Event Detection and Exception Handling strategies in the ASDEX Upgrade discharge control system

W. Treutterer, G. Neu, C. Rapson, G. Raupp, D. Zasche, T. Zehetbauer and ASDEX Upgrade Team

Max-Planck-Institut für Plasmaphysik, EURATOM Association,
Boltzmannstraße 2, 85748 Garching, Germany

Thermonuclear plasmas are governed by nonlinear characteristics: plasma operation can be classified into scenarios with pronounced features like L and H-mode, ELMs or MHD activity. Transitions between them may be treated like discrete events. Similarly, technical systems are also subject to discrete events such as failure of measurement sensors, actuator saturation or violation of machine and plant operation limits.

Such situations often are handled with a mixture of pulse abortion and iteratively improved pulse schedule reference programming. In case of protection-relevant events the complexity of even a medium-sized device as ASDEX Upgrade, however, requires a sophisticated and coordinated shutdown procedure rather than a simple stop of the pulse. The detection of discrete events and their intelligent handling by the control system has been shown to be valuable also in terms of saving experiment time and cost.

This paper outlines how ASDEX Upgrade's discharge control system (DCS) detects and deals with changes in the discrete system state. DCS performs event detection and exception handling in two stages: locally and centrally. The goal of local handling is to limit the effect of an exception to a minimal part of the controlled system. Thus, local exception handling facilitates robustness to failures but keeps the decision structures lean. A central state machine deals with exceptions requiring coordinated action of multiple control components. DCS implements the state machine by means of pulse schedule segments containing pre-programmed waveforms to define discharge goal and control method within a time window. Segments also include conditions to define exceptions and associated segment branching instructions. Thus, the state machine logic can automatically adapt to plasma and plant state.

Keywords: real-time control, event detection, exception handling, protection, system architecture

1. Introduction

ITER's quest to define the future functionalities of its control system is also stimulating the discussion among existing fusion experiments about best practices, which could be suitable and extended also for ITER. Exception handling is one of the areas, which has attracted special interest [1].

Each of the experimental devices has developed recipes for trouble-shooting in case of unforeseen problems. However, up to now, these methods have only rarely been in the focus of publications. Recently this situation has changed. Papers about new approaches for the Tore Supra event and exception handling in long discharges [2, 3] or the JET protection concept to protect the ITER like wall [4] describe issues characterized by complex relationships requiring non-trivial solutions. The ASDEX Upgrade methodology has also attracted attention in this context because of the effectiveness and versatility of its principles. This paper tries to contribute to the ongoing ITER development explaining the fundamental ideas of exception detection and handling in the ASDEX Upgrade Discharge Control System (DCS) as well as showing the impact of these concepts on the control system architecture.

Exception handling originated as a method to solve problems that appear unexpectedly and prevent achieving the nominal goal of an investigation. Technical failures are unfortunately an unavoidable occurrence and control algorithms can break if they are used under different conditions than assumed during their design. Finally, plasma physics with its multifaceted nonlinear characteristics frequently gives rise to changing operating conditions during a discharge. In general all those examples can be interpreted as changes of a discrete state – i.e. events. On the other hand, it is not always possible to detect the real source of events. Rather, only symptoms of such events can be observed. Even more, a control system will only try to recognize the symptoms of those events requiring a reaction. In order to distinguish the underlying event from the perceived symptom we use the term “exception” for the latter.

A comparison with medicine reveals some expedient analogies. An accident can severely deteriorate the normal condition of a human being. Injuries are diagnosed and a therapy is applied to cure them. The accident corresponds to an event, the diagnosed injuries to exceptions and the therapy to exception handling. In many cases, it is not even necessary to know accident details and it is the injuries that determine the therapy. Likewise, control systems handle exceptions rather than the underlying events. The analogy can even be used for principal methods of exception handling. For small accidents, it might be sufficient to apply an adhesive bandage covering the wound. This "first aid" can be accomplished by the injured person itself or by a first responder avoiding the necessity of visiting a hospital. Depending on the injury, however, first aid is only the first step and an elaborate medical exploration and treatment in a hospital is indispensable. The same pattern also appears in the exception handling philosophy of DCS. Local exception handling takes the part of rendering first aid. For complex diagnosis and therapy involving different system components central exception handling is the method of choice. Even if the distinction of the best approach is sometimes blurred, both methods complement each other.

In section 2 we will describe the relevant pieces of DCS control system architecture. In sections 3 and 4 local and central exception handling are explained in more detail. Finally section 5 shows the potential for future applications.

