ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM002

EPICS JAVA DEVELOPMENTS

K. Saintin†, L. Caouën‡, P. Lotrus
CEA/DRF/IRFU, Saclay, France

Abstract

The CEA IRFU/DIS [1] software control team is involved from feasibility studies to the deployment of equipment covering low level (hardware, PLC) to high level (GUI supervision). For their experiments, DIS software control solution is using two mains frameworks:

MUSCADE, a full Java in-house solution, an embedded SCADA dedicated to small and compact experiments controlled by PLC (Programmable Logic Controller), only compatible with Windows operating system (OS) for the server side.

EPICS [2], a distributed control systems to operate devices such as particles accelerators, large facilities and major telescopes, mostly deployed on Linux OS environments.

EPICS frameworks provides several languages for bindings and server interfaces such as C/C++, Python and Java. However, most of the servers also called IOC developed in the community are based on C/C++ and Linux OS System. EPICS also provides extensions developed in Java such as Appliance (the archiving tool), Phoebus Control-Studio [3] (GUI), and Display Web Runtime (Web Client). All these tools depend on CAJ a pure Java implementation Channel Access Library.

Today, MUSCADE users work under Windows operating system, and they need intuitive tools that provide the same features than MUSCADE in their EPICS projects. Thus, research and development activities mainly focus on EPICS solution adaptation. It aims to explore further CAJ library, especially on the server side aspect. In order to achieve this goal, several developments have been carried out since 2018:

PLCParserTool, PLC simulation to test EPICS synoptic without any hardware.

CAFEJava, Java IOC EPICS to provide process variables connected to any java application. We developed EPICS4MUSCADE bridge and Web Client for MUSCADE on this basis.

INTRODUCTION

IRFU is working on different kind of experiments from particle accelerator installed in big facilities to gas station installed in the middle of the fields. IRFU participates regularly to the construction of accelerators around the world (ESS [4], SARAF [5], and SPIRAL2 [6] …) and its control system team has developed EPICS skills for more than 20 years. EPICS and associated software are mainly designed for big facilities, which is not convenient for some of our experiment; especially small ones based on PLC control

To address this lack of feature, a large number of small control systems (more than 90 devices) are controlled using MUSCADE © Java Software. This framework is custom-built for our laboratory and offers a range of tools to control small-scale facilities and visualize experiments through synoptic (see Fig. 1).

The primary objective for our team is to migrate MUSCADE experiments to EPICS control system. EPICS knowledge is widely shared within a large community of developers and numerous facilities. However, because MUSCADE is a custom-built system, there are only a few persons capable of supporting these experiments.

To facilitate a gradual migration of all MUSCADE experiments, we have developed several tools that allow us to maintain the MUSCADE server-side functionality while leveraging the benefits of EPICS High-Level applications for display, archiving, and alarm notifications. In this article, we will focus on three of these developments and explain how they provide an adapted solution for a reliable migration.

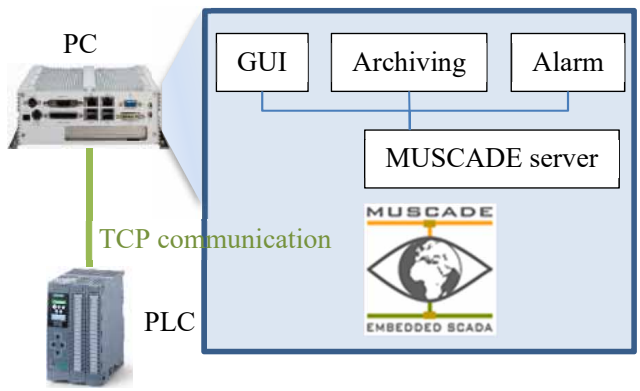

Figure 1: MUSCADE © SCADA an embedded solution.

DXF2BOB CONVERTER

At IRFU, MUSCADE users build their supervision views with AutoCAD for 2D drawings software, especially for cryogenic sequential function chart [7] (SFC or GRAFCET), which can be tricky (see Fig. 2). AutoCAD is a vector drawing solution, which is convenient for precise drawing, and so for complex synoptic. It is also a popular software, especially in the industry, by architects, project managers, engineers, graphic designers, city planners and other professionals.

MUSCADE provides a tool developed in Java, which gives the possibility to import AutoCAD file for generating synoptic views.

