cleaning discharges or others), it makes sense to use different *MasterPPs*. So one or more PPs are predefined as global *MasterPPs* by the experiment leader.

3.3. User interface

3.3.1. Authorization

Xedit's existing role concept also applies to the creation and modification of the *TaskLinks*: Only the component owners (and the experiment leaders in a special role) are allowed to create, modify, or delete *TaskLinks*.

3.3.2. TaskLinks manager

The *TaskLinks* Manager (Fig 2) has been implemented to manage sets of *TaskLinks* tables by an authorized component owner for their systems or diagnostics. Here, *TaskLinks* tables can be filled directly by specifying (1) the related *MasterPP* (or choosing one of the global ones), (2) the start/end positions, and (3) the IDs of already prepared *Tasks* (manually or even by selecting from the programs database).

TaskLinks tables can be newly created, copied, deleted, set as default (as fallback for other *MasterPPs*) or marked as draft (to set them inactive).

The *TaskLinks* Manager also ensures compliance with the rules:

- not-overlapping range settings in one *TaskLinks* table,
- the existence of the chosen *MasterPP* and *Tasks*,

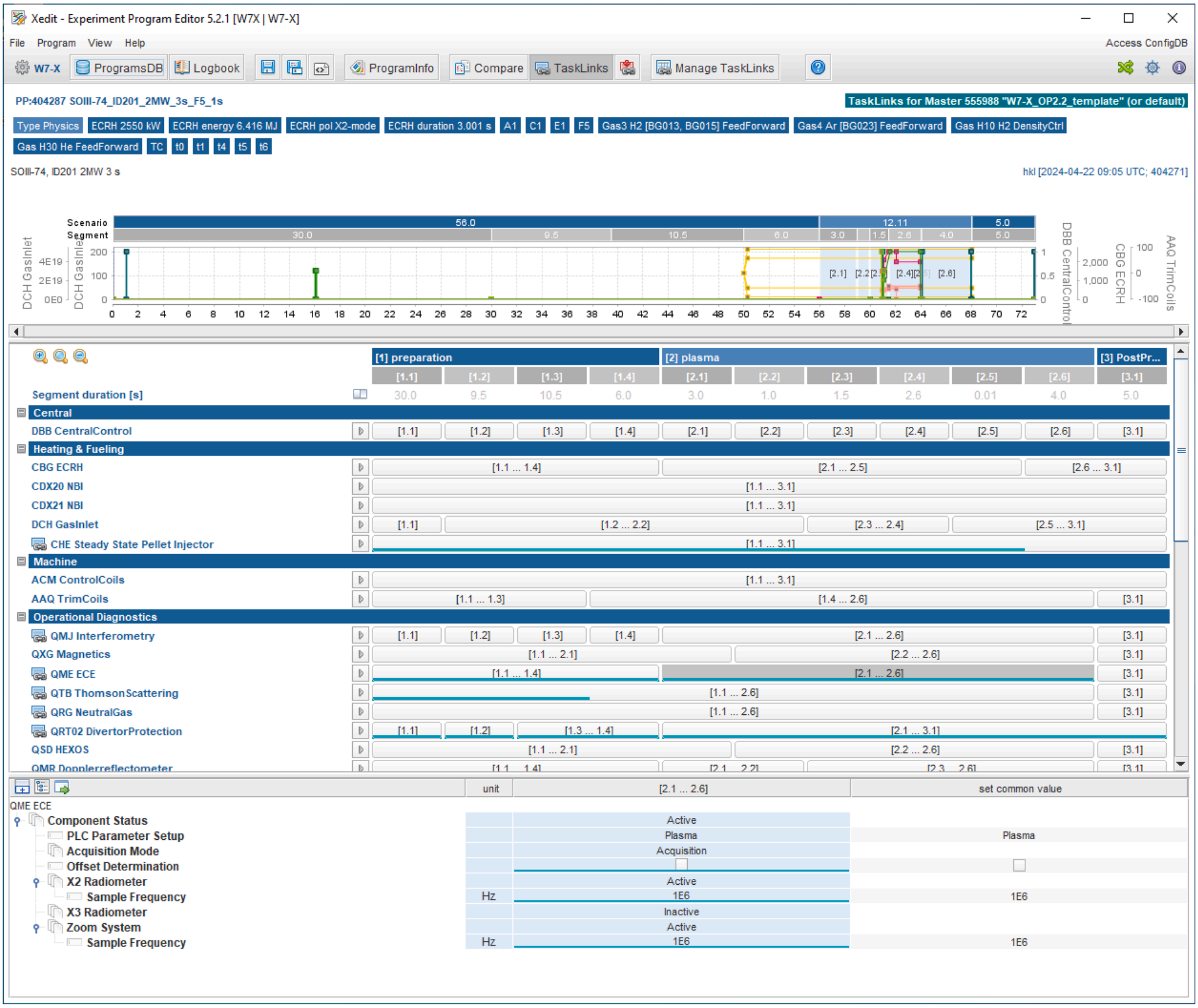

Fig. 4. Adoption of *TaskLinks* in the W7-X experiment program.

