ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THCPL04

SCIBORG: ANALYZING AND MONITORING LMJ FACILITY HEALTH AND PERFORMANCE INDICATORS

J-P. Airiau, S. Vermersch, V. Denis, P. Fourtillan, C. Lacombe
CEA - CESTA, Le Barp, France

ABSTRACT

The Laser MegaJoule (LMJ) is a 176-beam laser facility, located at the CEA CESTA laboratory near Bordeaux (France). It is designed to deliver about 1.4 MJ of energy to targets, for high energy density physics experiments, including fusion experiments. It operates, since June 2018, 5 of the 22 bundles expected in the final configuration. Monitoring system health and performance of such a facility is essential to maintain high operational availability. SCIBORG is the first step of a larger software that will collect in one tool all the facility parameters. Nowadays SCIBORG imports experiment setup and results, alignment and PAM control command parameters. It is designed to perform data analysis (temporal/crossed) and implements monitoring features (dashboard). This paper gives a first user feedback and the milestones for the full spectrum system.

INTRODUCTION

Monitoring and analyzing critical data (or indicators) of the LMJ is essential to operate such a facility. It contributes to manage availability and planning actions for the new experiments and follow the global performances for the achieved experiments.

GOALS

The number of equipment to control (from laser bays and target chamber) is very large and logically multiplies the number of indicators to monitor. Each equipment and associated control system generates control data (configuration, log, results, etc.); indicators are extracted from these data directly or indirectly. These data are of different types such as structured (log), semi-structured (scalar results, tabulated data) and un-structured (image, pdf report, etc.). To be able to correlate these data among themselves, it is necessary to standardize their format when they are extracted from the source producer. The data extraction sampling rate is determinant to estimate the database storage capacity and the performance to access to the formatted indicators. In order to speed up the request to the database, some additional information is associated to the data in the form of metadata. These metadata include the specific facility structure (bundle, quad, beam, section, diagnosis), they contribute to simplify the data classification.

Once the extraction and transformation done (data + metadata), the indicators are stored in a database (see Fig. 1). The expected application above this database will perform the following features:

- Computation of complex data from elementary data (z=f(a,b,c,...) , ex.: total energy on target), physical value (transfer function), security factor (vacuum windows damages),
- Monitoring temporal evolution of indicator (elementary data) and performances (complex data or physical value), performing security analysis (security factor),
- Crossed analysis y = f(x) (ex. amplification gain vs energy bank efficiency),
- Smart GUI including rules on data (ex.: warning on threshold overcoming) and multiple types of chart for analysis,
- Dashboard creation based on analysis synthesis
- Predictive analysis: predictive maintenance (planning of optic replacement), calibration (fine tuning of coefficient used on embedded software).
-

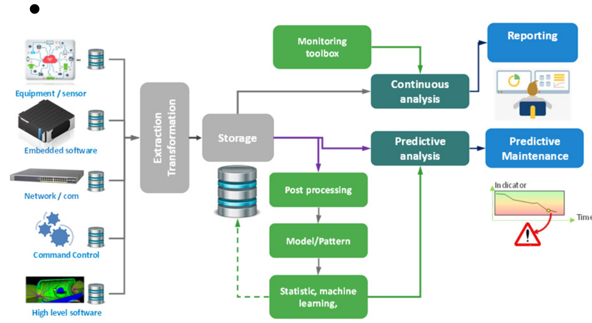

Figure 1: Functional architecture.

Hypothesis

In order to limit the data number and complexity, the type of indicator is limited to structured (log) and semi-structured (scalar and tabulated values) data. The extraction sampling rate is not less than 10 s. This hypothesis is based on the fact that most of the equipment to monitor has a response time bigger than this criterion. The embedded software and operating tools algorithms can generate data at very high frequency but we suppose that these indicators are available as tabulated data at the end of computation. This supposition limits the frequency of the extracting requests.

Data Perimeter

The global architecture of the LMJ facility is structured in layer from equipment to high level software [1].

