The SPIDER Pulse Plant Configuration Environment

Nuno Cruz[a,b], Cesare Taliercio[a], Adriano Luchetta[a], Gabriele Manduchi[a], Andrea Rigoni[a]

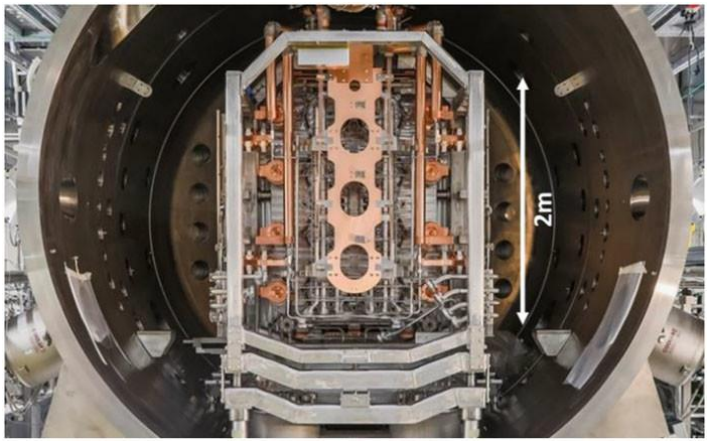
[a] Consorzio RFX, Corso Stati Uniti, 4, 35127 Padova Italy

[b] Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, Universidade de Lisboa, 1049-001, Lisboa, Portugal

IEEE-NPSS 23rd Real Time Conference
1st to 5th of August 2022

The SPIDER Experiment

- The Source for the Production of Ions of Deuterium Extracted from a Radio frequency plasma (**SPIDER**) experiment is a prototype devoted to the heating and diagnostic neutral beam studies in operation at the ITER Neutral Beam Test Facility (NBTF) at Consorzio RFX, Padova.
- SPIDER is the full-size ITER ion source prototype and the largest negative ion source in operation in the world.
- In view of ITER heating requirements to realize plasma burning conditions and instabilities control, SPIDER aims at
  - achieving long-time operation (3600 s) with beam energy up to 100 keV,
  - high extracted current density (above 355 $A.m^{-2}$ for H^- and above 285 $A.m^{-2}$ for D^-) at maximum beam source pressure of 0.3 Pa.
  - maximum deviation from uniformity must be kept under 10%.[1][2]



The inside of the SPIDER ion source.

NBTF High Voltage Hall with
- the insulating transformer (left)
- and the high voltage deck (right)

[1] V. Toigo *et al* 2019 *Nucl. Fusion* **59** 086058
[2] V. Toigo *et al* 2021 Fus. Eng. & Design **168** 112622

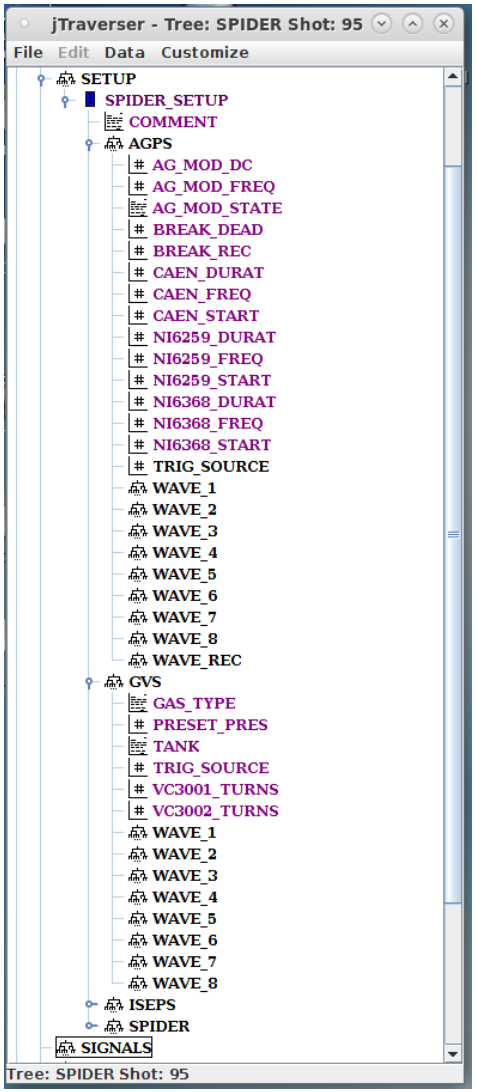
SPIDER Pulse Preparation – Past Situation

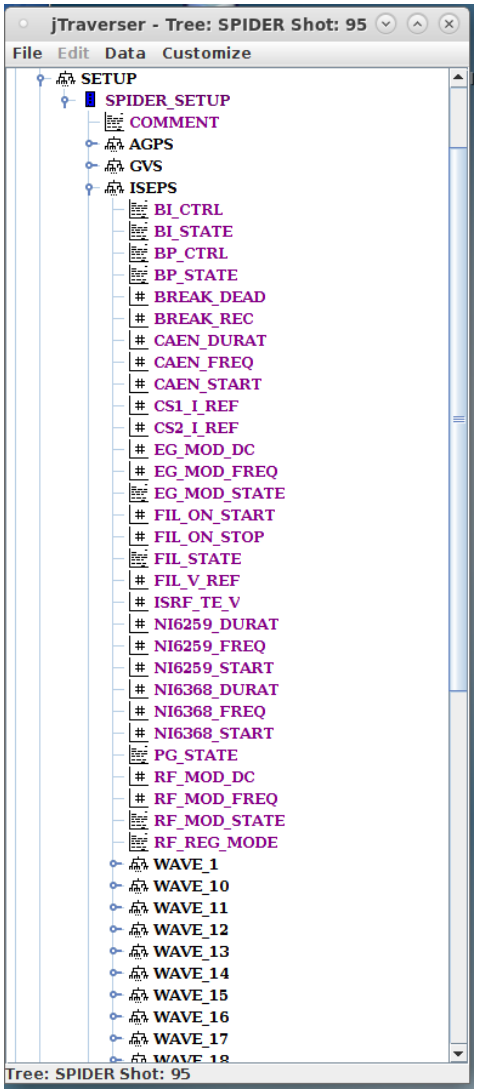
- SPIDER parameters were configured using an MDSplus Human Machine Interface (HMI) using specific pulse number to temporarily store the setup configuration;
- Using a command line the Session Leader (SL – Scientific Coordinator) could load previous setup from an executed shot or reference shot;
- SL could check all the configuration using MDSplus jScope, with all the waveforms;
- An external tool could create the long pulse setup in IDL and store it in a reference shot;

