USING AN EXPERT SYSTEM FOR ACCELERATORS TUNING AND AUTOMATION OF OPERATING FAILURE CHECKS

M. Ounsy, S. Pierre-Joseph, G. Viguier, Synchrotron SOLEIL, France
E. De Ley, Isencia, Belgium

Abstract

Today at SOLEIL abnormal operating conditions cost many human resources involved in plenty of manual checks on various different tools interacting with different service layers of the control system (archiving system, device drivers, etc.) before recovering a normal accelerators operation. These manual checks are also systematically redone before each beam shutdown and restart. All these repetitive tasks are very error prone and lead to a tremendous lack in the assessment of beam delivery to users. Due to the increased process complexity and the multiple unpredictable factors of instability in the accelerators operating conditions, the existing diagnosis tools and manual check procedures reached their limits to provide practical reliable assistance to both operators and accelerators physicists. The aim of this paper is to show how the advanced expert system layer [1] of the PASSERELLE [2] framework, using the CDMA API [3] to access in a uniform way all the underlying data sources provided by the control system, can be used to assist the operators in detecting and diagnosing abnormal conditions and thus providing safe guards against these unexpected accelerators operation conditions.

INTRODUCTION

Using Passerelle as a Model Driven Architecture programming environment and its integration with JBoss drools for modeling workflows based on event processing and rules' matching provides us a good platform to build step by step an "expert system" for diagnosing Accelerator abnormal operation conditions.

Processes (a.k.a sequences) developed to this aim will act on data collected from various data sources (control system, archiving system, experimental data) .This can add complexity to rules definition. Accelerator's operators who are expected to elaborate these rules must get rid of the complexity related to the programming technical details needed to access a given source of data. Integrating CDMA within Passerelle as an abstract data access layer will show us how this goal can be achieved.

After presenting overviews of the three building blocks of the system, Passerelle, JBoss Drools and CDMA we will show them working on an accelerator operational diagnosis example and then come to the conclusions.

PASSERELLE

Passerelle is a toolkit for designing sequences (and more generally data workflows) in a "drag and drop" graphical environment. Its core functionalities are based on the Java technology standards.

To design/develop/program sequences, called models, ISencia company provides a graphical IDE (Integrated Development Environment) (see Fig. 1). The IDE offers an execution engine as well as a number of essential framework services which will be described below.

Using Passerelle graphical IDE, a given process can be easily mapped from its functional design onto a graphical model of inter-connected components. The solution model can be gradually refined, starting from high-level composite components, to define how each composite component can be assembled from more elementary building blocks. Models can be immediately tested inside the Passerelle IDE, after which they can be directly deployed to the Passerelle model executors.

In the Passerelle process engine, a workflow is defined by a graphical assembly of actors that each performs a step of the complete process (Fig. 1). These graphical models are stored in XML files. Each actor has a well-defined interface/responsibility and is completely functionally independent from the other actors. There are actors available to control Tango devices, to make routing decisions in the workflow, to query databases or fetch experimental data from HDF5 [4] files.

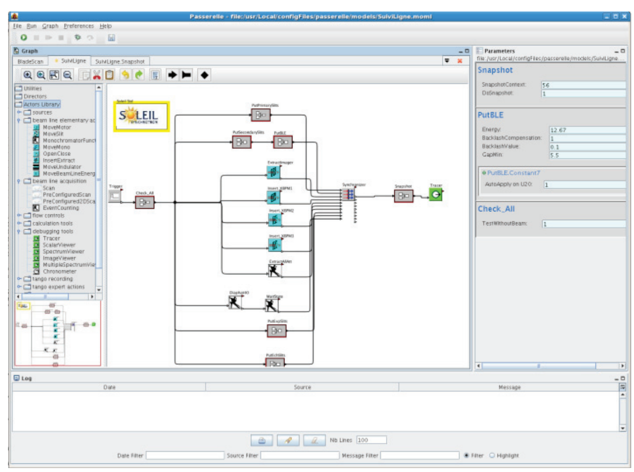

Figure 1: A step by step scan sequence.

Hence within Passerelle, all processes are defined in graphical models and can be launched in different execution environments. The most high level execution environment is provided by the Passerelle Manager, a web-based process automation server platform.