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THCPL04

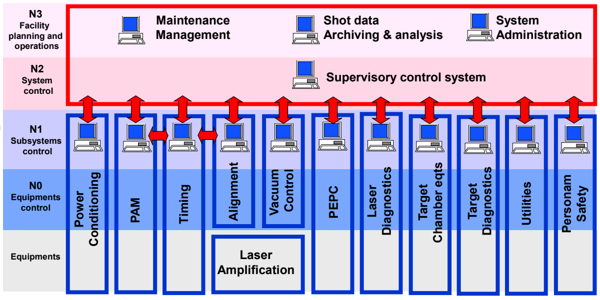

Figure 2: Control system architecture.

The data perimeter covers all the equipment, control (system and subsystem), software and database of the LMJ facility (see Fig. 2). The final application will be able to extract, transform and store all this data:

- Hardware log and IT monitoring,
- Network monitoring,
- L0 level control equipment (log and I/O parameters),
- L0 embedded software (log and I/O parameters),
- L0/L1 communications,
- L1 operating software,
- L2 operating software,
- Facility configuration parameters,
- Raw and transformed results,
- Physical values.

Development Strategy

This application can't interfere with the LMJ operations but must follow the evolution of the facility. The development and deployment strategy must respect these rules:

- Independence: no impact on the facility performances, no modifications on existing systems
- Scalability: adaptation to bundle extension (up to 22 bundles)
- Sustainability: must operate more than 10 years.

This software is developed in an iterative way to avoid user's resistance against new global software and to reach as many users as possible. The first release LEOPOLD (2015) performs the following features:

- Extraction on a limited perimeter: raw and transformed results, N1 Alignment embedded software result,
- Indicator visualization and analysis tools.

This stand-alone tool, developed in JAVA/JavaFx/H2, has quickly shown some performance limitations when using more than 2 bundles.

The second release: SCIBORG (French acronym for scientific database for reporting and graphical representation) has been initiated in 2016 and has been commissioned this summer. SCIBORG covers 80% of the expected features but on a limited perimeter.

SCIBORG

Software Architecture

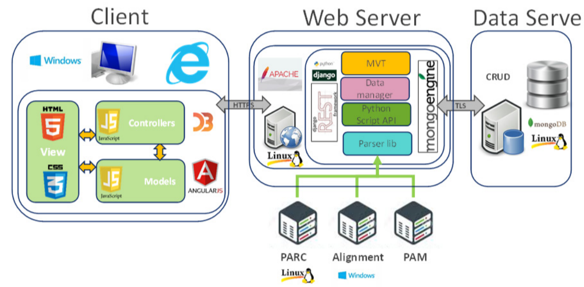

Figure 3: SCIBORG software architecture.

Based on our feedback with similar application, client/server architecture is a good option for this kind of multiuser tool (see Fig. 3). The technical choice has been chosen for the following reasons:

- Client :
- Easy access, deployment and longevity: WEB interface (Angular JS [2]),
- Rich web-oriented visualization toolkit: D3JS [3] (feedback from lab's developers),
- Server
- Django/python (feedback from PARC [4]
- Database: MongoDB [5] (API (Django/python), easy deployment and scalability (sharding)).

Parser and Template Data

Data to parse are described in an excel sheet called "template data file". Each line of the sheet represents a template indicator. The data's source indicates which parser to use. The extraction mode (by shot/automatic) set the extraction sampling rate. Three fields determine the remote physical localization of the data (physical path to the remote file, key path in the file, name). A unique identifier (SciborgId) is created with the previous localization information. Different types of data (integer, double, URL, string, etc.) can be stored in the application (see Fig. 4).

Data's granularity indicates the number of instances to generate on the basis of the template (ex.: granularity = beam -> 8 instances by bundle). Template field containing wildcard are replaced by the real instance value (ex.: beam template wildcard ${Bn} takes values in [B1, B2, B3, B4] for each beam path).

The instantiation process depends on the experiment bundle list and/or the operational bundle list when the indicator is not extracted during a shot sequence.

In addition to the previous metadata (source, type, etc.), these ones have been added to each indicator to help indexation/classification/display:

- data creation date: the date and time when the data source producer creates the data,
- measurement point (PAM, amplification section, final optic assembly, target, etc.),

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THCPL04

- Step (pre-sequence, rod shot, power shot, etc.),
- Shot Identifier (if existing),
- Units, display option, etc.

