GA–A26729

OPERATION REQUEST GATEKEEPER – A SOFTWARE SYSTEM FOR REMOTE ACCESS CONTROL OF DIAGNOSTIC INSTRUMENTS IN FUSION EXPERIMENTS

by
G. ABLA, T.W. FREDIAN, D.P. SCHISSEL, J.A. STILLERMAN,
M.J. GREENWALD, D. STEPANOV and D.J. CIARLETTE

JUNE 2010

GENERAL ATOMICS

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

GA–A26729

OPERATION REQUEST GATEKEEPER – A SOFTWARE SYSTEM FOR REMOTE ACCESS CONTROL OF DIAGNOSTIC INSTRUMENTS IN FUSION EXPERIMENTS

by
G. ABLA, T.W. FREDIAN,* D.P. SCHISSEL, J.A. STILLERMAN,*
M.J. GREENWALD,* D. STEPANOV[†] and D.J. CIARLETTE[‡]

This is a preprint of a paper to be presented at the 18th Topical Conference on High Temperature Plasma Diagnostics, May 16–20, 2010 in Wildwood, New Jersey and to be published in Review of Scientific Instruments.

*PSFC, Massachusetts Institute of Technology, Cambridge, Massachusetts USA
[†]ITER Organization, St Paul Lez Durance Cedex, France
[‡]Oak Ridge National Laboratory, Oak Ridge, Tennessee USA

Work supported in part by
the U.S. Department of Energy
under DE-FC02-04ER54698, DE-FC02-99ER54512
and DE-AC05-00OR22725

GENERAL ATOMICS ATOMICS PROJECT 30200
JUNE 2010

ABSTRACT

Tokamak diagnostic settings are repeatedly modified to meet the changing needs of each experiment. Enabling remote diagnostic control has significant challenges due to security and efficiency requirements. The Operation Request Gatekeeper (ORG) is a software system that addresses the challenges of remotely but securely submitting modification requests. The ORG provides a framework for screening all the requests before they enter the secure machine zone and are executed, by performing user authentication and authorization, grammar validation, and validity checks. A prototype ORG was developed for the ITER CODAC that satisfies their initial requirements for remote request submission and has been tested with remote control of the KSTAR Plasma Control System. The paper describes the ORG's software design principles and implementation as well as worldwide test results.

I. INTRODUCTION

Tokamaks are equipped with many diagnostics and control devices. For example, DIII-D National Fusion Facility located at San Diego, California has more than 50 diagnostics devices. As experimental magnetic fusion science has progressed, the number and complexity of the diagnostic devices on the tokamaks has continued to increase.

The US currently has three major experimental facilities with more than a thousand researchers from a variety of locations participating in experimental research. ITER, the next generation magnetic confinement experiment, will bring thousands of fusion researchers from China, Europe, India, Japan, Korea, Russia, and the U.S together in a common research program.

It is also expected that ITER will house many diagnostic instruments. While related, they will be independent and access methods to them will be vary. The operation of ITER experiments, similar to the operations of current tokamak devices, requires making constant changes to the configurations of those instruments during scientific experiments. Due to the size of the team and distance from ITER for many participating countries, it is not practical to expect all participants to always be located on site. Therefore, a capability that allows remote access and control of diagnostic instruments has significant value for ITER.

Remotely accessing and controlling an instrument is not a new field. For example, Mars rovers, and telescopes are two of the successful areas in the remotely accessing and controlling remote devices. However, in those areas the communication with the remote devices is done via a dedicated communication channel. The remote access described in this paper is via Internet, which requires significant security measures due to the openness of Internet. Furthermore, a variety of new diagnostic instruments are expected to be installed at ITER in its lifetime. Therefore, the remote access software tools and validation methods for ITER needs to be versatile enough to accommodate multiple diagnostic instruments, including the ones to be installed in the future.

Remote access to the diagnostic instruments is expected to be very secure. Security should be present throughout different aspects of the remote access process. First, any system that enables remote access to the experiment should address authentication. Then, the authorization permission to access a specific instrument need to be validated against each authenticated user. The content of each access request also needs to go through rigorous physics validation before it reaches the actual instrument.

The remote access system also needs to be very modular. It needs to provide a mechanism to communicate with the correct verification system of any existing or future instrument that will be placed on the experiment. With this mechanism, groups that are supplying diagnostic and control systems to future tokamaks will have a guide for their own verification modules that can be tested before placing it into operation. It is assumed that the verification and validation are for each independent diagnostics system. If two diagnostic instruments are tightly coupled, and there is no method to intercept the communication or validation method between them, the verification and validation process should then be done internally.

The Operation Request Gatekeeper (ORG) is a prototype solution to address the challenges of remotely accessing diagnostic instruments. It is proposed as the only channel for interaction for incoming access requests to the ITER experimental site. The prototype design includes modules that perform the identification of users and validation of their access privileges. It also provides a framework for attaching codes that perform request content verification and execution.