- only one *TaskLinks* table for the same *MasterPP*,
- only one *TaskLinks* table marked as default.

TaskLinks tables marked as "draft" are excluded from the checks.

3.3.3. Visual TaskLinks editor

The more convenient option to define *TaskLinks* including their intended ranges is to use the visual editor (Fig 3). It is accessible by the "View/edit" button and provides Xedit's familiar editing possibilities as used when editing a central or stand-alone experiment program. In relation to the *MasterPP* as underlying "blueprint", the user chooses the ranges of their *Tasks* and edits the parameters. At the same time, the user sees not only the program structure but also the current *TaskLinks'* timing and parameter settings of all other components.

With submitting the resulting timing and settings to the *TaskLinks* Manager, two steps are executed: (1) persisting the sets of parameters as component *Tasks* according to the W7-X Segment Control rules, and (2) filling the *TaskLinks* table with the (otherwise manually set) values for the ranges and *Task* IDs.

3.3.4. TaskLinks in central program editor

Xedit was enhanced to (1) continuously poll the ProgramDatabase for changes in the *TaskLinks* tables, (2) show differences to defined *TaskLinks* and (3) merge the *Tasks* as defined in the *TaskLinks* tables into the central experiment program. For this, a special "*TaskLinks* mode" was implemented.

Fig 4 shows Xedit in activated *TaskLinks* mode: After assigning a *MasterPP* to a PP, this mode marks all component *Tasks* with differences to the defined *TaskLinks* for this *MasterPP* (or alternatively from the default table) at this position in the PP. The experiment leader can adopt all *TaskLinks* of one or all components. Alternatively, he can also accept only certain *Tasks*. The linked *Tasks* replace the original *Tasks* in the program. The resulting PP is stored with the adopted *Tasks* parameters at the given ranges.

3.3.5. TaskLinks at program execution time

The actual adoption of the *TaskLinks* in the PP has to be apparent both for the experiment leaders and for the component owners at execution time. For this, the W7-X experiment control user interface Xcontrol [17] has been slightly enhanced.

Xcontrol is the tool to load a program into the execution environment and monitor its progress. In contrast to the general check against predefined rules in Xedit, Xcontrol reflects the actual segment feasibility resulting from the continuous status feedback of all components before and during program execution. As the previously linked *Tasks* are a regular part of the resulting *PP*, there is no difference in the feasibility check.

The implemented enhancement means that the persistent *TaskLinks* tables are now compared with the PP when it is loaded. Any differences are announced and the experiment leader can decide whether to ignore them (e.g. if a program is to run once more exactly as before) or to check and edit the PP accordingly. In addition, the experiment monitor has been extended to color-code differences between the *Tasks* and the related *TaskLinks* tables at component level in the same way as Xedit.

3.4. TaskLinks and component configuration changes

Experimental devices are subject to constant extension and modification. At W7-X, the adaption of existing programs to changes in the control and data acquisition system is addressed with a highly automated DevOps process [18]. As this process covers all program parts, it also includes the *Tasks* that are linked in the *TaskLinks* tables. This ensures that *TaskLinks* remain valid even after component changes and do not have to be adjusted separately.

3.5. Development process

Since the *TaskLinks* feature enhances the already mature experiment program editor, we could rely on the existing development and test environment. Unit tests, continuous integration, fast delivery to users by a centralized automated deployment are standard practice. This supports high quality for any feature enhancements. As a Java implementation with a shared MongoDB [19] instance for data persistence, it runs platform-independently in the users' favorite environment. As a strictly agile project, it has evolved in close collaboration with the experimenters throughout its lifetime - from requirements and user interaction analysis to intensive live testing in several operational campaigns at W7-X and WEST. Close monitoring of the users' work plus a service ticket system ensure fast and direct feedback.

An ever-present challenge is to overcome performance issues with the application's response to user interactions, especially when – with the increasing number of components – an increasing number of constraint rules and data visualizations need to be evaluated in real time. This requires an optimization effort that is not to be underestimated.

4. User experience and workflow

The *TaskLinks* framework has been deployed successfully at W7-X in the recent operating phase and was well received by the users. The acceptance was supported by the fact that the users work in their familiar environment for parameter editing, in the same way as in stand-alone operation, and even have the settings of other components as a reference.

Already in preparation of the experiment campaign, the experiment leaders communicate the global *MasterPPs* with their intended use cases. The component owners prepare their *TaskLinks* tables in advance, but also change them when necessary to react on certain conditions of their systems or diagnostics. With this, a joint program editing workflow has been implemented which is flexible enough to cope with different experiment requirements.

The framework utilizes distributed knowledge of the specific components – while the experiment leaders are still responsible for the overall-system behavior by deciding whether to adopt *TaskLinks*. It does not replace communication between technicians, physicists and experiment leaders but assists and accelerates it and significantly reduces errors during program preparation.