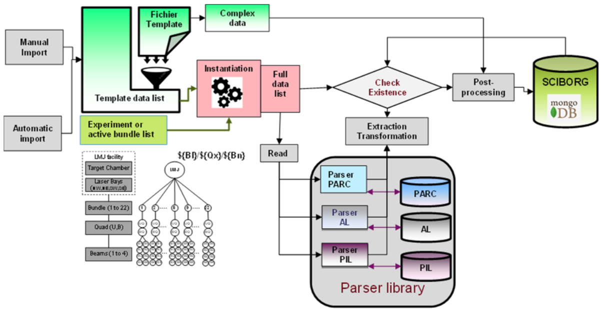

Figure 4: Modular parser.

Once the indicators are instantiated and sorted by source, they are transmitted to each parser for extraction (python plugins). This extraction process is multithreaded for quad and beam granularity (ex.: indicators at beam level are extracted simultaneously). All the extracted data are gathered in a list before insertion in the database (sequential).

Complex Data

The specific indicator expected for monitoring health and performances are not all available directly. To respect the constraint of independence (see development strategy), SCIBORG must be able to create new indicator as a combination or computation of the existing ones in the database. A simple API (to access existing data) and an editor (to change input parameters) have been developed in the software. SCIBORG uses two imported files to define a complex data:

- python script file: to do the computation according to the API
- XML definition file: containing type, name and default value of input and output parameter and description of the complex data.

This mechanism is based on the feedback of computation scenario definition used in PARC [6]. The complex indicator used elementary indicator (ex.: list of beam energy on target (expected, measured), type of shot), user input parameter (ex.: date of equipment retrofit) to compute a new value. The complex indicator must be inserted in the template definition file with its metadata (see Parser and template data).

The compute of "the standard deviation of difference on expected target energy for power shot since the equipment retrofit" is defined as:

$$DE_i = \frac{Abs(\sum_{s=1}^{n} E(measured)_i \; - \; \sum_{s=1}^{n} E(expected)_i\,)}{\sum_{s=1}^{n} E(expected)_i}$$

$$DE_i = \mathit{Energy\ difference\ by\ shot\ ater\ the\ specified\ date}$$

$$n = \mathit{bundle\ number\ by\ shot}$$

$$\mathit{complex\ data\ value} = \sqrt{\frac{\sum_{i=1}^{m}(DE_i - \overline{DE})^2}{(m-1)}}$$

Temporal and Crossed Analysis

The analysis toolbox must be able to plot curves as temporal function (y=f(t)) or as crossed function y=f(x) where x and y are indicators stored in the database.

The data sampling rate is not the same for all indicators (ex.: laser energy diagnosis data = shot by shot vs. manual configuration software = undetermined rate). The granularity of the indicator can be different (beam vs. quad). The sample number between the two axes (x, y) is not consistent for plotting. To solve this issue, a double padding algorithm create artificial sample to generate consistent grid on each axe of the curve (see Fig. 5).

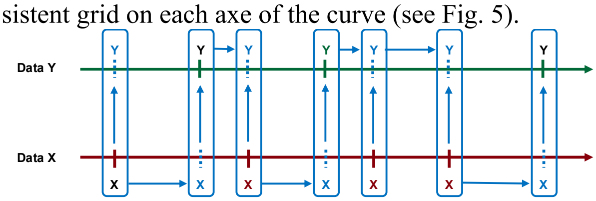

Figure 5: Double padding algorithm.

The multiple instances can be plot on the same graphical representation for comparison. A selection tool automatically associates the corresponding instance:

ex.: Bundle 01, Upper Quad, Beam 1 = BL01/UQ/B1

$x_{BL01/UQ/B1} = f(y_{BL01/UQ/B1})$

User can change the automatic settings to create crossed granularity plot (ex. beam splitter analysis: $x_{BL01/UQ/B1} = f(y_{BL01/UQ/B3})$. Different types of plot are available (standard chart, histogram, box-and-whisker, etc.). For each plot (if type is consistent), some standard statistic values are computed (min, max, mean, standard deviation, etc.) (see Fig. 6).

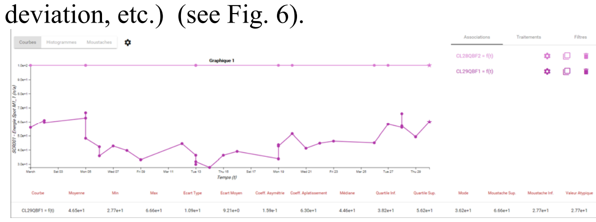

Figure 6: Analysis plotting.