However

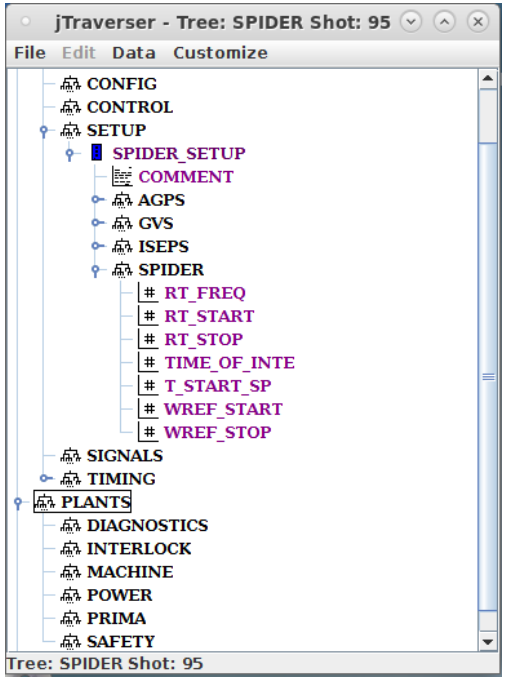
- Responsible Technician (RT – Engineer in Charge) had limited verification/comparison tools (jScope);
- No digital tool for configuration approval and communication process;
- Approval process had to be made on a signed paper;
- Only limited consistency check of parameters is made when loading config;
- This was a time consuming and error prone operation without automatic verification, approval sequence or feedback to the SL.
- It was designed and developed a solution using ITER Interface Operator Tools and NBTF tools with the experience from operating RFX experiment.

MDSplus SPIDER MDStree & Data Structure

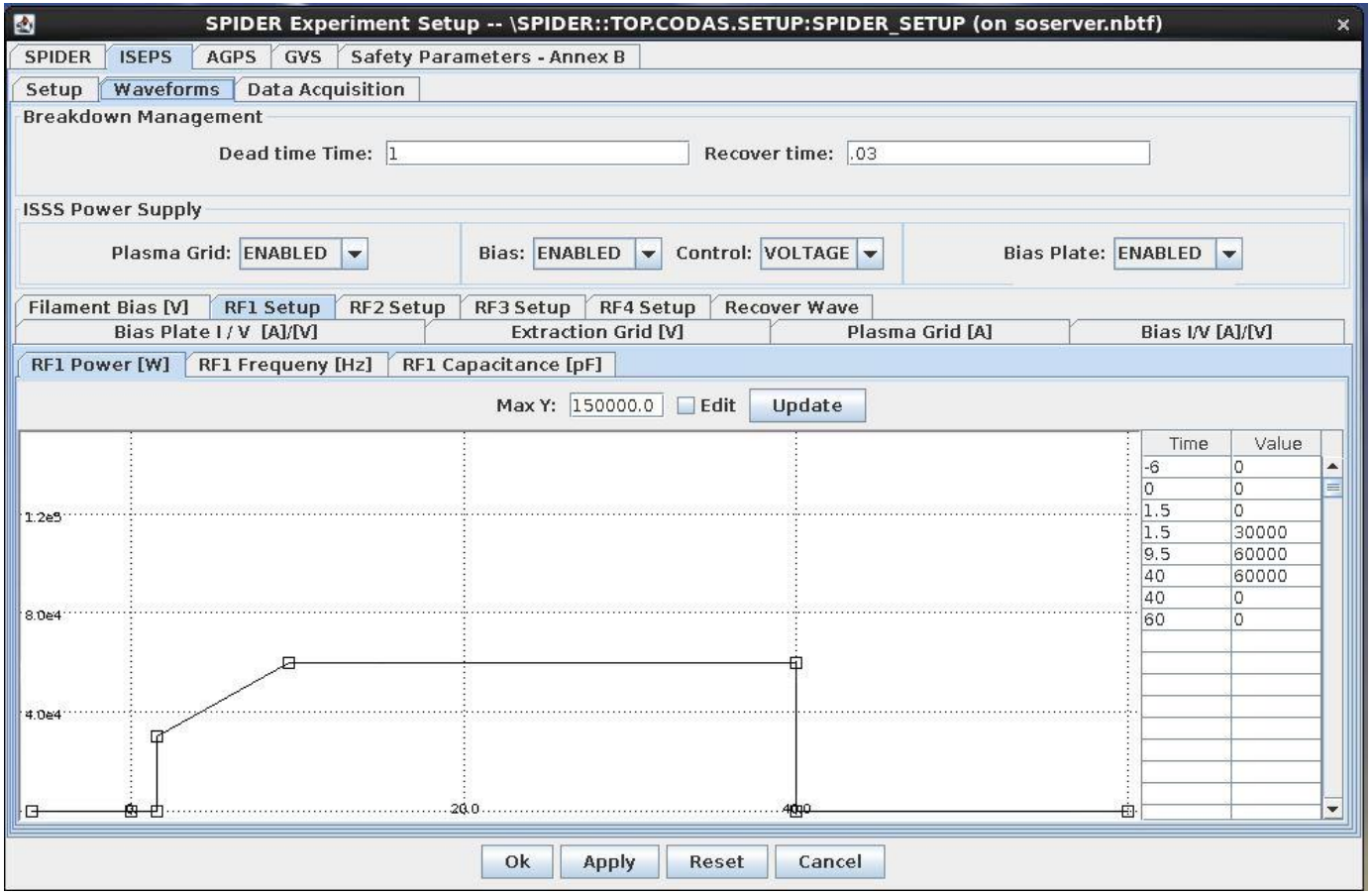




- All parameters are stored in the MDSplus SPIDER database
- Parameters can be accessed using the standard MDSplus libraries and tools

