

The ASDEX Upgrade Discharge Schedule

G. Neu, K. Engelhardt, G. Raupp, W. Treutterer, D. Zasche, T. Zehetbauer, and AUG Team
Max-Planck-Institut für Plasmaphysik, EURATOM Association, Boltzmannstrasse 2, D-85748
Garching, Germany
gregor.neu@ipp.mpg.de

Abstract:

ASDEX Upgrade's recently commissioned Discharge Control System (DCS) marks the transition from a traditional programmed system to a highly flexible „data driven“ one. The allocation of application processes (APs) to controllers, the interconnection of APs through uniquely named signals, and AP control parameter values are all defined as data, and can easily be adapted to the requirements of a particular discharge.

The data is laid down in a set of XML documents which APs request via HTTP from a configuration server before a discharge. The use of XML allows for easy parsing, and structural validation through (XSD) schemas.

The central input to the configuration process is the Discharge Schedule (DS), which embodies the dynamic behaviour of a planned discharge as reference trajectories grouped in segments, concatenated through transition conditions.

Editing, generation and validation tools, and version control through CVS allow for efficient management of DSs.

System Overview and Introduction:

Discharge Control System

ASDEX Upgrade's (AUG) fully digital Discharge Control System (DCS), is realized as a cluster of off-the-shelf PCI controllers, using custom built I/O boards for peripheral I/O, and interconnected via a distributed shared memory (DSM) network.

To execute a discharge, DCS monitors and controls numerous technical and physical quantities by evaluating measurements, comparing them to reference values, computing commands, and outputting them to AUGs actuator systems (fig. 1).

The autonomous machine control system (MCS), a collection of SIMATIC programmable logic controllers, configures and continuously operates the actuator systems. Actuator systems include power supplies, poloidal and toroidal coils systems, and a multitude of heating, and fuelling systems.

Measurements come either directly as real-time signals, provided by real-time diagnostics over a gateway connected to the DSM, or by computing them in dedicated input processes from sensors configured by conventional diagnostics. The sensor's characteristics are made available in proprietary description files, the diagnostic shotfiles, before a discharge. Diagnostics may have several operation modes, which may vary during a discharge, and whose transitions can be

triggered through special events issued by the DCS.

Application Processes of DCS:

The application processes (APs) running on DCS are organized into a group of traditional processes, e.g. for monitoring and feedback controlling a wide set of technical and plasma parameters, and a control layer, consisting of the Reference Value Injector (RVI), and the Segment Scheduler (SS)(figure 2). A detailed description of AP organisation and control schemes can be found in [1].

Whereas RVI provides the trajectory values of reference signals for all APs, SS continuously supervises the course of the discharge, and, upon the occurrence of pre-defined transition conditions, instructs RVI to switch to a new set of trajectories by issuing a new segment number. A transition condition is defined as a time window, a boolean expression evaluating real-time signal values, and the target segment number to be published when the expression is true.

The advantages of having a single RVI process to compute and output reference trajectories are threefold: APs can be simplified (and thus rendered more robust) by relieving them from the task of tracking (or interpolating) waveforms in a coordinated manner, the implementation of new tracking algorithms only affect one process, and, most importantly, RVI provides a well-defined point of intervention to override, or modify the reference values through the use of a pre-processor to the trajectory tracking [1]. The latter feature makes it possible to react to instabilities or to automate soft landing, and also forms the base for the development of real-time discharge optimisation strategies.

DCS configuration

DCS is highly configurable, or “data driven” through a number of XML files [2]: the allocation of APs to controllers, for instance, can easily be defined by editing the System Release file. In specific AP descriptor files (APdesc), all quantities processed by the algorithms implemented in the APs can be related to uniquely named real-time signals and system parameters, such as feedback gain matrices, thresholds, algorithmic constants. This feature allows to redirect inputs to alternative sources, when new, improved measurements become available, or for testing purposes. The values for system parameters stem from databases, and, together with those related to the configuration of the actuator and diagnostic systems, are compiled into the descriptor files just before a discharge. The result of this compilation, performed by a configuration server, is a complete AP configuration file (APconf)

The descriptor file of the Segment Scheduler (SSdesc) holds a set of programmes, expressed in a stack machine language, which correspond to the condition expressions mentioned above. The RVI also has a descriptor file (RVIdesc) which contains parameters for the pre-processor.

The use of XML for representing descriptor and configuration files offers several advantages: human readability, structural (and some semantic) validation through XML Schema Definition (XSD), easy extraction of information through freely available parsers for several platforms and

programming languages, XSLT transformation into HTML for web visualisation of the contents, and the immediate availability of standard management tools (version control, visualisation of modifications, etc.)