The following sections describe the software design principles of the ORG and its implementation details. The last section also reports on the ORG prototype deployment in the remote support of KSTAR operations as well as test results.

II. DESIGN PRINCIPLES AND SOFTWARE ARCHITECTURE

The ORG is the only proposed channel for all actions, defined as requests, requiring information entry into the Plant Operation Zone (POZ) of ITER. The term request is meant to imply anything from a simple command (e.g. change a scalar value on a diagnostic instrument) to a complex serious of instructions (e.g. a new pulse schedule file) to perhaps installing or updating a software package (e.g. a request verification module). It is the ORG's job to screen the authenticity and validity of requests before they are applied to diagnostic and control instruments. Based on the screening results, the ORG decides whether or not to execute the requests. The screening is a pipeline that performs the following specific tasks: (1) Authentication of the remote request submitter, (2) Verification of the access control permission of the requester, (3) Validation of the request format and appropriateness of the request content, and (4) pass the request to POZ, where detailed validation and execution of the request take place.

Carrying out the tasks above relies on appropriate database designs. The authentication step requires a user authentications database, which can rely on the user account system and site security implementation of the specific tokamak experimental site. The verification of the access control permissions needs to access a database, which stores information about the authorized actions for each user. The information about accessible diagnostic instrument systems as well as the history of actions taken by the requests also needs to be stored.

The ORG design has two distinct components, although the design does not preclude their deployment on a single physical computer. The first component is the Receiving Component (RC), which is responsible for tasks 1 to 3. The other component is the POZ Component (POZC) that is responsible for task 4. The design also adheres to the principle where no component writes into the other component's database. This is done for tighter security; remote reading is allowed but writing is not. Therefore, the RC can read information from the POZC database but not write. A request that the RC deems worthy of passing on is queued up for transfer and the POZC is notified of the queued request. It is up to the request processor in the POZC to reach out and read the requested information from the RC.

Figure 1 shows the details of the ORG system. As stated in the previous section, the first RC action is checking authentication and authorization. After that is accomplished, the RC checks that the request is properly formatted and has the proper grammar. As an example, if there is an accompanying file with the request, the file type and format are checked. Or, if the request is in the form of XML it is checked for proper syntax. To be able to make this check, the RC reads out of the POZ from the Configuration Database what is the appropriate syntax (e.g. XML schema). Once this check is satisfied, the RC will perform an initial simple check on allowable values (range validation). Simple values and commands must be of proper type and format for entry into the request database. The RC determines the allowable values by reading from the Configuration Database.

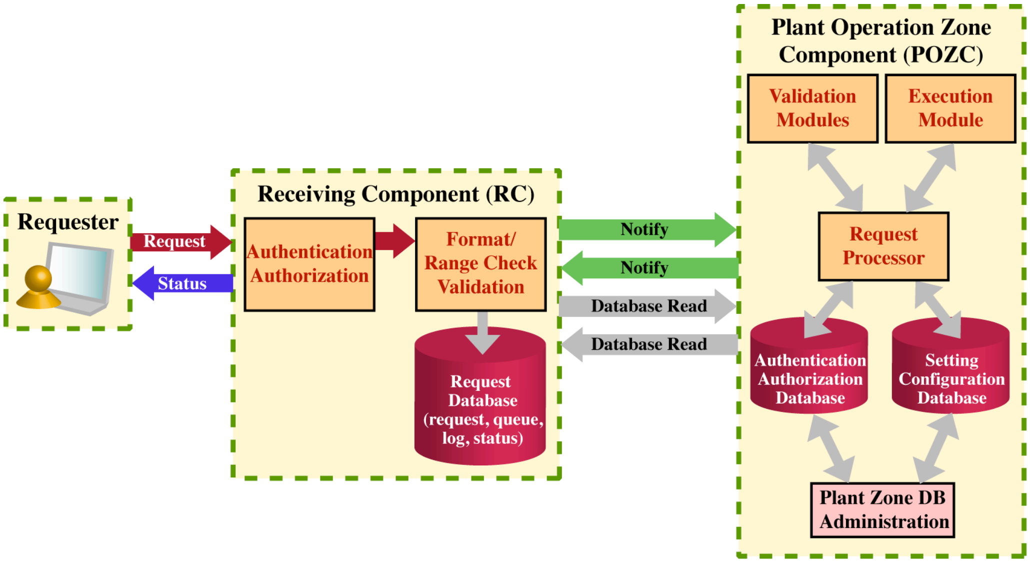
Plant Operation Zone Component (POZC)
Validation Modules
Execution Module
Receiving Component (RC)
Requester
Request
Status
Authentication Authorization
Format/ Range Check Validation
Notify
Notify
Database Read
Database Read
Request Processor
Request Database (request, queue, log, status)
Authentication Authorization Database
Setting Configuration Database
Plant Zone DB Administration

FIG. 1. A detailed diagram showing the interaction of the RC with the POZC. For security reasons, all database interactions between the RC and the POZC are read only. The administration of the databases inside the POZC is only allowed from inside the POZC.