2. DCS Architecture

DCS is the second-generation control system of the ASDEX Upgrade experiment. Its architectural design is

Wolfgang.Treutterer@ipp.mpg.de

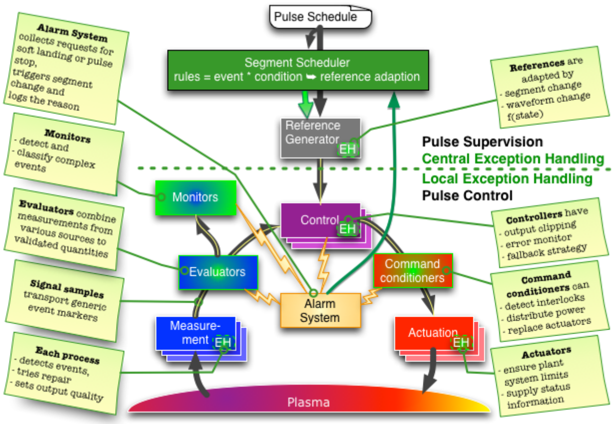

Fig. 1: DCS components with exception handling

founded on experience from the previous version [5, 6, 7, 8]. From the beginning, event detection and exception handling were essential requirements and had a big influence on conceptual decisions. Some of the basic principles: function separation, modularity and generalisation are not only important features for a transparent, flexible, scalable and evolutionary system. These properties manifest themselves also in the methods dealing with off-normal situations.

One big structural decision in DCS is the separation between pulse control and pulse supervision. Pulse control comprises the classical components of a feedback control system: measurement, monitoring and control and command output. Besides their nominal functionality each of the components is also prepared to handle off-normal cases locally. Pulse supervision has the role of a pulse manager. It interprets the pulse schedule and takes automated decisions to realise the goals defined therein. The ASDEX Upgrade control system keeps the classical pulse control functions with their fixed and proven functionalities strictly separated from the highly flexible and dynamic pulse supervision. Pulse control functions communicate with pulse supervision only via reference and command signal samples and know nothing about waveforms and how they are generated. That way, the frequent modifications in pulse supervision specifications do not require any accompanying functional adaption of pulse control. Fig. 1 gives an overview on DCS control system modules and their exception handling capabilities as described in the following sections.

2.1 Pulse Control

Measurement modules include interfaces to data acquisition and diagnostic systems as well as reconstruction algorithms. Diagnostic specific knowledge can be used to detect malfunctions and to substitute unusable data based on redundant information, if available. Monitoring and control modules process input data, process states and calculate commands to actuators in order to realize desired behavior or to protect the device. They have a wider system scope, use data from various sources and thus have the ability to operate on more complex events and exceptions. Command output modules are responsible to send control system output signals to actuating systems and to exchange state information. Command output may include command conditioning, where analog command amplitude is transformed to one or more actuator commands. Command conditioning has a decoupling function between controller and actuator(s). It can establish a balance between particular actuator requirements like relay characteristics and generic controllers with continuous command amplitudes (e.g. PI controller). NBI output exhibits two examples for command conditioning. Continuous command amplitude for the cumulated heating power from the poloidal beta controller is first distributed over a number of beam sources according to source availability and a priority scheme. Subsequently the power amplitude for each source is translated via pulse width modulation into a series of power on/off instructions.

2.2 Pulse Supervision

Pulse supervision accommodates that fusion plasma devices are operated with waveforms changing over time, unlike most industrial plants, which use constant set points. These waveforms are an essential part of the pulse schedule, which defines the reference behavior of a pulse. Each of the different pulse phases are characterized by a distinct physical and technical behavior requiring adaption of the applied control functions as well as their references. The adaption rules in principle form a second, nonlinear feedback control loop. Designing these rules is often an experimental process, such that rules can be volatile.

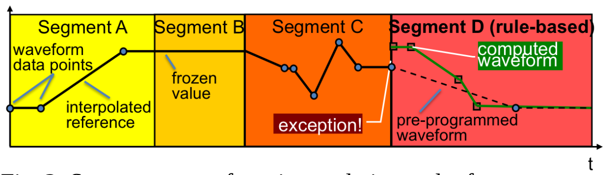

Fig. 2: Segments, waveform interpolation and reference adaption in segment D (see section 4.1)

Pulse supervision consists of a reference generator and a segment scheduler, both based on data structures called segments. The reference generator performs interpolation for any time between two data points in a waveform as illustrated in Fig. 2. Waveforms are given as a sequence of time-value data points. DCS allows for various interpolation techniques, linear, step and pulse. After the last data point the final value will be carried forward. The waveforms in an ASDEX Upgrade pulse schedule are structured into segments. Especially the non-plasma phases are described by different segments: initialization, wait for fly-wheel generator spin-up, toroidal field ramp-up, poloidal field pre-magnetization and central solenoid ramp-up, breakdown, plasma plasma ramp-up, plasma flattop, plasma termination, toroidal field ramp down, cool-down, end of pulse. The time in the data points is interpreted relative to the starting time of the segment. This allows segments to be re-arranged and re-used in different pulses. The common segment time creates coherence between otherwise unrelated quantities. Thus, a segment can be seen as a procedure description aiming at an overall goal.