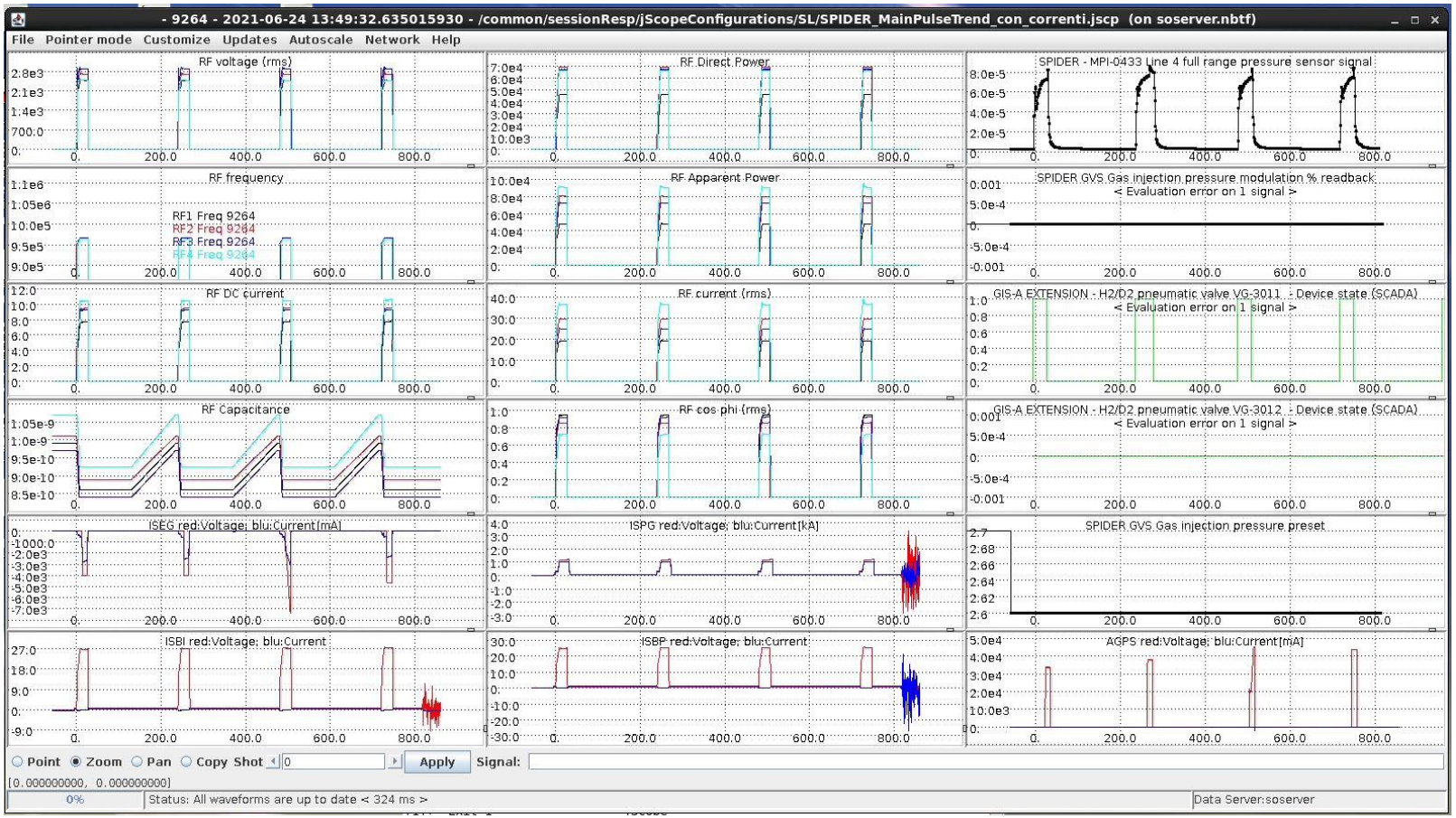


MDSplus Device Setup – SPIDER SL Configuration Tool

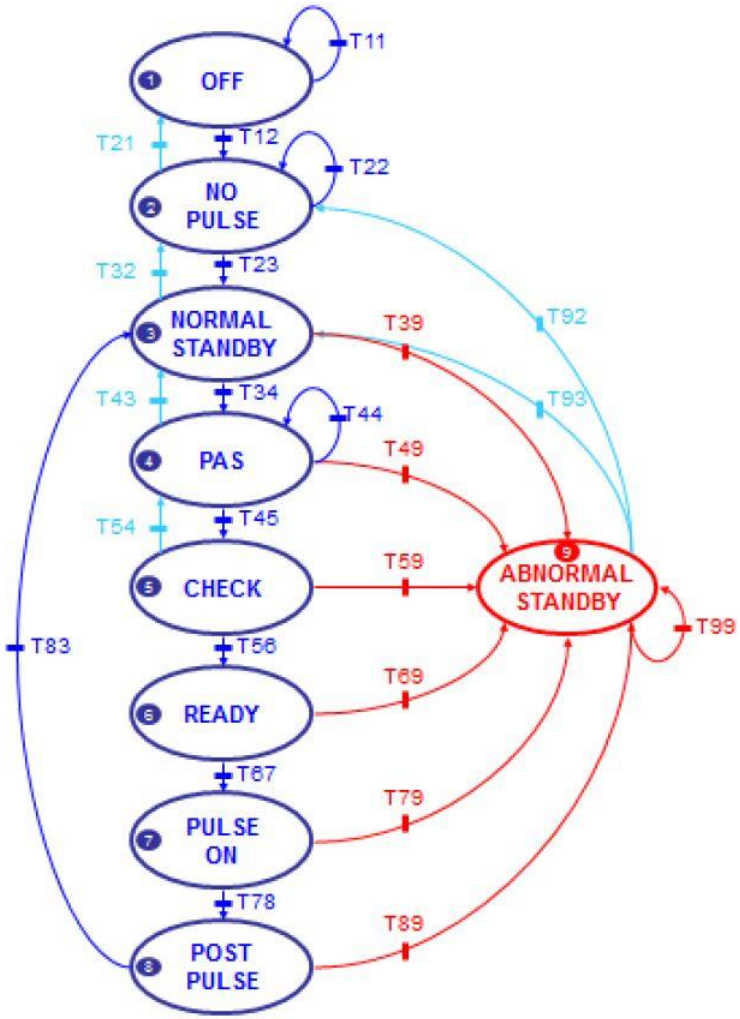


- All parameters can be set by SL using the MDSplus Device Setup tool
- The control and feed forward waveforms can be checked and changed using this tool, in case of simple waveforms
- A separate IDL tool can generate longer, repetitive and more complex waveforms