† katy.saintin@cea.fr
‡ loic.caouen@cea.fr

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM002

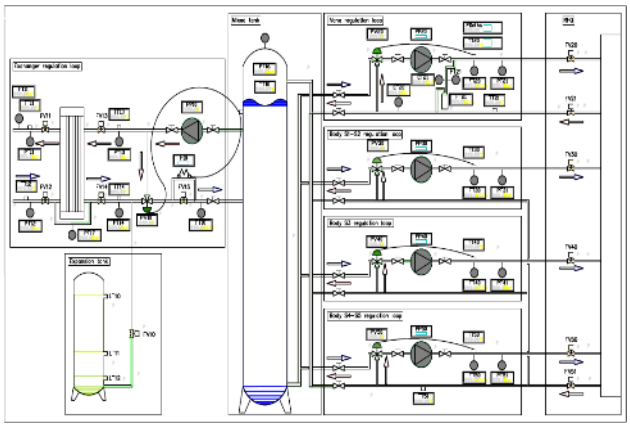

Figure 2: Synoptic design with AutoCAD.

EPICS's framework provides a similar tool for creating animated synoptic views connected to EPICS variables. This tool called Phoebus CS, is user-friendly and is based on a drag-and-drop philosophy.

To avoid to develop twice a GUI in AutoCAD and in Phoebus software; we have developed a tool, which is parsing AutoCAD file and generating a view in Phoebus application. Therefore, this DXF2BOB tool converts an AutoCAD file format (dxf extension) to a Phoebus view file (bob extension) (see Fig. 3).

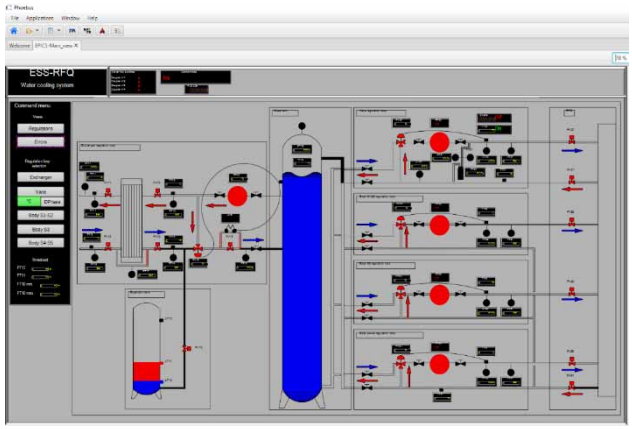

Figure 3: Phoebus synoptic generated from AutoCAD.

Today, this tool has enabled several experiments to initiate the migration from MUSCADE to EPICS, like with SARAF, an accelerator in Israel, and ISEULT, an MRI magnet located at Neurospin in France.

The migration work is still substantial, and to enable a complete migration, this tool must be supplemented by other developments described later in the document.

Indeed, a solution has been developed for the GUI, but it was necessary to find a way to share MUSCADE variables to the EPICS world. To achieve this, we have developed a gateway that facilitates this process.

EPICS4MUSCADE

The EPICS framework provides a Java API (Application Programming Interface) called CAJ (Channel Access for Java), which allows running an IOC (Input Output Controller) server. This server is making available to, all Channel Access clients (the protocol on which EPICS is based), the instrumental variables, also called "process variables".

Typically, EPICS IOCs are programmed in other languages such as Python and primarily in C, but the framework offers alternative bindings such as LabVIEW and Java.

Given that MUSCADE is a full Java solution, it seemed logical to use CAJ API and develop a gateway for sharing MUSCADE variables to EPICS, all at a lower development and time cost.

Therefore, we develop a tool to share MUSCADE variables to EPICS process variables, this program runs as a service and works as a C EPICS IOC. CAJ also supports specifics EPICS fields for managing alarms and archiving.

The initial implementation of this gateway was deployed for monitoring the temperature and humidity levels in our department's building. This temperature monitoring project, carried out with MUSCADE (see Fig. 4), involves hourly readings of the temperature in specific rooms, such as our server room. It also sends notifications to the facility managers when the temperature exceeds a threshold value. This project aims to assess the building's energy consumption in order to optimize energy flows and address energy efficiency challenges.

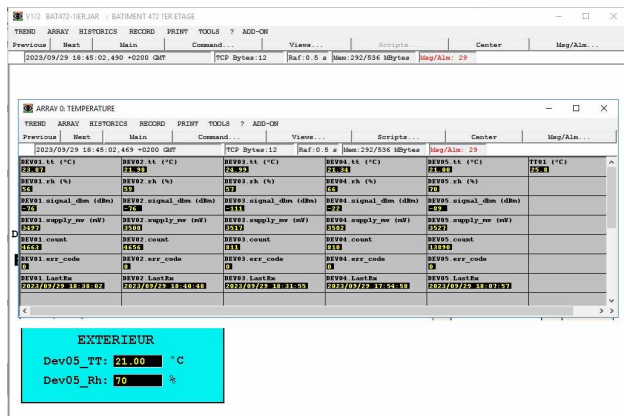

Figure 4: Temperature monitoring with MUSCADE.