In addition, segments include a number of branching conditions and associated target segment identifiers. These determine the period of validity of the current segment as well as implicitly the goals of the follow-up target segments. In its simplest form a chain of segments makes up an ASDEX Upgrade pulse with a time condition leading to the next segment in the chain. A special notation, the "common segment" serves as a container for default conditions, which are evaluated in all segments and primarily used for device protection. The segment scheduler has the task to evaluate the conditions of the currently active segment and branch to the respective target segment, if one of the conditions applies.

Wolfgang.Treutterer@ipp.mpg.de

Pulse supervision has two main handles to react to exceptions: automated adaption of waveforms or changing the active segment and, thus, the current goal. Since these reactions occur on the central pulse management layer, the technique is also called central exception handling. While segment branching started as a pulse structuring method, it developed into a powerful tool for exception handling and pulse optimization. More details will be presented in section 4.

3. Local Exception Handling

Local exception handling was introduced as a tool for problem solving. Experimenting consists in the execution of a plan under the assumption of a certain response of the device. Problems manifest themselves through exceptions as deviations from the expected (normal) behavior. Local exception handling makes an attempt to solve problems within the affected function component. This approach supports a major architectural design goal: keeping system components reusable by obeying the principles of encapsulation and abstraction.

The analysis of frequent problem sources shows that many important exceptions occur inside or in the direct vicinity of control system components. These exceptions either belong to the category of technical faults or are the result of design assumptions, which are not valid in the actual experiment context. Sometimes, an alternative algorithm better suited to the current state can be applied instead. An input value range specification, for example, is an indicator for such assumptions. The range often represents plausibility limits. A frequent form of invalid assumptions manifests itself in mathematical faults. Algorithms sometimes use mathematical operations with limited input data range (e.g. division: denominator <> 0; square root, logarithm: argument > 0). If experiments are run near the boundary of the valid region measurement noise or spikes or limited numerical resolution (quantization) can lead to invalid input arguments. Such offending arguments and inputs can, however, be easily detected and often solved by algorithm specific workarounds.

3.1 Basic Exception Handling

Basic exception handling forms the fundament of local exception handling. It attempts to fix problems, which are rooted in control system components at or near the source. Each component is specialized to solve a certain task. Thus, it is the natural place to store knowledge about problems and possible solutions. This strategy helps to prevent problem propagation. Other system components can continue to work as normal. Moreover, decision structures stay contained. System components need only to solve problems, which occur in their own scope. Diagnostic systems have expert knowledge of the (plasma) physics they observe, of the data acquisition and its possible faults and of the redundancy potential that can be used for repair. Actuator systems, on the other hand, have expert knowledge on the (plasma) physics they act on, on their own technical limits and peculiarities and on replacement options with other actuators of the same type. Feedback controllers can recursively fall back to specified save modes, and, at least, to feedforward operation. Also each control function can know about its design limitations (assumptions) and have built-in replacement strategies using inputs from alternative sources.

Sometimes, a problem cannot be fixed by a single control system component and degradation of its output signals cannot be avoided. In such cases basic exception handling allows to forward exceptions. But a function has only a narrow scope. Since during a pulse the composition of the control chain can be dynamically adapted to the actual state, general-purpose functions cannot know whether their problem is significant to other currently important components. Therefore, all ASDEX Upgrade components attach quality information to all samples of their output signals. These tags called confidence state can take the values GOOD, CORRECTED, RAW or INVALID. Thus, the problem in a generalized form but not the handling strategy is forwarded to subsequent functions. Each subsequent function checks the quality information, raises an exception, if it is unacceptable, and chooses an appropriate reaction strategy. Finally the result is output, again with an appropriate quality tag. An example is the use of plasma current profile information. The profile requires polarization angles from the real-time MSE diagnostic. For measurement an auxiliary neutral beam is required. Without this beam the samples for the polarization angles are INVALID. A subsequent profile reconstructor thus can also just produce invalid samples and a profile controller suspends falls back to feedforward mode due to invalid input signals. As soon as the beam starts, the sample quality changes to GOOD and the profile controller resumes its work.