EXPERT SYSTEM LAYER

JBoss Drools was integrated within Passerelle to cope with "expert system" like modelling. That is to cater with automated analysis and decision where the needs are such as:

- Dynamic routing and filtering logic based on previously obtained information
- Automated validation of results obtained in previous steps.
- Correlating acquired data and elementary interpreted results to obtain a final explanation to a problem situation.
- Automated advises for corrective actions in case of problems.

The above needs could be answered by traditional programming using obvious "if-then" logic constructs but these kinds of programs become quickly almost impossible to adapt for complex content-dependent logic operating on large volumes of data. Alternatively, rule engines are more specialized software systems for addressing these kinds of problems. A rule engine can be viewed as a sophisticated if/then statement interpreter. The inputs to a rule engine are a "rule execution set" and some "data objects". The outputs from a rule are determined by the inputs and may include the original input data objects with possible modifications, new data objects and side effects.

The central part of such a system is an inference engine that is able to match rules against facts or data to come out with conclusions which result in new facts or actions, see Fig. 2.

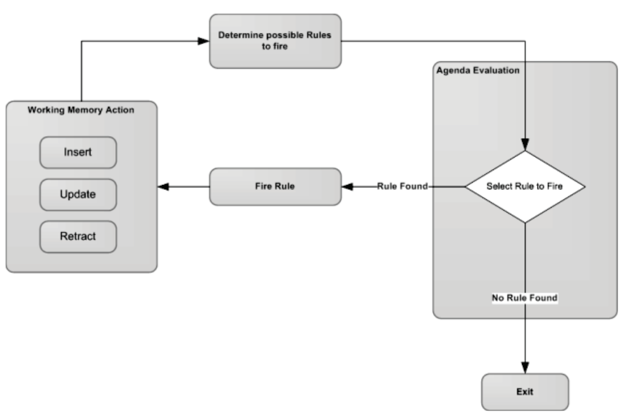

Figure 2: Iterative inferencing.

JBoss Drools is the leading open-source rules engine written in Java and it is easy to integrate in any Java application via:

- Standard JSR-94 Java Rule Engine API
- Drools' own "Knowledge API" with more advanced capabilities

Passerelle uses the Drools Knowledge API and has specialized actor libraries to integrate the usage of the Drools rule engine in an automated process. In this way it is possible to apply rules to analyse the results of the work done in the previous steps of the process. Then we can act on these results using the common paradigm of sequences and actors. Passerelle Manager stores all analysis results *(including timed task execution traces)* in a relational database. The web UI includes rich views on this data to help with application development and to post-analyse the execution traces of models.

CDMA LAYER

The CDMA is a core API that accesses data through a data format plug-in mechanism and data content definitions (sets of keywords). Using an innovative "mapping" system between content definitions and physical data organizations, the CDMA allows client application developers to access these data in a uniform way independently of the data file container AND schema. Each provider of a given data source has to develop a data access plug-in for its own data file formats along with the mapping between content definitions and its data files, see Fig. 3.

The CDMA includes:

- A client layer API for writing client applications accessing data
- A developer API to build the data access plug-in

Even if the CDMA abstracts the data source physical format for the data reader, this one still has to bother with the way this data is organized inside its file support. It means that he has to know precisely the data schema. To solve this issue the CDMA introduces an innovative way for accessing data. This is the dictionary mechanism.

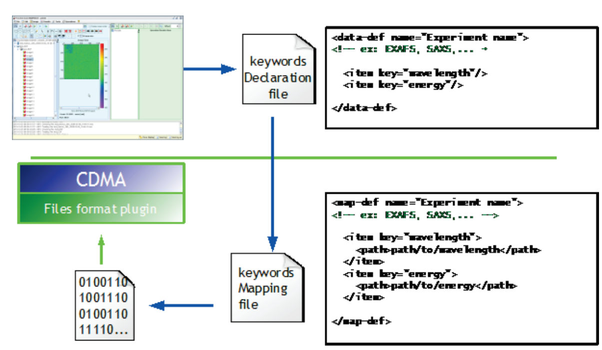

Figure 3: CDMA.

The dictionary mechanism relies on two XML files:

- a "Data Definitions" file which is a set of keywords matching data items contained in the data source
- a "keywords mapping" file which is the dictionary itself mapping between the keywords and real paths to the addressed data items.

