ITER CONTRIBUTION TO CONTROL SYSTEM STUDIO (CSS) DEVELOPMENT EFFORT

N. Utzel, L. Abadie, F. Di Maio, J.Y. Journeaux, A. Wallander, I. Yonekawa,
ITER Organization, Route de Vinon sur Verdon, 13115 Saint Paul Lez Durance, France
F. Arnaud, G. Darcourt, D. Dequidt, Sopra Group, 13791, Aix-en-Provence cedex 3, France

Abstract

In 2010, Control System Studio (CSS) [1] was chosen for CODAC - the central control system of ITER [2] - as the development and runtime integrated environment for local control systems. It became quickly necessary to contribute to CSS development effort - after all, CODAC team wants to be sure that the tools that are being used by the seven ITER members all over the world continue to be available and to be improved. In order to integrate CSS main components in its framework [3], CODAC team needed first to adapt them to its standard platform based on Linux 64-bits and PostgreSQL database. Then, user feedback started to emerge as well as the need for an industrial symbol library to represent pump, valve or electrical breaker states on the operator interface and the requirement to automatically send an email when a new alarm is raised. It also soon became important for CODAC team to be able to publish its contributions quickly and to adapt its own infrastructure for that. This paper describes ITER increasing contribution to the CSS development effort and the future plans to address factory and site acceptance tests of the local control systems.

INTRODUCTION

The ITER project aims to demonstrate the feasibility of commercial production of fusion energy. It is an international project that involves seven members (China, Europe, India, Japan, Korea, Russia and USA) who provide all plant systems (magnet, vacuum vessel, divertor, cryostat, diagnostics…) in-kind through so called procurement arrangements. The majority of them include local control systems that need to be integrated into CODAC – the central control system of ITER.

To mitigate the risks during integration, a major effort has been invested to provide not only guidelines and standards applicable to all local control systems but also a framework that implements these standards and guarantees that the local control systems can be integrated into the central one.

This framework is based on EPICS [4] – Experimental Physics and Industrial Control System – which is a client/server architecture and a set of tools for building scalable control systems with:

- Distributed real-time Process Variable (PV) database,
- Software Bus – Channel Access – that allows the clients (requestors) to do operations such as Search, Get, Put or Add Event (add monitor) and the servers (providers) to only send data to the client when it has changed for instance in the case of a monitor request.

CODAC control system framework also includes some EPICS extensions. Control System Studio is one of its key components providing common services such as the operator interface, the alarm system, engineering archival and the electronic logbook necessary to monitor and operate both local and central control systems.

SCOPE AND OBJECTIVES

Control System Studio is an Eclipse-based collection of tools to monitor and operate large scale control systems, such as those in the accelerator and fusion community. It is the result of a collaboration amongst many laboratories and universities.

This collection of tools consists of more than 350 core and application plugins and nearly half of them are integrated into the CODAC control system framework to provide common services:

- **Operator Interface** (OPI) that connects to the local control system, animates graphical widgets according to an EPICS PV value, alarm status/severity and connection/read-write status, shows PV's range and alarm limits and allows the operator to interact with the process by providing input data and sending commands,
- **Alarm System** that monitors alarms in the control system and provides essential support to the operator by warning him of situations that need his attention, showing guidance, allowing him to open dedicated displays, execute commands and acknowledge alarms,
- **Engineering Archival** that monitors and archives EPICS PV values on a dedicated storage and provides a graphical user interface for displaying live and historic data in a plot, making some computations, adding annotations and exporting values into different file formats such as Excel spread sheet or Matlab,
- **Electronic Logbook** that registers events which have been manually or automatically generated during operation, to keep track of problems, human decisions or actions which were taken during the course of the activity and which may have had an impact on the outcome of the activity.

The main objectives of the integration of CSS in the CODAC framework were firstly technical, to evaluate, select and quickly adapt the tools to their environment. Secondly it was important to adopt a strategy of continuous improvement of these common services by joining the Control System Studio Collaboration.