MDSplus jScope



SPIDER Pulse Sequence



- Configuration Tool shall be integrated into the SPIDER pulse sequence
- Pulse definition and approval shall be executed during NORMAL STANDBY state
- Multiple Sequence Pulses:
  - Submission to Operator Technician during NORMAL STANBY state
  - Resubmission of sequence waits for POST PULSE state to reconfigure

SPIDER Configuration Tool Requirements

SPIDER Configuration Tool Requirements

- It shall be a single tool to be used by Session Leader (SL) and Responsible Technician (RT), but with different viewer and permission settings;
- Human Machine Interface (HMI) shall be similar to the current SPIDER setup, reusing to the maximum extent the current design already developed;
- A new panel with the session operation limits for automatic verification of the setup validity;
- Include the functionality of loading configuration from a specific shot number or reference shot as a starting point for configuration changes;
- Add automatic verification if the loaded configuration is compatible with the present plant configuration;
- The application shall implement the synchronization between the setup tasks (both SL and RT) and the pulse sequence:
  - Shot number 100 shall be used to store the configuration in work (reference);
  - SL submits the parameter changes to RT before the pulse sequence is at CHECK STATE or READY STATE – These states shall inhibit the submission of a new configurations;
  - When SL submits a new configuration, the changes regarding the previous pulse shall be highlighted in the RT panel;
  - Submission of new configuration by RT to the model can only be made in PAS STATE;

SPIDER Configuration Tool Requirements

- The application shall be developed in Java or similar for Windows and Linux compatibility
- The definition of waveforms shall remain as an independent application, maintained by the SPIDER team (IDL tools)
- A new tab shall be created for definition of active diagnostics for the current pulse
- Regarding the diagnostics configuration:
  - A list of parameters for a specific diagnostic shall be integrated in the tool to be set by the SL
  - It shall be possible to load pre-defined tables of diagnostics for a certain type of pulse
  - It shall be also possible to change each diagnostic status in case of need by the SL
  - The SL shall be able to turn off certain diagnostics if they are not mandatory for the specific experimental program

SPIDER Configuration Tool Requirements

- Regarding the communication between SL and RT:
  - The submission of a new configuration by SL to RT shall be clearly assigned using a SUBMIT button
  - This action shall highlight all parameter changes in RT panel to be accepted
  - The parameters highlighted shall be with reference to the previous pulse
  - Special attention shall be taken to avoid that a new submission of parameters by SL can cancel the previous highlight of a changed field
  - Each parameter change (or a small group according to a certain logic) shall have an action or accept button to turn highlight off
  - Each tab title shall remain red (highlighted) as long as any parameters inside that tab is still highlighted
  - The final configuration can only be validated (submitted to OT) when there are no highlighted parameters or tabs
  - The final configuration is submitted by means of a SUBMIT button
  - The session limits shall be modified only by RT. The SL can check what the parameters are, but he cannot change them.

A long and detailed list of requirements was prepared.

Main SPIDER Configuration Tool Requirements

- Develop a single tool to be used by Session Leader (SL) and Responsible Technician (RT), but with different viewer and permission settings;
- Use a Human Machine Interface (HMI) similar to the previous SPIDER setup tool;
- Include a panel with the session operation limits for automatic verification of the setup validity;
- Add the feature of loading a configuration from a specific shot number or reference shot as a starting point for configuration changes;
- When SL submits a new configuration, the changes regarding the previous pulse (or a pulse loaded by RT) shall be highlighted in the RT panel;
- The definition of waveforms shall remain as an independent application, maintained by the SPIDER team (IDL tools)
- The application shall be developed using the maximum compatibility with the tools already in use and ITER Control Tools – MDSPLUS; EPICS; Control System Studio OPI
- Integrate the SPIDER pulse preparation cycle with the configuration and parameter approval, as well as SPIDER pulse sequence state machine.
- Introduce a pulse pre-approval method for the submission of automatic multiple pulses sequence

Integrating MDSplus and EPICS – EPICS CA

```
# ISEPS TAB

 record(stringin,"pvTest_$(pulse):ISEPS:EG_MOD_DC")  {
         field(DTYP,"Soft Channel")
         field(DESC,"ISEPS:EG_MOD_DC")
 }

  record(stringin,"pvTest_$(pulse):ISEPS:EG_MOD_FREQ")  {
         field(DTYP,"Soft Channel")
         field(DESC,"ISEPS:EG_MOD_FREQ")
 }

  record(stringin,"pvTest_$(pulse):ISEPS:EG_MOD_STATE")  {
         field(DTYP,"Soft Channel")
         field(DESC,"ISEPS:EG_MOD_STATE")
 }


 record(stringin,"pvTest_$(pulse):ISEPS:RF_MOD_DC")  {
         field(DTYP,"Soft Channel")
         field(DESC,"ISEPS:RF_MOD_DC")
 }
```

