Fusion Engineering and Design 85 (2010) 394–398

Contents lists available at ScienceDirect

Fusion Engineering and Design

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

Scientific component framework for W7-X using service oriented GRID middleware

A. Werner[a,*], J. Svensson[b,1], G. Kühner[a], T. Bluhm[a], P. Heimann[c], S. Jakob[d], C. Hennig[a], H. Kroiss[c], H. Laqua[a], M. Lewerentz[a], H. Riemann[a], J. Schacht[a], A. Spring[a], M. Zilker[c], J. Maier[c]

[a] Max-Planck-Institute for Plasmaphysics, Teilinstitut Greifswald, Wendelsteinstr. 1, 17491 Greifswald, Germany
[b] Euratom-UKAEA, Culham Science Centre, Abingdon, Oxon OX14 3DB, United Kingdom
[c] Max-Planck-Institute for Plasmaphysics, Boltzmannstr. 2, 85748 Garching, Germany
[d] FACHHOCHSCHULE STRALSUND, Zur Schwedenschanze 15, 18435 Stralsund, Germany

ARTICLE INFO

Article history:
Available online 16 December 2009

Keywords:
Cloud computing
Grid computing
SOA
Wendelstein 7-X
Modelling
Data analysis
Plasma control

ABSTRACT

Future fusion experiments, aiming to demonstrate steady state reactor operation, require physics driven plasma control based on increasingly complex plasma models. A precondition for establishing such control systems is widely automated data analysis, which can provide integration of multiple diagnostic on a large scale. Even high quality online data evaluation, which is essential for the scientific documentation of the experiment, has to be performed automatically due to the huge data sets being recorded in long discharge runs. An automated system that can handle these requirements will have to be built on reusable software components that can be maintained by the domain experts: diagnosticians, theorists, engineers and others. For Wendelstein 7-X a service oriented architecture seems to be appropriate, in which software components can be exposed as services with well defined interface contracts. Although grid computing has up to now been mainly used for remote job execution, a more promising service oriented middleware has emerged from the recent grid specification, the open grid service architecture (OGSA). It is based on stateful web services defined by the web service resource framework (WSRF) standard. In particular, the statefulness of services allows to setup complex models without unnecessary performance losses by frequent transmission of large and complex data sets. At present, the usability of this technology in the W7-X CoDaC context is under evaluation by first service implementations.

1. Introduction

A major challenge for fusion experiments is the demonstration of reactor relevant plasmas. The main aspects are to achieve high plasma performance for reaching a burning plasma state and to establish and hold this state for long durations or even for steady state, i.e. much longer than the longest equilibration times. In such long pulse discharges or steady state operations, the plasma state depends on the boundary conditions rather than on initial conditions, i.e. pre-programmed reference control parameter will not be sufficient to reach a certain plasma state like e.g. high density high confinement modes [1] or ITER advanced steady state scenarios [2]. Those plasma scenarios can be stabilized only by a physics based and sufficiently intelligent plasma control.

Although stellarator plasmas are stable on a short time scales, i.e. no disruption mitigation, plasma position control, resistive wall mode suppression and others are required for operation, a stabilization on longer time scales is nevertheless essential. In particular for Wendelstein 7-X, major long term stabilization loops have been identified for magnetic edge configuration determined by long induction (L/R) and skin times (approximately 60 s) and thermal first wall loads determined by thermal relaxations (1–100 s) and accompanying recycling and out gassing processes [3]. Of course, such long time constants pose a general problem for plasma controllers and those stabilization challenges can be tackled by a plasma physics driven control only, which includes also plasma modeling of predictive parts for feed forward or adaptive controllers.

Tokamaks are well developed with plasma controllers on short MHD time scales, however, the challenges mentioned above will occur for long pulse operation, regardless of the magnetic confinement concept. Preparative work on plasma physics control has been already carried out by Tore Supra for limitation of first wall power load while sustaining the desired plasma performance [4]. JET has demonstrated the control of the q profile [5] and ASDEX Upgrade as well as TEXTOR develops the suppression of neoclassical tearing modes [6].

* Corresponding author. Tel.: +49 3834882520.
E-mail address: andreas.werner@ipp.mpg.de (A. Werner).
[1] Present address.

0920-3796/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.fusengdes.2009.10.010