If both of these checks pass, the RC writes into a local request database and sends a notification to the Request Processor that is inside the POZC. The Request Processor at an appropriate time reads out of the request database all the required information, combines this with information from the configuration database, and then sends the request to the appropriate verification module(s). If the request fails to pass the verification process, then Request Processor rejects to forward the request to execution module(s).

If the request passes the verification process, then Request Processor sends the request to the appropriate execution modules(s), in which the request is acted upon. Both the verification and execution modules are written by the most appropriate organization, those with the most detailed domain-specific knowledge. The ORG design also specifies the verification and execution module interfaces.

The status of each submitted request at each stage is available to the requester. All steps that a request traverses, namely authentication/authorization, grammar check, range validation, verification and execution, whether successful or not, are logged and can be obtained by the requester or by another authorized user. The aim is to provide users with sufficient information to correct errors in their requests without exposing unnecessary internal information. The remote users and applications query the current and historical status of the request in order to monitor the remote access and control activities. The format of the status is general enough so that different applications can use the information for further processing or for customized display.

III. ORG PROTOTYPE DEVELOPMENT

The prototype ORG system was developed as a web-based application. The two components of the ORG, the RC and POZC run on web servers. Clients interact with RC to submit request and require status via HTTP GET/POST messages. The clients can be a desktop GUI applications or interactive web-pages.

Development utilized the Python[1] programming language, and the Django framework.[2] Django is a Python-based framework for web development. Its Model-Template-View (MTV) architecture follows the popular Model-View-Controller (MVC) software design pattern, which provides clean separation of tasks and responsibilities among the critical aspects of an application. Along with its architecture, Django provides many capabilities that benefit web application development work. One such capability is its Object-Relation-Mapper capability, which can provide an easy access to an underlying database. In Django, the interface details of an underlying storage mechanism or specific database software are hidden. Thus, the database and its data become a python object. Another beneficial capability of Django is the Template engine that is responsible for rendering display information via the web server. Taken all together, application developers can concentrate on the functional logic of the software without worrying about how the end data is displayed.

There are three important aspects of the ORG software system: (1) database models; (2) the logical process engine that perform the authentication/authorization, grammar validation, range check, detailed verification and execution; and (3) the web page and web service generator. These three aspects of the development were very well represented with Django's MTV architecture. The database tables were created with Django Models. All the logical components of the ORG were implemented in the Django's View model with Python. The different web page styles are being designed with Django's Template Language and stored in the template repository. Django is responsible for rendering the HTML pages (or XML data). Django's built-in user management capability was used for ORG user authentication. Finally, the interaction between python codes and web server was being handled by the modwsgi/modpython interface. The open source MySQL[3] was used as the database server and Apache[4] was used as the web server.

Two types of ORG clients were developed. The first type of client is an interactive web application. It provides an environment that enable users to login, build a request, submit a request and monitor validation as well as execution process. Another type of client is a group of command line-based executable scripts that provide all the necessary functionality for submitting requests and monitoring their status. Both types of clients utilize HTTP GET and HTTP POST methods to carry out their functions. While the web-based client provides a highly interactive and easy to use interface, the command line based scripts provide a basis for developing automated request submission capabilities or integrating ORG client capabilities into existing desktop GUI applications.

IV. INITIAL DEPLOYMENT AND TESTS

The prototype ORG system was deployed and tested in both lab environment and real tokamak remote operations. Figure 2 shows one of the typical test deployments.