Another signal property is the availability of samples. Algorithms may be active only during certain time windows. Outside these windows, no samples are produced. A similar case occurs in case of interruptions of the communication network. To inform control system components of such changes a production state with values RUNNING, OUTDATED, TIMEOUT or STOPPED is attached to output signal samples together with the confidence state.

3.2 Extended Exception Handling

Extended exception handling consists of a number of higher-level functions with a larger scope of the built-in system model extending coverage on plasma events. Advanced reconstructors use input signals from different diagnostic sources. Problems in one diagnostic system thus can be mitigated by replacement strategies, bayesian filtering etc. Advanced command conditioners can use alternate actuating systems (e.g. ICRH instead of NBI for central heating). Monitors implement complex algorithms to detect events using information from a variety of sources. The result is summarized in an event state vector signal.

The larger scope of extended exception handling also justifies the ability to take system-wide decisions by triggering soft-landing or pulse stop actions via the DCS alarm system. In order to trigger an alarm the function attaches a special tag to a descriptive log message. The alarm system collects all alarms raised in a control cycle, filters the highest rank and forwards it to pulse supervision, while the textual part is forwarded to the log messaging system for visualization. The alarm system is open to all control functions. By using the alarm system the function designer qualifies the function to be capable of extended exception handling.

4. Central Exception Handling

Especially in plasma operation there are cases like impurity accumulation or MHD mode activity, where the individual measurements or reconstructions are correct but plasma condition nevertheless experiences an undesired change. In other cases, local repair actions fix

Wolfgang.Treutterer@ipp.mpg.de

visible symptoms but the underlying event still exists and continues disturbing the plasma. In the medical analogy this would be comparable with fighting fever but not curing the inducing disease. Then, central exception handling is necessary to conduct an integrated response coordinating the reaction across the multitude of involved control functions. This coordination can comprise turning specific functions on and off, changing the settings of functions and changing reference values for feedback controllers and command output functions.

While the essence of local exception handling is solving problems in order to sustain the nominal functionality, central exception handling can also be applied to achieve more versatile targets. This central instance adapts references to comply better with the actual boundary conditions. Highest priority has, of course, device protection. But in less severe cases waveforms can be tweaked or an alternative goal and the associated reference waveforms can be selected by a segment change to save precious experiment time and costs especially during long pulse operation. Finally the same methods as for problem solving can be used to implement optimization procedures. Reference adaption is implemented in DCS by two methods: direct modification of individual reference waveforms and indirect modification changing the active segment.

4.1 Reference Computation

The DCS reference generator has the capability to add, move and remove data-points of reference waveforms. This feature can be used to change waveform shapes according to a measured state. DCS reference generation occurs in three steps. In the normal case, the pulse designer defines reference waveforms explicitly in the schedule. During the main step the waveforms are interpolated. In addition, dedicated pre- and post-processor algorithms can be attached to each segment such that the waveforms and the interpolated value can be manipulated according to specified rules.

Soft landing in AUG is a multi-staged procedure for a controlled ramp down of the plasma current from its actual value to zero and is realized by direct reference waveform modification. In the first stage, plasma shape is steadily transformed to a limited configuration, next PF currents are ramped to zero, preserving the shape. After the plasma current has ceased, the residual currents are reduced at maximum ramp speed. In parallel, external heating is reduced but only to a degree that keeps divertor pumping effective. The references of control mode signals, plasma current, position and shape references and the feed forward commands for PF an CS coils, as well as for the heating sources are repeatedly computed by a segment preprocessor dependent on the measured state and governed by a finite state machine.

Reference computation needs not be limited to soft landing. It can be used as a general-purpose tool to tailor the pulse schedule to the plasma state. Another application could e.g. adjust the plasma current reference immediately after breakdown to match the actual situation in order to avoid a current overshoot and the accompanying current profile deformation.

4.2 Segment Scheduling

If conditions prohibit reaching the target of the current segment a branch to another segment can switch to an alternative goal or initiate a termination procedure. Segment branching is similar to a GOTO statement in programming.