ISBN 978-3-95450-139-7

A DISTRIBUTED ARCHITECTURE

CODAC services common to all ITER plant systems, including the operator interface, the alarm system, the engineering archival and the electronic logbook, are distributed over many CODAC servers and terminals which are installed in the ITER control room and CODAC server room.

These services interface with approximately 220 local control systems, each of which includes a set of tightly coupled slow and/or fast controllers, with one and only one Plant System Host, which implements plant-specific and generic functions using EPICS PVs.

In order to facilitate integration, the local control systems are grouped into roughly twenty control groups: buildings, water cooling, cryogenic, fuelling, diagnostics, etc. To reflect the functional breakdown, CODAC services are distributed over the control groups with one alarm server and one archive engine per control group.

Finally, CODAC services are present on the "Central supervision, monitoring and data handling" layer with the alarm and archive central databases, the electronic logbook and the operator terminals.

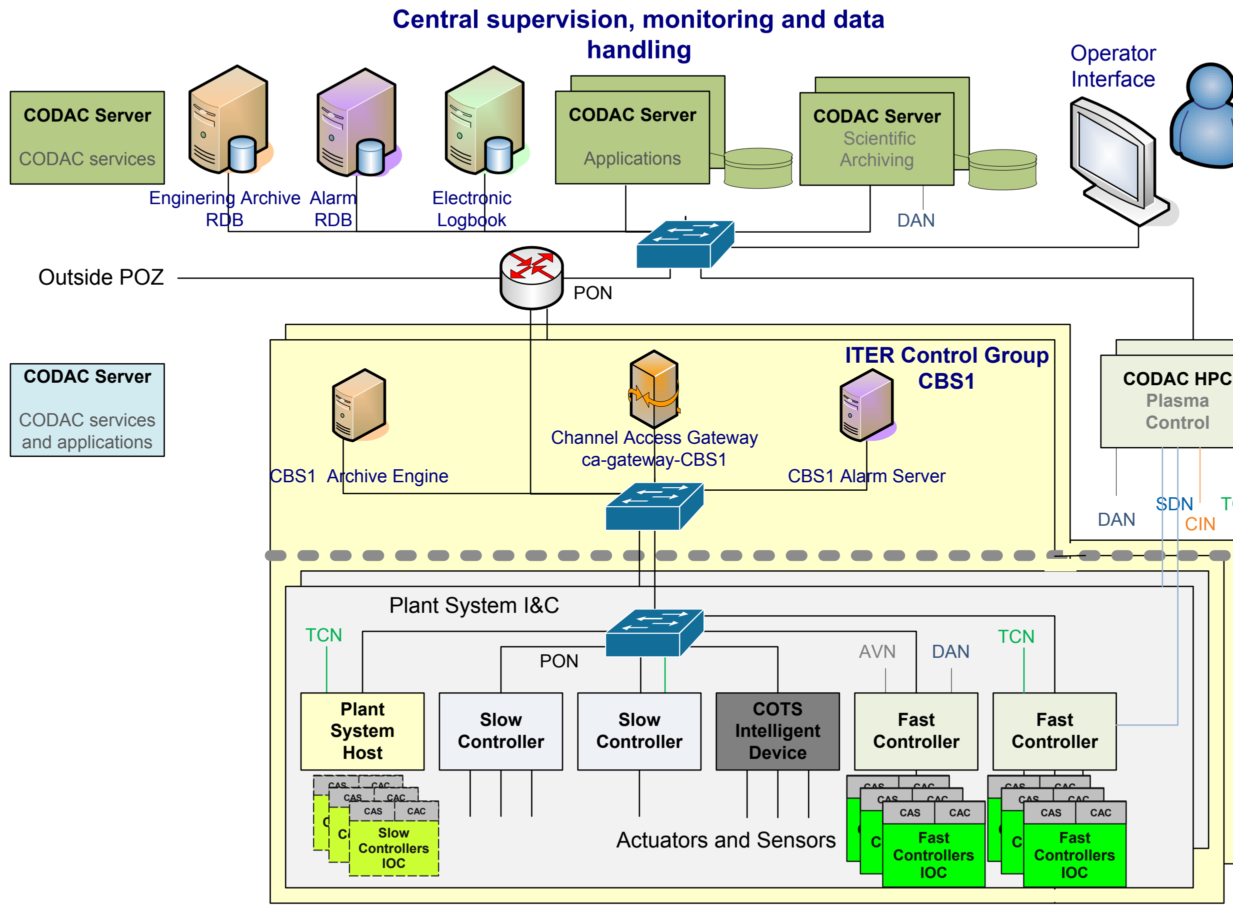

Figure 1: CODAC Services Distributed Architecture showing globally how CSS main components are distributed over the entire architecture.

The advantage of distributing CSS components according to the functional breakdown adopted by CODAC is that this will allow the integration and operation of the local control systems over a long period, privileging scalability and modularity.

This distributed architecture has already been setup in a production environment and one local control system has been successfully integrated for the site electrical power distribution. This real use case gives us confidence in our technical choices and shows the importance of an integrated solution that has to be configured and deployed over the