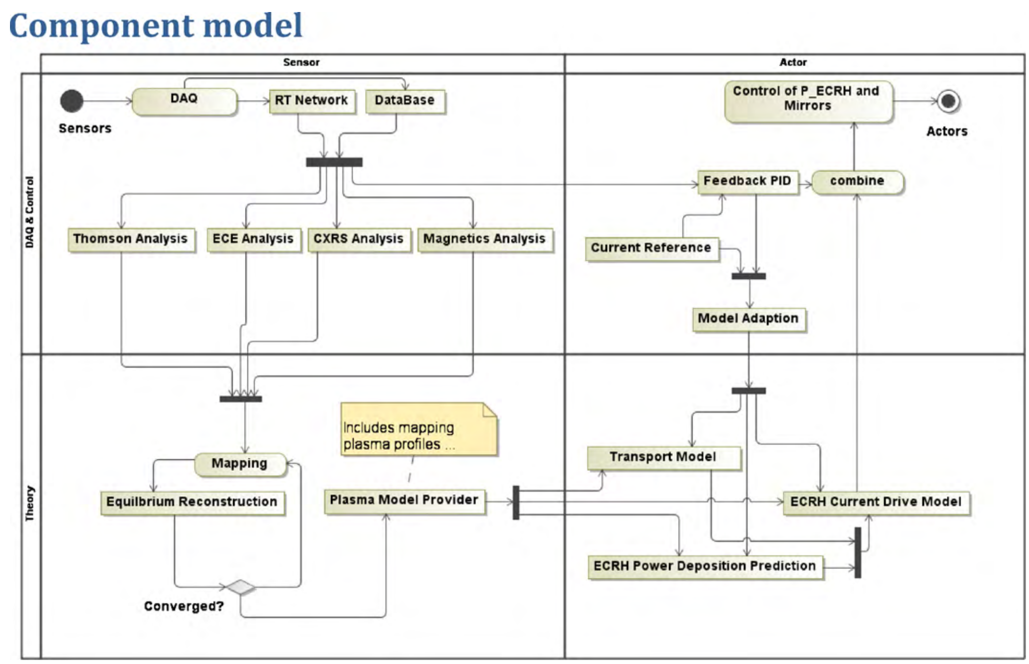

Fig. 1. UML activity diagram of a possible control loop for current stabilization by ECCD.

The development of such complicated control scenarios requires a framework, which allows for the arbitrary combination of data analyses, plasma modeling and real time plasma control. We propose a component architecture, in which software modules with dedicated tasks are distributed on dedicated computing nodes. For Wendelstein 7-X, a promising approach to the framework implementation supporting distributed components is given by the service oriented architecture (SOA).

An advanced control scenario for Wendelstein 7-X is composed of several processes being represented by software. Fig. 1 depicts a potential solution to the current stabilization challenge of Wendelstein 7-X required for a stabilized magnetic edge island configuration in divertor operations. It comprises activities of several aspects, like the more experimental aspect in terms of data acquisition and control versus the theory model aspect as well as the sensor domain versus the actor domain. These activity diagrams support the identification of soft- and hardware components and their reusability for other activities. In the following, components describe a piece of software, which is written in any programming language, running on an appropriate platform and executed on one or more computing nodes. The diagram in Fig. 1 depicts a classical solution to this control problem in a sense that analysis and models are arranged in chains. A proper application of such a chain for control purposes poses further non-functional requirements on the component and the real time system. One is in general a quality of service requirement, that is to guarantee response times, e.g. by multi-tier solutions. And furthermore, a drop out of eventually one response sample should be smoothly interpolated by the real time systems. Beside graphs as presented in Fig. 1, Bayesian graph nodes can be also represented by components as recent developments of integrated data analyses using Bayesian graphical models [7,8] demonstrate.

For future reactor relevant plasma development, the flexible arrangement of such components is likely to become essential, since plasma control under inclusion of theory models will be a necessary and continuing development process. Within this process, it is likely that theory models and analysis processes will be replaced by new ones and different approximative models are being used for different experiment scenarios. Thus, the W7-X CoDaC group wants to provide a framework, in which the plasma physics for control and data analysis can be easily developed in terms of software components combination and replacement. A suitable framework should fulfill several requirements. (R1) Components must be easily pluggable to other components. (R2) The component provides a machine readable self-description and client functions and data structures can be generated automatically. (R3) Components should be developed in any language and support interoperability between different platforms. (R4) Components should be discoverable and monitored. (R5) Components should be developed and maintained by the corresponding domain experts. Further requirements are the possibility for parallelization of distributed memory applications (R6) and the decoupling of development cycles of different components (R7). Finally, a widely used standard with respect to communication protocols and component creation should be employed in order to gain from developments of a large community (R8).

In the envisaged SOA approach, the components, or more precise, their interfaces are exposed as services. These services represent functions, which can be composed to a high level process chain, the so called orchestration of services. Many different technological service implementations exist. However, for the combination of services in a heterogeneous computing environment SOAP/XML web services according to standards defined by the web consortium [9] are a good choice with respect to the given requirements R1–3. Such services communicate mainly via network connections and, thus, it is essential to design the components with respect to their communications and the accompanying quality of service requirements. Atomic transactions should be prevented and a good number for the function granularity assessment is the ratio of real computation time to the sum of the required times for network communication and XML parsing. In local area networks network roundtrip times are typically below 0.5 ms and more than 1000 service function calls can be easily achieved, since XML parsing for typical SOAP message sizes is even faster on modern computers.

Working with services provides additional advantages. Web services expose data structures and functions, which is almost opposite to the paradigm of transferring bulky data sets and process