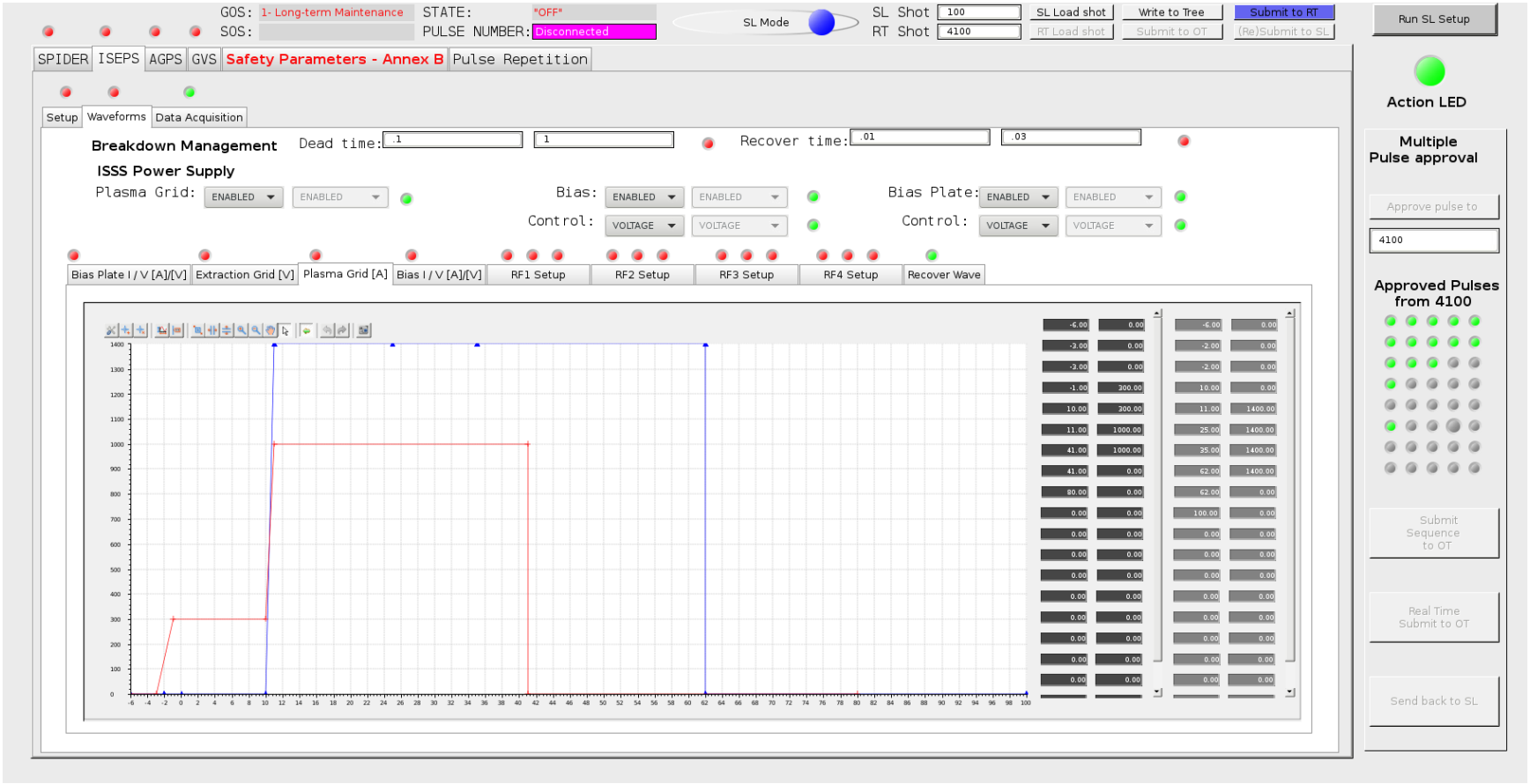
- The Experimental Physics and Industrial Control System (EPICS) has been adopted for I&C in ITER operations
- "EPICS is the software backbone of the CODAC control system." in ITER CODAC Team words
- Using a SoftIOC all necessary EPICS PVs for SPIDER configuration are created

```
#ISEPS
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:EG_MOD_DC",root+"ISEPS:EG_MOD_DC"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:EG_MOD_FREQ",root+"ISEPS:EG_MOD_FREQ"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:EG_MOD_STATE",root+"ISEPS:EG_MOD_STATE"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:RF_MOD_DC",root+"ISEPS:RF_MOD_DC"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:RF_MOD_FREQ",root+"ISEPS:RF_MOD_FREQ"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:RF_MOD_STATE",root+"ISEPS:RF_MOD_STATE"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:FIL_ON_START",root+"ISEPS:FIL_ON_START"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:FIL_ON_STOP",root+"ISEPS:FIL_ON_STOP"))
list.append(MDS_PVs("\\TOP:CODAS:SETUP:SPIDER_SETUP:ISEPS:FIL_STATE",root+"ISEPS:FIL_STATE"))
```

- Python scripts make the link between MDSplus database and EPICS CA
- Each MDSplus node corresponds to an EPICS PV

Integrating MDSplus and EPICS – CSS OPI

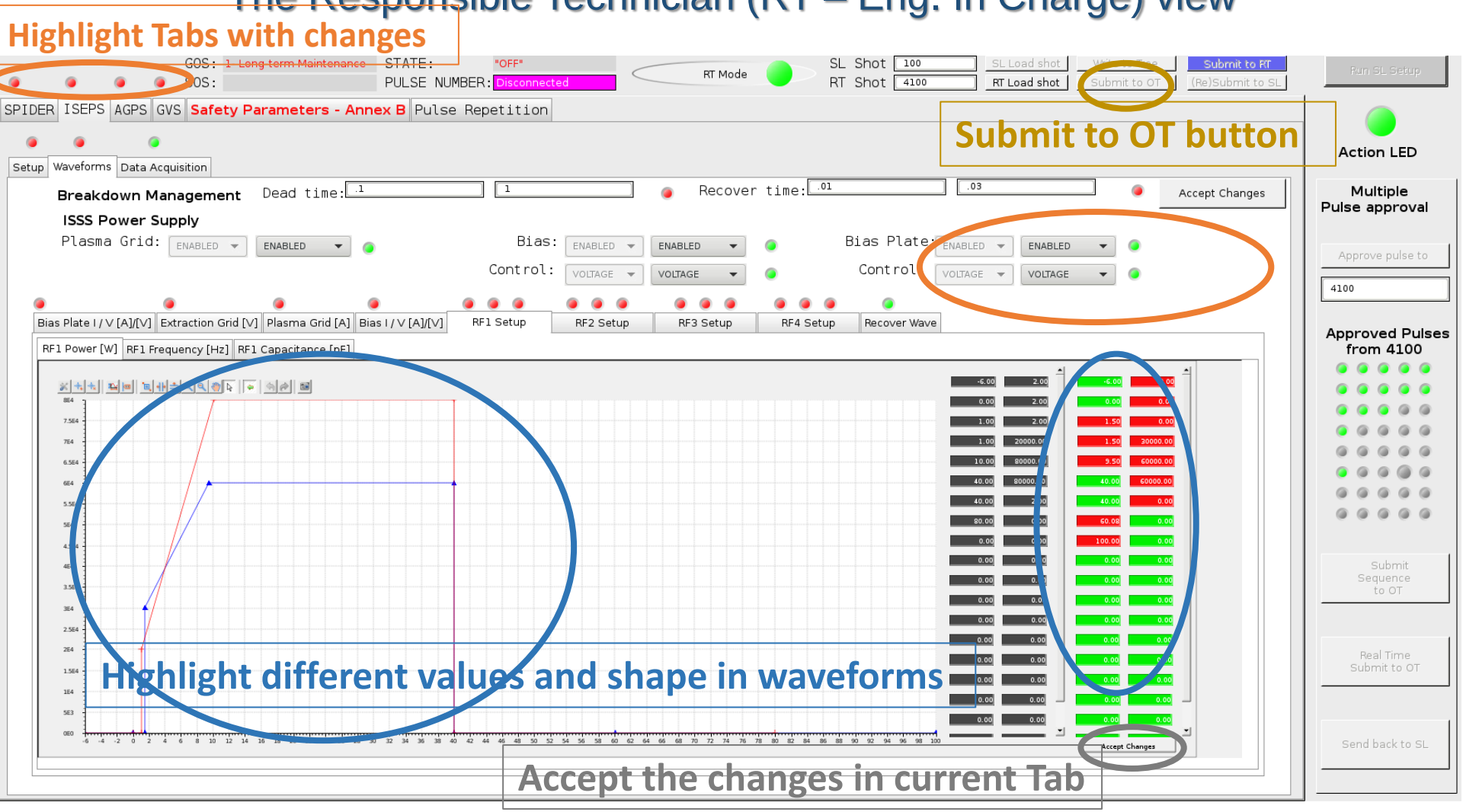
The Session Leader (SL – Scientific Coordinator) view