entire distributed architecture. This also requires the definition and enforcement of common operating processes and procedures. With time, different versions of the same functional component in different control groups will have to interoperate, as well as different versions between the central level and the control groups.

ITER CONTRIBUTION TO CSS DEVELOPMENT EFFORT

Industrial Symbols Library

Working on the real use case, it was soon necessary to introduce an industrial symbol library of objects such as electrical circuit breaker, relay, pump or valve to monitor and operate the local control systems. Based on CAD

ISBN 978-3-95450-139-7

symbols, a library of more than 250 industrial symbols was setup in svg and png format and published to the CSS source repository as a new plugin.

Some displays were also developed in a demonstration folder to illustrate the use of these symbols.

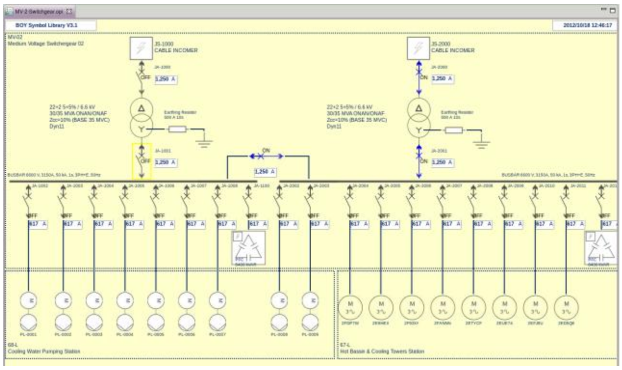

Figure 2: CODAC Electrical Symbols Use Case demonstrating the Industrial Symbol Library.

Subsequently, it was necessary to develop a new graphical widget to handle these industrial symbols and display them in their On/Off, Open/Close or 1/0 position according to an EPICS Boolean PV value. In addition to the symbol position update, the black and white original image has to be redrawn in the configured On/Off color.

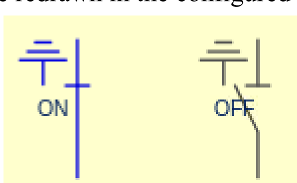

Figure 3: CODAC Electrical Symbol in On/Off Position.

Multistate symbols also needed to be managed even though their use is quite limited compared to the Boolean symbols.

This industrial symbol library and symbol widgets were the first main ITER contribution to CSS development effort and found a direct application in the site electrical power distribution use case. This was also the opportunity to collaborate on existing widgets by adding new functions such as the image rotation and flip.