Discharge Schedule:

Whereas the APdescs, SSdesc , and RVIdesc describe the static parameterisation of the DCS, its dynamic behaviour, as expected by the experiment leader, is entirely determined by the Discharge Schedule. More specifically, the DS serves as input for the configuration of RVI and SS. (figure 3)

The DS is structured into segments, which represent distinct phases of the discharge, for instance, ramp-up of toroidal field (TF) generator, and loading of the central solenoid capacitor, ramp-up of TF current, ramp-up of poloidal field currents, ignition, start of position control and plasma flattop, ramp-down of plasma and coils, end of TF flattop.

For each segment, the DS (fig. 4) contains signal trajectories with time-tagged setpoints and tracking rules from which RVI can compute the reference values for a specified real-time signal . Currently only two tracking rules are implemented: linear interpolation, and step function. Higher order polynomial interpolation, and function-generator-like generation (e.g. sawteeth or sine waves of a given amplitude and phase) could be implemented in future. Besides the value, also the data-type, and dimension (for vectors or higher order arrays) are specified for each setpoint element. An entry rule which defines how the RVI is to proceed before the first setpoint for a signal trajectory is defined in a segment, and an exit rule, defining how the RVI is to proceed if the segment extends beyond the last given setpoint, complete the signal trajectory element.

Each segment also holds at least the watchdog condition (as attributes: time and target segment) which automatically leads to switching to a follow-up segment after the specified watchdog time has elapsed, and, possibly a supervision section with additional transition conditions.

In addition to the segment specific conditions, two additional “global” conditions exist, which lead to immediate pulse termination, or soft landing [3].

Every DS has header information which is not aimed to be processed by the DCS but rather serves administrative purposes. These include the DSs name, a comment indicating its purpose, the id of the creator, and the date of creation. Additional information is not directly represented but available through the version control management tool (see below).

Discharge Schedule Management:

DS editing:

A special JAVA tool, the Discharge Schedule Editor, is available for experiment operators to edit the DSs and select which DS should be executed on the next shot.

An editing window presents a time-ordered table of setpoints for a subset of signal trajectories

in a selected segment. The user may modify the values of the trajectory setpoints at a given time, remove single points or all points at a given time, insert new points or modify existing time values. The groups of signals to be edited together are organized in a set of named views, which can be flexibly configured by each user, by selecting and ordering the set of signals to be displayed. Collections of views can be saved in and retrieved from one or more user set-up files. Standard views exist for fuelling, heating, position and shape control, and plasma performance control. During an edit session the user may switch from one view to another, and select which segment from the current DS to display. A function to show a plot window to display the signals of the current view in graphical form is provided.

Currently all editing is manual, the editing relies on entries in a real-time-signal database which specifies the value range for continuous signals, and the position names and values for switch signals.

Special functions such as setting the segment watchdog timer or adding a comment to the discharge programme are supported.

In future, the editor will be enhanced by generation and verification modules, which the user can activate or switch off during an editing session. All modules have inputs (signals and parameters), triggering conditions (specific combinations of values for the inputs) and execution code which embodies the activity of the module. Generation modules will, as their name implies, generate output trajectory values, which are inserted in the DS, whereas verification modules will return a boolean value indicating whether the verification was successful, and possibly a message explaining the reason of an unsuccessful verification. Examples of the former could be the automatic generation of timing sequences, or the adaptation of the poloidal field current trajectories to a modified plasma current. Verification modules could range from simple checking of actuator reference values against their time varying monitoring ranges, to verifying that expected tilt forces lie within the limits permitted by the machine.

A comparison function, to display the differences between DSs, an undo/redo feature, and a function for including/referencing segments from other DSs, will also be provided by the next release of the editor

Storage and Version Control:

From the editor, modified DSs can be saved to a disk file temporarily, or submitted to the official version-controlled repository (CVS), with an appropriate comment, if considered sufficiently stable for execution. A submitted DS can additionally be tagged for execution, which notifies the DCS to load and execute this schedule on the next discharge or “shot”.

Existing DSs can be loaded from disk files, or particular versions retrieved from the repository. It is also possible to retrieve the DS executed for a past discharge by specifying its shot number.

The DS repository, currently containing about 200 DSs, is organised in directories, representing categories of discharges, such as Standard, Beta Scaling, Edge Physics, Improved H-Mode, or

NTM stabilization.

Pre- and Postshot Processing

Before a discharge is executed, the DS is processed to extract information to be displayed on the operator user interface, and to perform validation procedures.