"Control System Studio (CS-Studio) will power ITER's dashboard, what we call the visualization layer—panels, graphs, sliders, symbols, metres and switches on the operator consoles in the ITER control room."

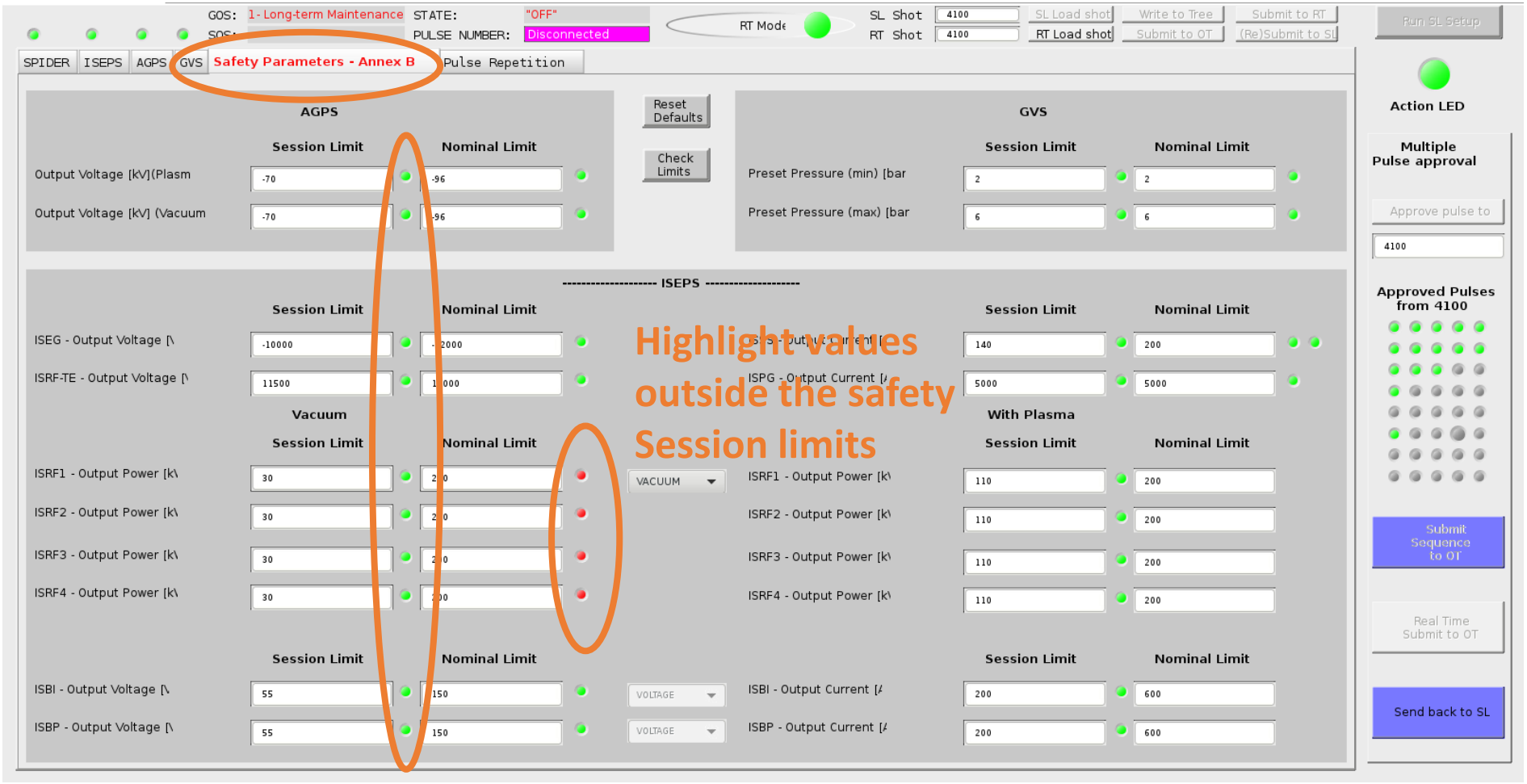
Integrating MDSplus and EPICS – CS-Studio OPI