ISBN 978-3-95450-139-7

FIRST OPERATIONAL SCENARIO ON SOLEIL ACCELERATORS

The monitoring and tuning of thousands of parameters is essential to drive high-performing accelerators and beamlines. To keep track of large volumes of these technical data, the archiving service [5] is a key component of the Tango [6] control system used at Soleil. This service is used for daily operation such as:

- Vacuum, temperature follow-up
- Insertion devices follow-up…

The archiving service collects Tango control data (also named Tango attributes) to log them in a database. Data collection is done by Tango devices called Archivers (see Fig. 4).

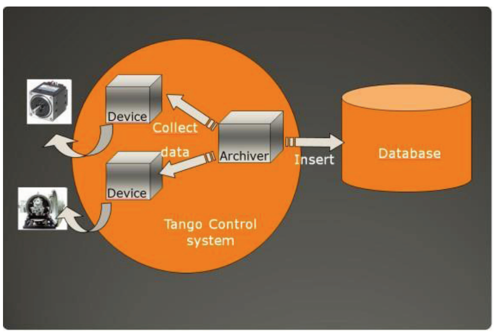

Figure 4: Tango archiving data collection.

Abnormal behaviour of the archiving service can result in loss of logged data which can become very critical if this data is related to personal safety for example. Here comes the need for a diagnosis tool aware of which attributes are to be archived and within which period and monitoring the database itself independently from the Archivers to see if data is really collected and persisted.

This diagnosis tool must also try if possible to answer why a given attributed is not archived (KO attribute) or why it is logged with null values (NULL attribute).

CURRENT MANUAL DIAGNOSIS

The Tango archiving comes with a Watcher device that regularly triggers the collection of potential archiving problems for defined attributes. The results are stored in a dedicated table in the Archiving database.

Control room operators have a view on the Watcher, showing the list of problematic Tango attributes.

In order to correct the situation, following steps must be performed manually:

- Consult the Tango DB, using the Jive UI, to check if the *devices* for the problematic attributes are *registered* and if they are *started*.
- For problematic attributes where the parent device is registered and started, the problem cause must be with the attribute itself.
- The operator must then check each *attribute's presence* and *metadata*: read/write access, data type and data format.
- This involves manually checking the Tango DB again using Jive. The presence of the attribute in the current device registration must be checked comparing each time its metadata in the Tango DB registration with the attribute metadata in the Archive DB.
- Finally the root causes for problematic attributes can be categorized :
    - Device not configured
    - Device not started
    - Unknown attribute on device
    - Archived metadata differs from actual metadata in Tango DB

DESIGN OF THE SOLUTION

The above steps involve repetitive and error-prone tasks where the operator must compare views in different tools. Thanks to an integration of Tango, CDMA and Passerelle, it is possible to automate most of this work to obtain an integrated diagnostic process.

Passerelle supports the following types of activities in such a diagnosis (and repair) process:

1. Collecting and consulting the available data
2. Analysing the problem
3. Formulating decisions / diagnoses
4. Preparing advised actions to repair the problematic situation
5. Executing the actions

In the scope of this paper, the process stops at step 3. It is the operator's responsibility to actually decide on the execution of any required repair actions.

Data Collection

Data collection must be performed on three types of sources:

- Archive DB
- Tango DB (accessed as a device)
- Individual Tango devices

The archive DB can be accessed via CDMA. The Tango DB and other devices can be accessed via the Tango APIs. For each type of data source, Soleil has developed the corresponding Passerelle actors. These can be flexibly reused and combined in any required diagnostic process, besides their existing usage in control sequences.

Problem Analysis

The problem analysis involves the integration of rules-based logic and decision trees. This is provided in Passerelle through the integration of the JBoss Drools engine as described above. Concretely in this scenario we have 2 analysis steps:

ISBN 978-3-95450-139-7

- ANALYSE ATTRIBUTES:
  Identifying KO and NULL attributes, grouping them per parent device.
- FILTER ATTRIBUTES for OK DEVICES: Generate list of problematic attributes that must be further analysed, i.e. where the parent devices are working fine.