Examples of the latter are the scanning of the reference trajectories for coil currents, to see if these lie within the ranges defined by the coil actuator systems, or the verification of the correct sequencing of the timing event trajectory, driving the behaviour of diagnostics and power supply systems. The segment transition conditions are also checked to signal possible loops or non-existent target segments.

When the SS requests its configuration file, the configuration server extracts supervision information from the DS and merges it into the SSdesc. The DS's signal trajectories are merged with the RVIdesc to complete the configuration data for the RVI (fig. 3).

Finally, in a publish & subscribe phase, where all APs (including the RVI) issue lists of produced and required signals, the completeness of the DS with respect to the requirements of the APs of the current system release is asserted. The operators are warned when the DS contains reference trajectories for signals which are not required by any AP, indicating that the DS was possibly designed for a different system release.

After execution of a discharge, the corresponding DS is tagged with the shot number for later retrieval in the editor.

Conclusion:

References:

- [1] W. Treutterer et al., "ASDEX upgrade's new plasma control scheme", Fusion Engineering and Design, Proceedings of the 5th IAEA TM on Control, Data Acquisition, and Remote Participation for Fusion Research, Volume 81, Issues 15-17, July 2006, Pages 1927-1931, <http://dx.doi.org/10.1016/j.fusengdes.2006.04.010>
- [2] G. Neu et al., "ASDEX Upgrade Tokamak Control Parameter Organization and Configuration Management", paper WP527, ICALEPCS '03, Gyeongju, Korea
- [3] G. Raupp, W. Treutterer, et al., Control Processes and Machine Protection on ASDEX Upgrade, this conference