The Responsible Technician (RT – Eng. In Charge) view



SPIDER Config. Tool: Safety Parameters Checking

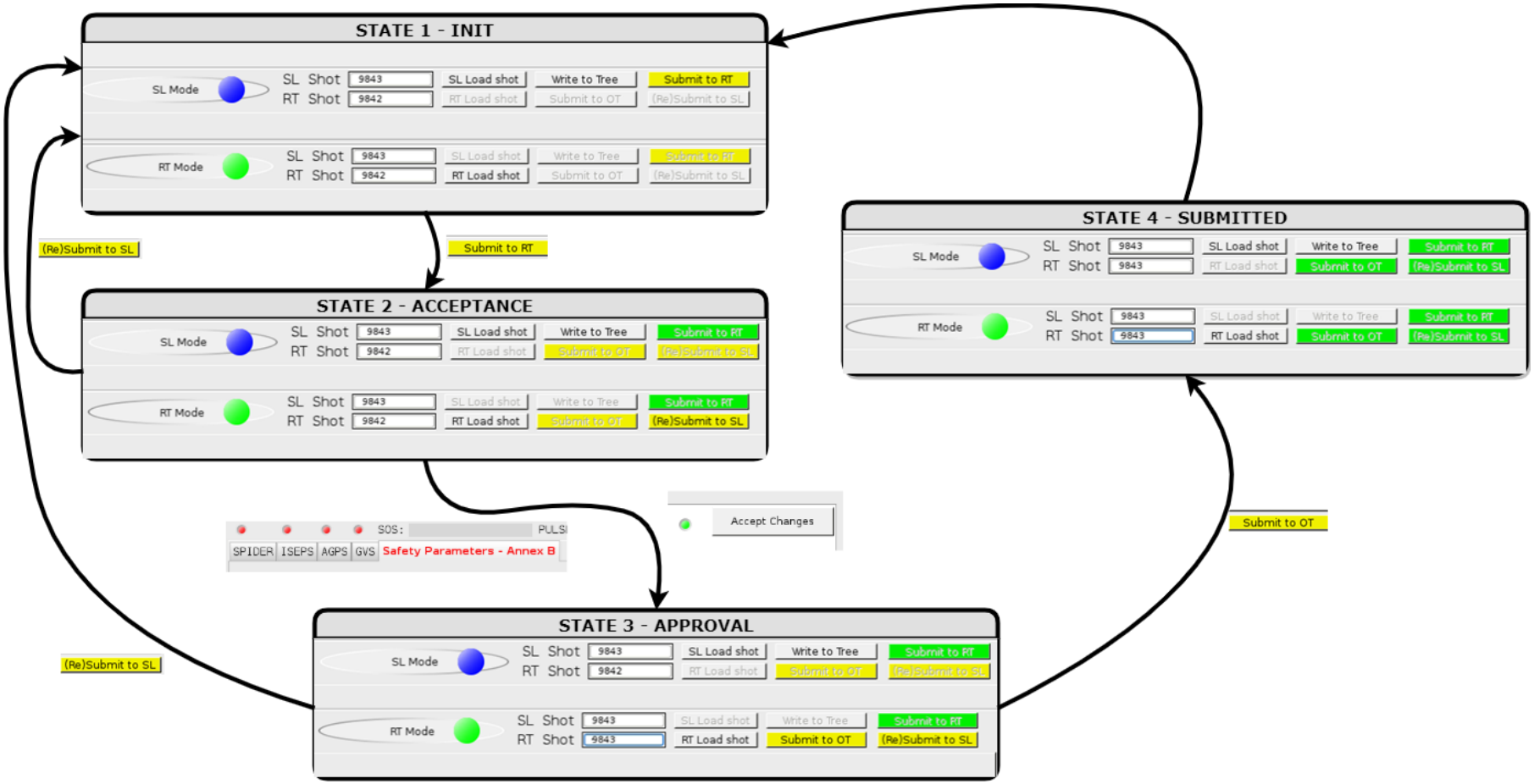
Highlight Tab with errors



Highlight values outside the safety Session limits

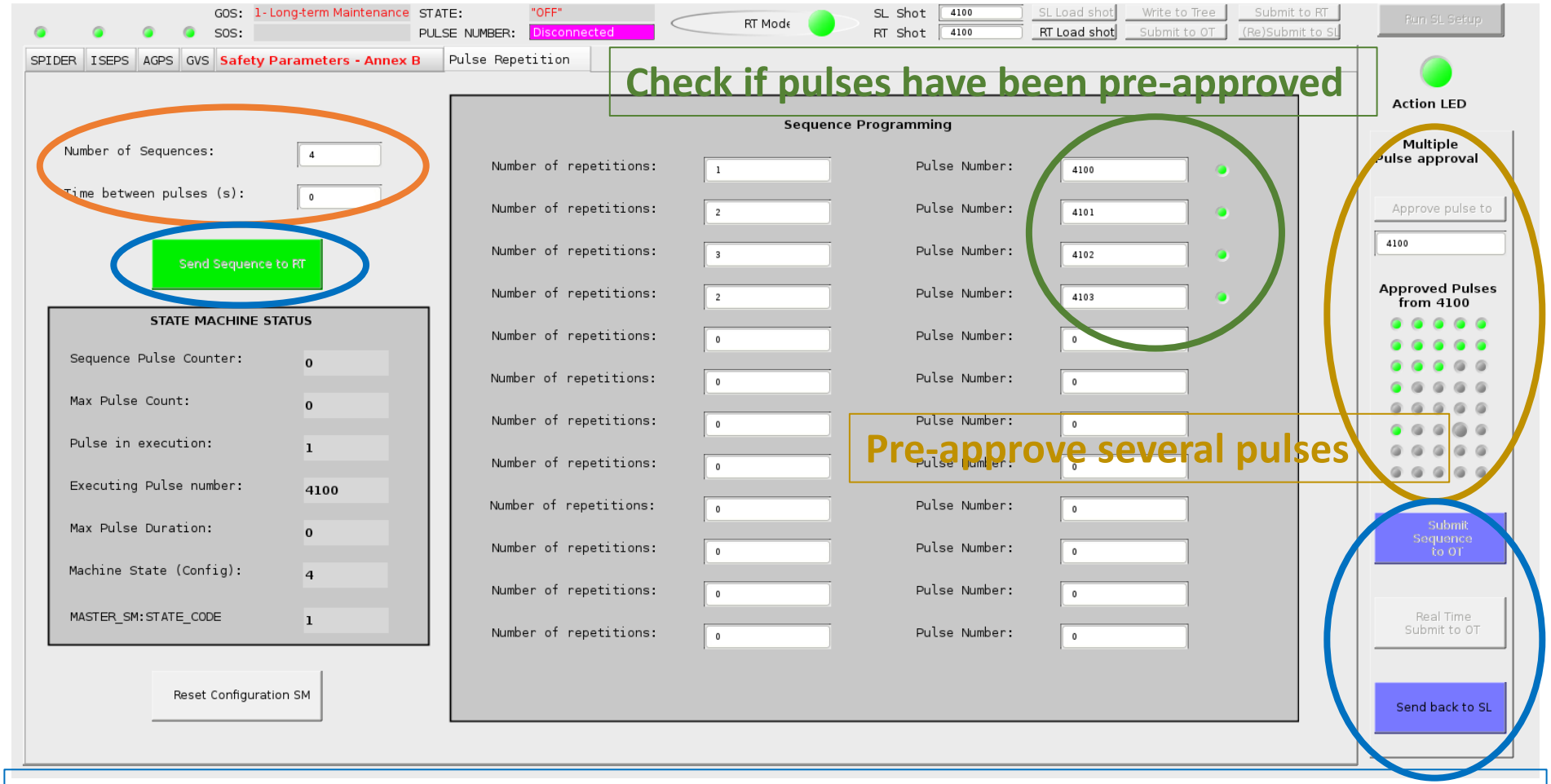
Highlight the Session Limits are inside the Nominal Limits