Alarm Notifier

Still working on the real use case, a request was made to be able to send an email when important alarms were raised by the local control system. This user requirement was translated into the development of a new plugin, client to the alarm system and in charge of the execution of automated actions configured for any alarm triggers – these actions being sending an email, an SMS or just executing a script. This model allows any contributor to develop his/her own automated action. ITER is in charge of the plugin itself, the automated action API and its first implementation - the automatic sending of an email with alarm information.

The alarm notifier plugin receives all alarm state transitions from PV OK to PV in alarm state – minor or major, then to acknowledged PV (for PV configured as to be latched), and finally to PV OK again as well as any alarm severity changes.

Each time the automated actions configured for the trigger PV are executed.

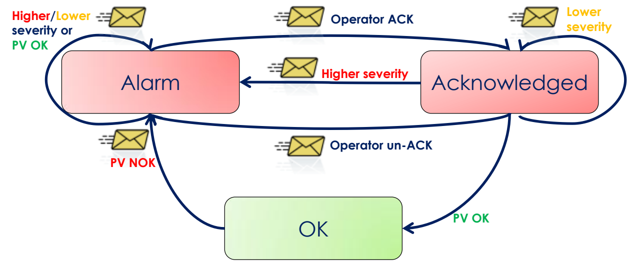

Figure 4: Alarm State Transitions that execute an automated action configured for a latched alarm.

The setup of automated actions on the site electrical power distribution system is on-going and should be operational for October 2013. This should reduce the number of shutdowns as the persons in charge of the maintenance will be notified by email as soon as an important alarm is raised and they can take the corrective action in time.

PV Name Auto-complete

Every laboratory has its own naming convention and ITER is no exception. But during many CSS training sessions, help was requested for entering EPICS PV name to minimise typing errors.

It was decided to develop a CSS core plugin for the auto-completion and allow any contributor to implement his/her own PV configuration database interface, ITER being in charge of the plugin, the auto-complete API and the history name lookup. ITER also provides EPICS database source file parsing for the project imported in the user workspace.

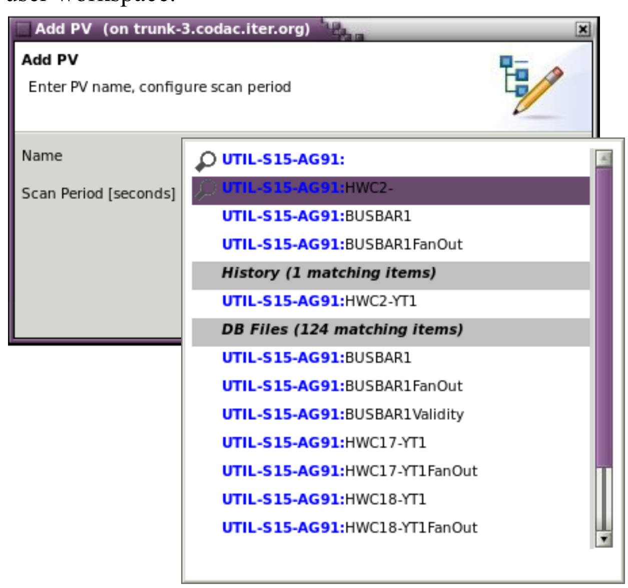

Figure 5: As the user starts to enter a PV name, the auto-complete displays some proposals.

ISBN 978-3-95450-139-7

Web Monitoring Interface

In addition, ITER contributed to CSS development effort by adapting the plot tool and alarm user interface in order for them to be run via a web-browser-equipped device such as a PC, laptop, tablet or smart phone.

ITER Contribution Publication

As first contributions could be sometimes a bit off the mark, ITER needed to publish as quickly and frequently as possible in order to get feedback – this a way to learn from each other, improve and get accepted by the community.

This required the adaptation of CODAC development and packaging infrastructure in order to adopt CSS Mercurial and then the GitHub repository as the main source repository for CSS and synchronise it with CODAC SVN for a nightly build of the framework - the objective being to be able to pull or push back a fix within one day.