The temperature monitoring is carried out through sensors manufactured by the company ATIM [8], specialized in wireless home automation. A probe that communicates via radio waves provides access to various thermal and hygrometric measurements through the MQTT protocol. As the project is currently being controlled with MUSCADE, we have developed a gateway to publish the values via an EPICS IOC. This quick-to-implement solution allows us to benefit from the services provided by the EPICS framework, such as archiving (EPICS Archiver Appliance [9]), displaying views through a web browser (DBWR [10] Display Builder Web Runtime), and alarm management [11]. These services are well-known by our team. This solution enables a gradual migration of the system to EPICS since the entire MUSCADE server part is kept. It also provides additional time for a full EPICS migration (see Fig. 5).

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM002

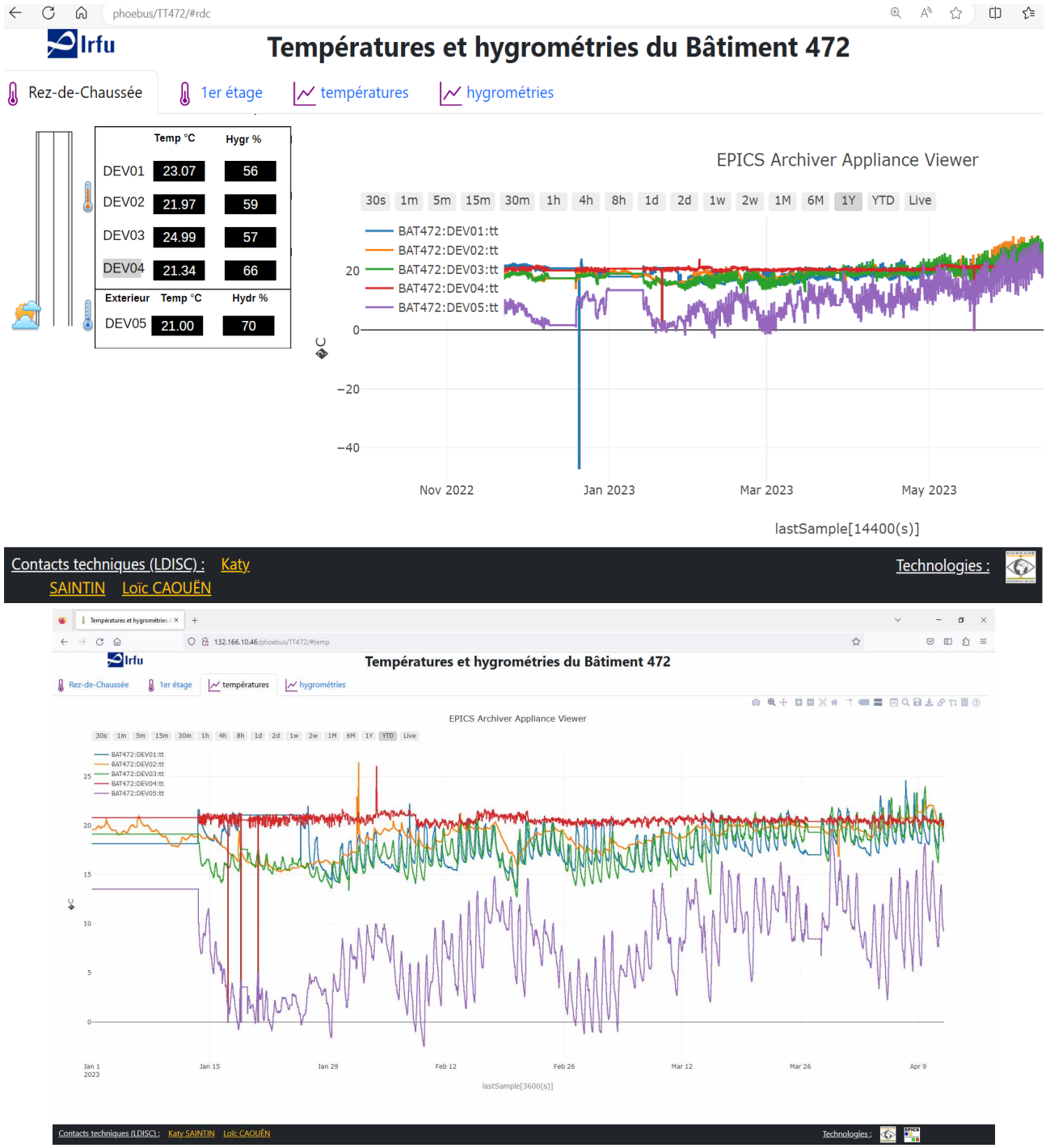

Figure 5: Temperature monitoring with EPICS High Level Application connected to MUSCADE server.