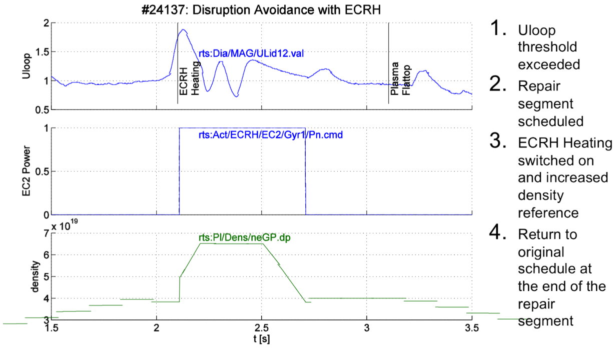

Fig. 3: Disruption avoidance using an Insert Segment

If possible, it is preferable to improve a degraded system state with an intermediate repair procedure and subsequently continue the actual segment and its investigation target. Such procedure is accomplished by an "insert segment" and is comparable to a function CALL statement. Insert segments have been successfully used for disruption avoidance. Fig. 3 shows time-traces from pulse #24137 where a sudden increase of the loop voltage signalizes a pending disruption due to an MHD mode. As a response, central exception handling scheduled a repair segment with additional ECRH heating and an accompanying density pulse. All other references are frozen at the last values of the original segment. Upon termination of the repair segment, control is returned to the interrupted original segment.

Quite often more than one exception requires handling at the same time. This requires an arbitration strategy. For this purpose DCS employs a prioritization scheme based on the sequence of condition statements in the segment description. The exception with the topmost position takes precedence. After a segment change the conditions and the prioritization of the newly scheduled segment applies, thereby implementing a flexible and configurable state machine.

5. Conclusion

ASDEX Upgrade has benefitted a lot from the built-in exception handling strategies. Even at the lowest level basic exception handling and the prescribed use of quality tags has educated function designers to consider possible problems like invalid input signals and algorithm limitations. This is the fundament for robust and stable operation.

Extended exception handling with its limited handling methods soft-landing or pulse stop might seem too simplistic for a complex device like ITER and only suitable for a short pulse machine as ASDEX Upgrade. Many issues solved by extended exception handling in ASDEX Upgrade will probably migrate to the domain of central exception handling in ITER. But this method bears considerable potential for automatized responses in a future fusion power plant, which is probably no longer based on pulse schedules but on built-in procedures.

Central exception handling, finally, is a flexible and extremely powerful tool, usable also for pulse optimization. Up to now the additional complexity considering all consequences of segment branching or segment insertion has prevented pulse designers from massive application of this technique. In a long-pulse machine like ITER such constraints appear to be loosing weight and are ameliorated by the development of pre-pulse simulation and validation tools, which will also include exception handling.

Wolfgang.Treutterer@ipp.mpg.de

References

[1] G. Raupp et al., “Real-time exception handling—Use cases and response requirements”, http://dx.doi.org/10.1016/j.fusengdes.2012.06.002

[2] Ph. Moreau et al., “Failure mode and recovery strategies for the operation of the Tore Supra Tokamak”, Proceeding of ICALEPCS, Kobe (2009), pp. 591–593

[3] R. Nouailletas et al., “A generic method for real time detection of magnetic sensor failure on tokamaks”, Fusion Engineering and Design, vol. 87, no. 3, pp. 289 – 297, 2012.

[4] A. Stephen, et al., “Centralised coordinated control to protect the JET ITER-like wall”, 13th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS’11), 2011

[5] G. Raupp et al., "Protection Strategy in the ASDEX Upgrade Control System", Fusion Technology (Proc. of the 18th Symposium on Fusion Technology, Karlsruhe, 1994) , (Ed.) Herschbach, K. and Maurer, W. and Vetter, J. E. (Elsevier, Amsterdam), Vol. 1 (1995), 679-682

[6] W. Treutterer , et al., "The New ASDEX Upgrade Real-Time Control and Data Acquisition System", Fusion Engineering and Design 66-68, 1-4 (2003), 755-760, doi:10.1016/S0920-3796(03)00298-9

[7] G. Raupp , et al., "ASDEX Upgrade CODAC overview", Fusion Engineering and Design 84, 7-11 (2009), 1575-1579, doi:10.1016/j.fusengdes.2009.01.031, 25th SOFT Conference, 15.09.2008-19.09.2008, Rostock, Germany

[8] W. Treutterer, T. Zehetbauer, V. Mertens, G. Neu, G. Raupp, D. Zasche, ASDEX Upgrade team, Plasma feedback controller reorganisation for ASDEX Upgrade's new discharge control and data acquisition system, Fusion Engineering and Design, Volume 74, Issues 1–4, November 2005, Pages 501-505, ISSN 0920-3796, 10.1016/j.fusengdes.2005.06.151.

Wolfgang.Treutterer@ipp.mpg.de