FUTURE WORK

During factory and site acceptance tests, the CODAC framework will be used to connect to the local control system to monitor and operate it, to execute sequences of commands, record actions and events and produce reports with runtime statistics.

This will require the integration and possibly adaptation of another CSS service – the experiment automation (Scan Server) which allows editing and the control of the execution of a sequence of commands via CSS. The list of supported commands – Set, Wait, Loop and Log – can be extended for ITER needs via extension points.

Another additional service of interest is the alarm web reporting tool that should help to assess the alarm system by charting the number of alarms per day and displaying the "top ten" alarms for a time period. Usually, it is just one or two EPICS PVs that trigger all the alarms (see Figure 6), so it is then easier to spot those and do something about it.

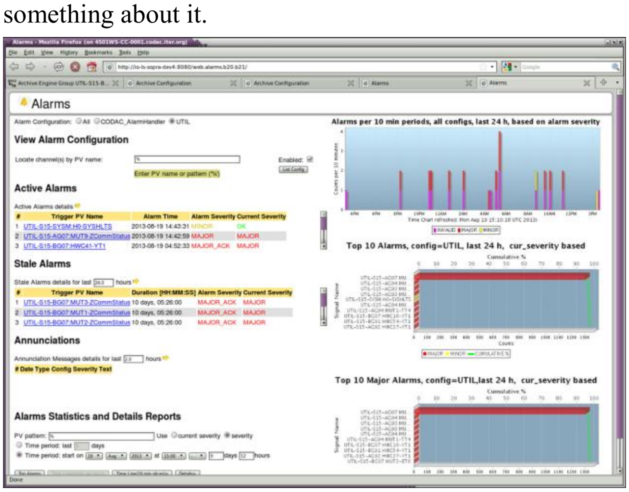

Figure 6: Alarm Web Reporting Tool for site electrical power distribution showing the top ten alarms.

Similarly the archive web reporting tool will provide runtime statistics about the number and frequency of archived samples for each local control system during acceptance tests.

CONCLUSION

Getting involved with the open-source projects ITER is genuinely interested in, is a win-win strategy. CODAC primary target is to setup and distribute worldwide a complete and robust lifetime framework with similar functionality to any commercial SCADA - Supervisory Control And Data Acquisition system. Its software design is based upon the widely-used EPICS control system toolkit and CSS set of services.

Joining the mailing list and starting "small" by reporting and fixing integration issues rapidly became insufficient as new requirements emerged from the first local control system real use case. The development of new CSS plugins for the industrial symbol library and the execution of automated actions for important alarms allowed ITER to get familiar with CSS's various methods for version control, bug tracking, patch submission, coding conventions, and development discussions. Being part of the CSS development effort also means that the new plugins cannot be just "ITER specific" and need to be flexible enough to be adapted by other laboratories via extension points.

The next challenge for ITER is to propose tools in its CODAC framework for factory and site acceptance tests to automate sequences of commands, track the actions and events, and produce reports. Once again, CSS proposes some valuable services that will be integrated and extended to fit ITER requirements.

The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

ACKNOWLEDGMENT

The EPICS worldwide development community played an important role by creating the EPICS set of tools in the first place. Then, Control System Studio Collaboration was an ideal place to exchange and continuously improve CSS in a team spirit.

REFERENCES

[1] CSS Control System Home Page, http://controlsystemstudio.github.io/

[2] A. Wallander et al, "Approaching Final Design of ITER Control System", ICALEPCS 2013, San Francisco, http://jacow.org

[3] F. Di Maio et al, "CODAC Core System, the ITER Software Distribution for I&C" – a Status Report, ICALEPCS 2013, San Francisco, http://jacow.org

[4] EPICS Home Page, http://www.aps.anl.gov/epics/

ISBN 978-3-95450-139-7