IOC SIMULATION

In some cases, the development of graphical interfaces is ahead of the hardware and low-level development. Indeed, to prototype a project, schematics and view specifications are often drafted at the beginning of the project. As a result, developers require a solution to create user interfaces being disconnected or with no available hardware behind, in order to test their applications.

To address this issue and accelerate testing and development phases for application components, a program using the CAJ API was developed. This program simulates an EPICS IOC by delivering process variables with the same type and names as defined in the project's final nomenclature. The advantage of this system is that once the views are thoroughly tested, they can be switched on the actual system. If the nomenclature has been followed conscientiously, there will be very few adjustments required. This was the case for the cooling system of ESS RFQ Skid. The cooling system views were created and tested at our Saclay site and initially connected with our simulation program.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUMBCM002

Once the views were deployed on the accelerator in Sweden, the supervision views worked flawlessly from the outset. Only a few minor fixes were necessary (see Fig.6).

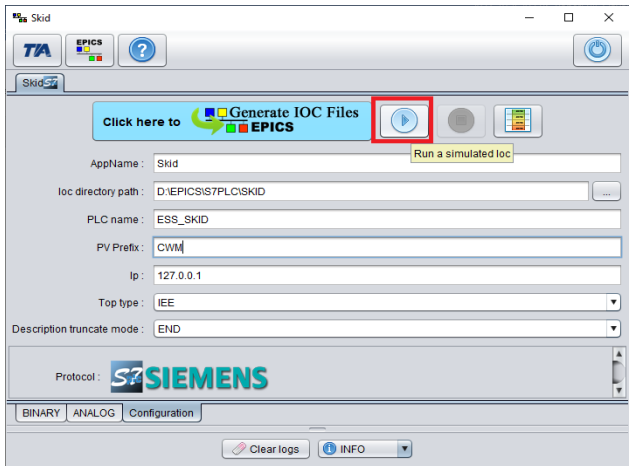

Figure 6: ESS Skid cooling system simulation.

CONCLUSION

In conclusion, all these developments, useful for testing phases, allow for a gradual migration from MUSCADE to EPICS at a lower cost while avoiding regression issues. These tools also accelerate development; however, it is important to consider these deployments as temporary. Whether for simulation or for EPICS4MUSCADE gateway, systems should be fully migrated to EPICS. On the other hand, this article highlights the underutilization of CAJ API capabilities in Java. For instance, we could develop gateways to other control systems that provide a Java binding, such as TANGO control system or on archiver appliance service.

ACKNOWLEDGEMENTS

The authors would like to thank especially Christian Walter for his help and expertise on MUSCADE system.

REFERENCES

[1] Irfu, http://irfu.cea.fr

[2] EPICS, http://epics-conrtol.org

[3] Phoebus CS, https://www.controlsystemstudio.org

[4] A. Gaget and T. J. Joannem, "The Control System of the Elliptical Cavity and Cryomodule Test Stand Demonstrator for ESS", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1545. doi:10.18429/JACoW-ICALEPCS2019-THAPP02

[5] F. Gougnaud *et al.*, "Status of the SARAF-Phase2 Control System", in Proc. *ICALEPCS'21*, Shanghai, China, Oct. 2021, pp. 93-97. doi:10.18429/JACoW-ICALEPCS2021-MOPV001

[6] F. Gougnaud and P. Mattei, "The First Steps of the Beam Intensity Measurement of the Spiral2 Injector", in Proc. *ICALEPCS'09*, Kobe, Japan, Oct. 2009, paper WEP053, pp. 504-506.

[7] A. Gaget and T. J. Joannem, "The Control System of the Elliptical Cavity and Cryomodule Test Stand Demonstrator for ESS", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1545. doi:10.18429/JACoW-ICALEPCS2019-THAPP02

[8] ATIM, https://www.atim.com/en

[9] M. V. Shankar, L. F. Li, M. A. Davidsaver, and M. G. Konrad, "The EPICS Archiver Appliance", in Proc. *ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 761-764. doi:10.18429/JACoW-ICALEPCS2015-WEPGF030

[10] K.-U. Kasemir and M. L. Grodowitz, "CS-Studio Display Builder", in Proc. ICALEPCS'17, Barcelona, Spain, Oct. 2017, pp. 1978-1981. doi:10.18429/JACoW-ICALEPCS2017-THSH303

[11] K.-U. Kasemir, "CS-Studio Alarm System Based on Kafka", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1511. doi:10.18429/JACoW-ICALEPCS2019-WESH2001