On each set of data (x,y), some filters can be applied to reduce the number of sample : temporal windows, null or invalid value, dependency on other value (z) criterion (ex.: z < constant).

The analysis is defined as a model. A new user analysis is an instance of the model containing the user selection (x, y definition, granularity association, etc.) and preferences (type of plot, colors, etc.). It can be saved as a file and reload with an updated set of value (ex.: day 1 = creation of the analysis includes 10 samples in database for x and y, day 2 = reload the analysis with 20 samples in database).

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-THCPL04

Dashboard

Dashboard offers a synthetic view of one or more indicators for performances (EP diagram), security (damaging threshold) or maintenance (debris shield replacement) purpose.

A simple editor allows users to create a dashboard by inserting comments, indicators, saved analysis and external data (jpeg) in the available layout (2 models). Like the analysis, a new dashboard can be saved as a file and reload with an updated set of values. The update is only effective on indicators and analysis associated to the dashboard.

User Feedback

User interest for such a tool is undeniable. The most extensive user group is the laser's engineer lab members. They have initiated this project (LEOPOLD) and specify most of the features of SCIBORG. SCIBORG has proved its utility in monitoring the health and performances indicators. It helps to identify degraded mode by analyzing indicators that break rules. By example, the alignment process uses equipment known as the common reference. This equipment is composed of a sensor illuminated by a laser to perform the alignment of the target. Monitoring the intensity of the focal spot on the sensor has been useful to detect a malfunction of the laser and the need to replace it.

"LMJ DOCTOR" FULL SYSTEM MILESTONE

As a doctor, the final tool will help to investigate diagnosis and suggest a cure to an LMJ "illness".

This third release is under investigation to satisfy all the expected features including the full data perimeter and an interaction with predictive analysis tools. The first meetings made last month have demonstrated the need of a common tool that centralized the indicators. The continuous analysis (multidata monitoring) on the full data perimeter (from hardware to low- and high-level software) is the most wanted feature. Users hope to speed up issue analysis and fix with the help of transversal synthetic view of their domain.

A second important feature is the API between the database and the predictive toolbox (data mining, pattern matching, deep learning). Machine learning demonstrators are in development for the PAM temporal shaping tuning and damage growing. These demonstrators use "offline" data (old set of LMJ data or bench data) because the access to these data is not easy (manual upload from equipment) and/or not available on a long time (not stored). This API will make the access easier to the "online" data.

This tool has to be accessible outside the shoot room to perform "rear base" analysis (ex. user office).

The development strategy of the future system starts with identifying all the indicators by interviewing experts. This identification phase is expected to be done for the end of the year. Depending on the number of indicator and the complexity of their extraction, the parsing process could increase the network traffic and slow down the operating time for shot sequence. The storage tools sizing is directly linked to this number of indictor and the retention time expected. For the specifications, we will request support from big data and network experts. The achievement of specification phase is planned for the end of 2022. LMJ doctor will be operational in 2025 when all the 22 bundles will be commissioned.

REFERENCES

[1] J. P. A. Arnoul and J. I. Nicoloso, "The Laser Megajoule Facility: Control System Status Report", in *Proc. 14th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'13)*, San Francisco, CA, USA, Oct. 2013, paper TUCOAAB02, pp. 487-489.

[2] JavaScript-based open-source front-end web framework, https://angularjs.org/

[3] Data Driven Documents - JavaScript library for manipulating documents based on data - Mike Bostock. https://d3js.org/

[4] S. Vermersch, "The Laser Megajoule Facility: The Computational System PARC", in *Proc. 15th Int. Conf. on Accelerator and Large Experimental Control Systems (ICALEPCS'15)*, Melbourne, Australia, Oct. 2015, pp. 38-41. doi:10.18429/JACoW-ICALEPCS2015-MOC3O06

[5] The database for modern applications - MongoDB, Inc. https://www.mongodb.com

[6] J-P. Airiau *et al.*, "PARC: A Computational System in Support of Laser Megajoule Facility Operations", in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17)*, Barcelona, Spain, Oct. 2017, pp. 1034-1037. doi:10.18429/JACoW-ICALEPCS2017-WEAPL05