Even if the distributed editing is not a pressing problem for smaller experiments like WEST or MPEX, it can be helpful in automatically reusing or replacing of the behavior of certain components in experiment programs.

CRediT authorship contribution statement

Anett Spring: Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Conceptualization. **Heike Riemann:** Writing – review & editing, Software, Conceptualization. **Marc Lewerentz:** Software, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Anett Spring reports financial support was provided by Euratom Research and Training Programme. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The W7-X CoDaC group likes to thank all the users for thorough and patient testing of the newly introduced *TaskLinks* feature during the last campaign. Their comments and feedback always contribute significantly to improvements - as it is the case for all our frameworks, applications and interfaces.

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Data availability

Data will be made available on request.

References

[1] O. Grulke, et al., Overview of the first Wendelstein 7-X long pulse campaign with fully water-cooled plasma facing components, Nuclear. Fusion 64 (2024) 112002, https://doi.org/10.1088/1741-4326/ad2f4d.
[2] M. Endler, et al., Wendelstein 7-X on the path to long-pulse high-performance operation, Fus. Eng. Design 167 (2021) 112381, https://doi.org/10.1016/j.fusengdes.2021.112381.
[3] A. Spring, et al., A W7-X experiment program editor—A usage driven development, Fus. Eng. Design 87 (Issue 12) (2012) 1954–1957, https://doi.org/10.1016/j.fusengdes.2012.05.007.
[4] M. Lewerentz, et al., Experiment planning using high-level component models at W7-X, Fus. Eng. Design 87 (Issue 12) (2012) 1949–1953, https://doi.org/10.1016/j.fusengdes.2012.09.002.
[5] H. Riemann, et al., Experiment planning using the high level parameter concept, Fus. Eng. Design 85 (Issues 3–4) (2010) 478–481, https://doi.org/10.1016/j.fusengdes.2009.10.009.
[6] H. Laqua, et al., Resource checking and event handling within the W7-X segment control framework, Fus. Eng. Design 87 (Issue 12) (2012) 1958–1960, https://doi.org/10.1016/j.fusengdes.2012.04.023.
[7] T. Bluhm, et al., Wendelstein 7-X's CoDaStation: A modular application for scientific data acquisition, Fus. Eng. Design 89 (Issue 5) (2014) 658–662, https://doi.org/10.1016/j.fusengdes.2013.12.039.
[8] C. Hennig, et al., ArchiveDB—Scientific and technical data archive for Wendelstein 7-X, Fus. Eng. Design 112 (2016) 984–990, https://doi.org/10.1016/j.fusengdes.2016.05.026.
[9] S. Dumke, et al., Next generation web based live data monitoring for W7-X, Fus. Eng. Design 129 (2018) 16–23, https://doi.org/10.1016/j.fusengdes.2018.02.022.
[10] M. Grahl, et al., W7-X logbook REST API for processing experimental metadata and data enrichment at the Wendelstein 7-X stellarator, Fus. Eng. Design 160 (2020) 111819, https://doi.org/10.1016/j.fusengdes.2020.111819.
[11] A. Spring et al, Metadata framework for assisting experimental planning and Evaluation at Wendelstein 7-X IEEE Transactions on Plasma Science 48, no. 6 (2020) 1409–1414, doi:10.1109/TPS.2019.2958034.
[12] A. Spring, et al., Establishing the Wendelstein 7-X steady state plasma control and data acquisition system during the first operation phase, Fus. Eng. Design 123 (2017) 579–583, https://doi.org/10.1016/j.fusengdes.2017.03.127.

[13] W. Treutterer, et al., ASDEX Upgrade Discharge Control System—A real-time plasma control framework, Fus. Eng. Design 89 (Issue 3) (2014) 146–154, https://doi.org/10.1016/j.fusengdes.2014.01.001.
[14] R. Nouailletas, et al., The WEST plasma control system: integration, commissioning and operation on first experimental campaigns, Fus. Eng. Design 146 (2019) 999–1002, https://doi.org/10.1016/j.fusengdes.2019.01.139. Part A.
[15] P. Moreau, et al., Measurements and controls implementation for WEST, Fus. Eng. Design 123 (2017) 1029–1032, https://doi.org/10.1016/j.fusengdes.2017.01.046.
[16] J. Rapp, et al., The Material plasma Exposure eXperiment: mission and conceptual design, Fus. Eng. Design 156 (2020) 111586, https://doi.org/10.1016/j.fusengdes.2020.111586.
[17] A. Spring, et al., User control interface for W7-X plasma operation, Fus. Eng. Design 82 (Issues 5–14) (2007) 1002–1007, https://doi.org/10.1016/j.fusengdes.2007.05.052.
[18] M. Lewerentz, et al., Implementing DevOps practices at the control and data acquisition system of an experimental fusion device, Fus. Eng. Design 46 Part A (2019) 40–45, https://doi.org/10.1016/j.fusengdes.2018.11.022.
[19] MongoDB Community Edition, https://www.mongodb.com/, 2024 (accessed 1 Sep 2024).