Figures

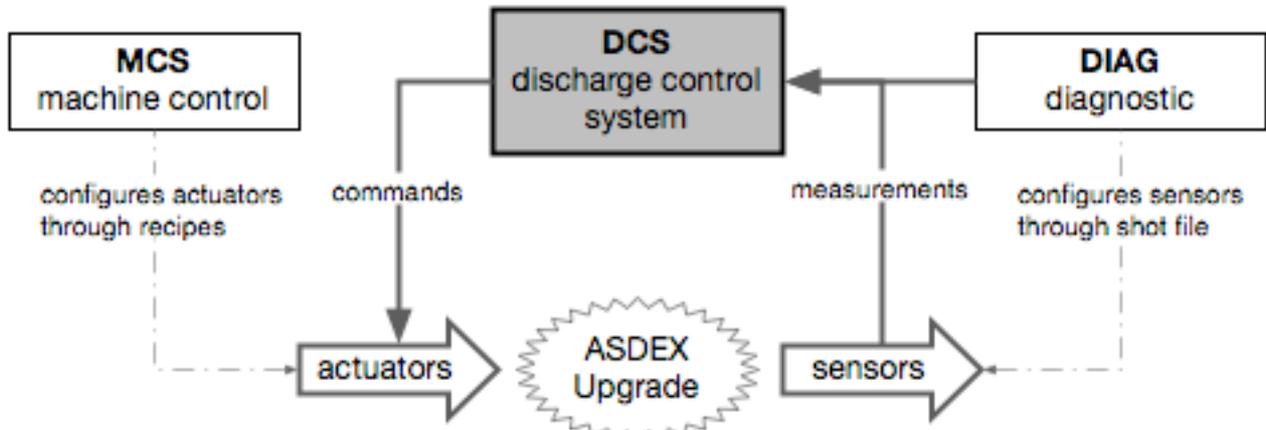


Figure 1: Systems Overview

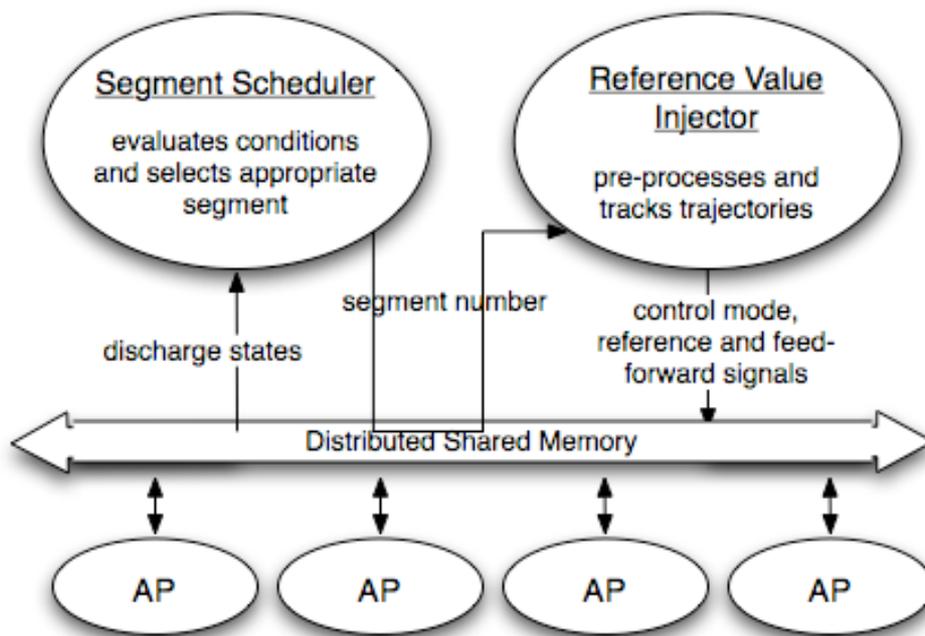


Figure 2: DCS Application Processes and Control Layer

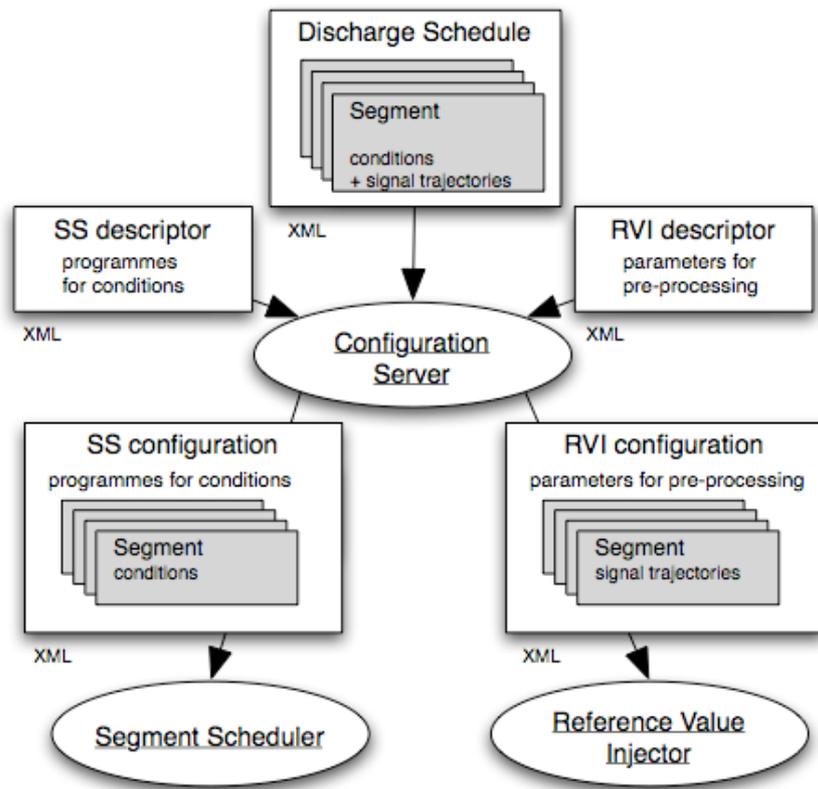


Figure 3: Control layer configuration

```

<DP_DSC xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="DP_StdH_1MA.xml"
  comment="1.0 MA; Standard H-mode, T1,T4 1.5-6s"
  created_by="rln"
  creation_date="Wed Jun  3 14:43:25 1992"
  last_modified="Tue Apr 04 09:28:57 MEST 2006"
>
<SEGMENTS>
  ...
  <SEGMENT name="BZ155/Start TF Gen/OH-Kond.laden"
    wd_time="600.0" wd_target="BZ205/Start TF Strom">

    <Supervision>
      <Condition program="Generator ready"
        startTime="0.0" endTime="600.1"
        targetSegment="BZ205/Start TF Strom"/>
    </Supervision>

    <SIGNAL_TRAJECTORIES>
      <SIGNAL_TRAJECTORY name="rts:Act/EV/TSE.dp">
        <ENTRY_RULE is="None"/>
        <EXIT_RULE is="Last"/>
        <EXECUTION_RULE is="Step"/>
        <REFERENCE>
          <POINT time="0.005001">
            <VALUE type="uint32_t" dimension="1">0x4f840000</VALUE>
            <!-- Timing event TS1, Generator to High -->
          </POINT>
        </REFERENCE>
      </SIGNAL_TRAJECTORY>
      ...
    </SIGNAL_TRAJECTORIES>

  </SEGMENT>
</SEGMENTS>
</DP_DSC>

```

Figure 4: Discharge Schedule (simplified)