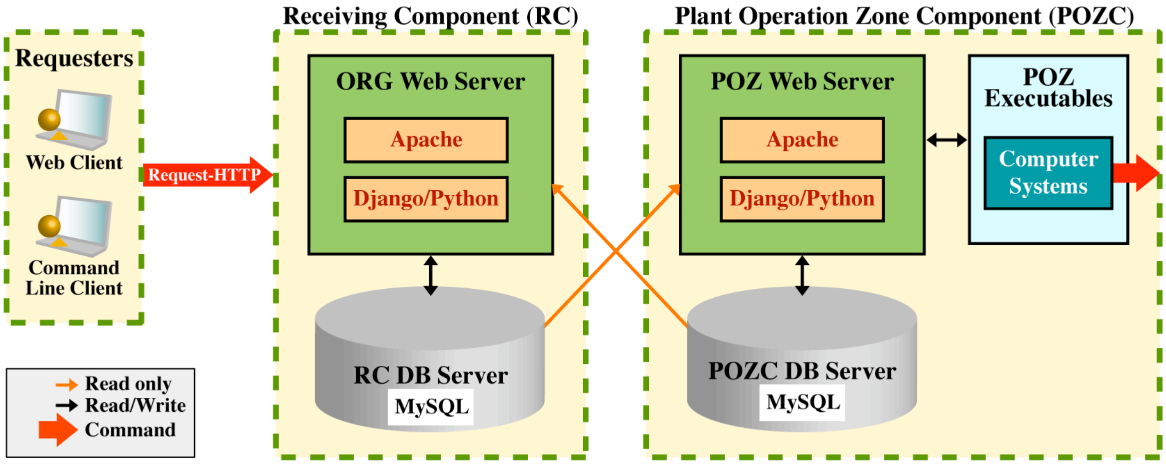

FIG. 2. A diagram showing a typical deployment of the ORG.

The first test was a simulated request submission environment to ITER. In this environment, all of the ORG capabilities were tested against a group of simulated diagnostic and control hardware, such as RF Power system and TF coil system. The tested capabilities were:

1. The ORG administration capabilities
   a. User account creation, users authentication, and role authorization.
   b. Diagnostic and Control system description on ORG and configuration of the allowable parameter/setting changes.
   c. Creation of corresponding verification and execution scripts for the described Diagnostic and Control systems.
2. The ORG client capabilities
   a. Request creation, request submission, request verification, and monitoring.

Tests showed that the ORG is capable of describing the parameter settings of multiple software and hardware systems. The tests also proved that ORG can handle a variety of request submission scenarios, such as simply changing a scalar or Boolean value to uploading a complex file and executing its contents.

The prototype ORG was also deployed and tested in real fusion experiment environment. In this test the ORG was used for remotely controlling the Plasma Control System (PCS)[5] deployed at KSTAR, South Korea from General Atomics (DIII-D) in San Diego, CA USA.

As the preparation for the tests, both RC and POZC components of the ORG software were deployed on a server at the KSTAR site. The PCS pulse file submission was described in the ORG configurations so that it accepts a pulse file, performs verification and forwards it to the PCS server of the KSTAR tokamak. Normally, the PCS pulse files are prepared on a PCS client. The PCS client is an interactive GUI application that runs locally at the experimental site. In this test, the PCS GUI client was slightly modified to include the ORG client functionality so that pulse files can be prepared at the DIII-D site locally and uploaded to the KSTAR ORG, which is responsible for submitting the file to the KSTAR PCS server if the submission passes all the screening and verification processes of the ORG.

During the test, multiple PCS pulse files were created at the DIII-D site, uploaded to the KSTAR ORG, and successfully executed by the KSTAR PCS. The ORG provided tools to monitor the request verification and execution processes ORG user authentication and authorization management capabilities were also successfully tested.

V. CONCLUDING DISCUSSIONS AND FUTURE WORK

The design and prototype ORG satisfies the initial ITER CODAC requirements for remote request submission to ITER. The ORG tests were successful in remote access control of diagnostic instruments in a simulated ITER environment. The ORG test of remote PCS operation of KSTAR was also successful. Both test environments demonstrated how remote control of instruments on a fusion tokamak can be accomplished in a secure and safe way. The secure aspects of the ORG involve the ability to send communication over the Internet and to guarantee the authenticity of the sender, the authorized permissions of the sender, and the integrity of the request. The safety aspects employed are automatic validation algorithms that add layers of safety beyond human input.

There are several areas the ORG can be extended. One such area is the request verification process. In the current prototype implementation, "human verification" of the requests is not considered, although the ORG framework does not reject such an extension. Therefore, the scenario – "user submits a request but execution must be approved by an operator", currently is not native supported. This may be required in the actual tokamak operation. Another area is to improve the ORG's adaptability to highly interactive request submission and monitoring applications. Such improvement was planned and will be developed by tightly integrating ORG with PCS deployments in future remote operations.

REFERENCES

[1] Python Programming Language — Official Website, http://www.python.org

[2] Django web site, http://www.djangoproject.com/

[3] MySQL web site, http://www.mysql.com

[4] Apache HTTP Server Project, http://httpd.apache.org/

[5] B. G. Penaflor, J. R. Ferron, and M. L. Walker, Proc. 19th Symposium on Fusion Technology, Lisbon, Portugal, 1996 (North-Holland Publishing Company, 1997) vol. 1, p. 965.

ACKNOWLEDGMENT

The design and prototype development of the ORG was funded by the ITER organization. The ORG tests in the KSTAR production environment was supported partly by US Department of Energy and KSTAR. The authors thank the PCS groups of DIII-D National Fusion Facility and KSTAR experiments, as well as US ITER office and ITER CODAC team for their support of this project. This work was partially supported by the U.S. Department of Energy under DE-FC02-04ER54698, DE-FC02-99ER54512 and DE-AC05-00OR22725.