SPIDER Config. Tool: Single Pulse State Machine



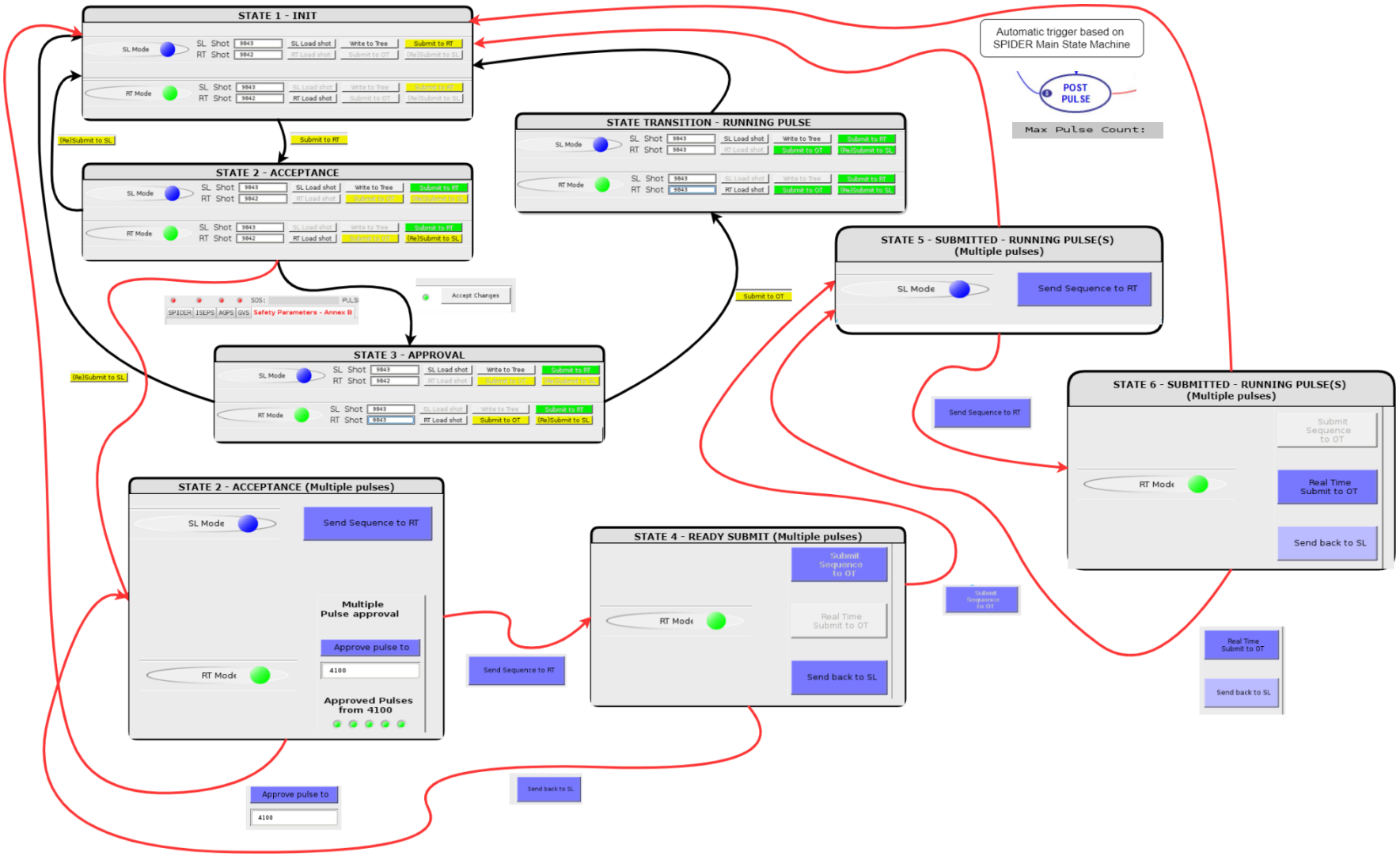
SPIDER Config. Tool: Multiple Pulse Programming

Program number of Sequences and time interval between discharges



Actuate on the State-Machine by submitting and sending actions to other operators

SPIDER Config. Tool: Multiple Pulse State Machine



SPIDER Tool – HMI importance

- Color Code (relevant for efficiency and fast operation):
  - Green button – Action completed
  - Blue button – Active action to be submitted
    - White text – disabled button: not all conditions are met or operator has no permission
    - Black text – operator can submit action
  - Grey button – Not relevant for state transition

Summary & Conclusions

- Requirement definition based on the needs of SPIDER operation has been completed
- The definition of development tools available based on ITER CODAC Plant System I&C requirements has been set
- Integration of MDSplus and EPICS for a common development environment of the SPIDER configuration HMI
- Tool implementation with the SPIDER requirements, following ITER guidelines for I&C uniformization:
  - Automatic verification of safety parameters (session limits)
  - Multiple pulse approval and automatic real-time reconfiguration of pulse sequence
  - Full integration with SPIDER pulse sequence
- Beta version of the tool has been tested and presented to operators
- Live tests @ SPIDER control room ready - after current shutdown

Acknowledgements

This work has been carried out within the framework of the **EUROfusion Consortium** and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

The work leading to this presentation has been funded partially by **ITER** under activity work programs AWP2020-2021. This publication reflects the views only of the authors and ITER Organization cannot be held responsible for any use, which may be made of the information contained therein. The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

IPFN activities received financial support from ***Fundação para a Ciência e Tecnologia*** through projects UIDB/50010/2020 and UIDP/50010/2020.