Diagnosis

The diagnosis is also implemented via rules-based logic. It collects all analysis results and creates a final diagnostic report, identifying the root causes as described above.

IMPLEMENTATION

The above has been implemented in the following model, Fig. 5:

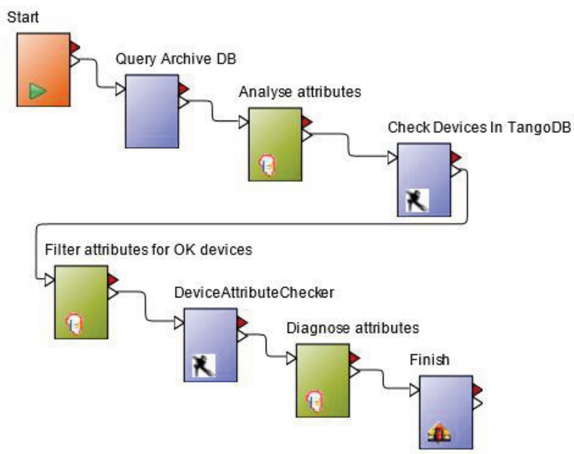

Figure 5: Graphical model of diagnosis.

Such a graphical model can be used as a work instrument between the different stakeholders: operators, scientists, IT etc.

This flow can be executed on-demand or in an automatically scheduled way. The execution traces for one run can be as below, Fig. 6:

Id	Creation TS	Status	Initiator
1877	2013-09-12 11:40:54	FINISHED	.Diagnose attributes
1876	2013-09-12 11:40:53	FINISHED	.DeviceAttributeChecker
1875	2013-09-12 11:40:53	FINISHED	.Filter attributes for OK devices
1874	2013-09-12 11:40:53	FINISHED	.Check Devices In TangoDB
1873	2013-09-12 11:40:53	FINISHED	.Analyse attributes
1872	2013-09-12 11:40:53	FINISHED	.Query Archive DB

Figure 6: Execution traces for one run.

This indicates that the run was done without errors for each step. Detailed results for each step can be consulted as well. For the final diagnosis step, the results show the root cause for each problematic attribute, Fig. 7

sys/tg_test/1/double_scalar	Attribute metadata error
sys/tg_test/1/double_scalar.DATA_FORMAT	0
sys/tg_test/1/double_scalar.DATA_TYPE	Error : archived different from actual 0<>5
sys/tg_test/1/double_scalar.WRITABLE	Error : archived different from actual 0<>3
sys/tg_test/1/short_scalar	Attribute metadata error
sys/tg_test/1/short_scalar.DATA_FORMAT	0
sys/tg_test/1/short_scalar.DATA_TYPE	Error : archived different from actual 0<>2
sys/tg_test/1/short_scalar.WRITABLE	Error : archived different from actual 0<>3
test/motor/1-1/offset	Device not configured
test/motor/1-1/position	Device not configured

Figure 7: Results showing root causes.

The device test/motor/1-1 has not been configured in the Tango DB.

For the sys/tg_test/1 it seems that the operational device does no longer correspond with the original metadata in the Archive DB, for two attributes.

CONCLUSION

The new Soleil Expert System, integrating CDMA, Passerelle and Tango has been applied in a concrete scenario to support daily operational monitoring. It combines traditional process automation with knowledge-based technology.

For this first scenario of "Tango archiving system monitoring" the next step will be to automate the repair phase for simple cases.

Then 2 major applications are foreseen to enhance beam availability.

- The first one will aim to increase beam stability by analysing the various insertions devices positions during the last hours to be able to make appropriate beam orbit corrections.
- The second will analyse the data collected during a beam loss on the various accelerator sensors (like BPM , Transverse Feedback System, Radio Frequency signals, etc .) to give the operator the best diagnose on the reason for beam loss so that next injection can be done as quick as possible with a trustable diagnosis.

REFERENCES

[1] E. De Ley, D. Jacobs, "Rules-based Analysis with JBoss Drools: Adding Intelligence to Automation" Icalepcs, Grenoble, France, October 2011.
[2] G. Abeillé, M. Ounsy, A. Buteau "A Graphical Sequencer for Soleil Beamline Acquisitions" Icalepcs, Knoxville, Tenessee, October 2007.
[3] S. Poirier et Al "Common Data Model Access: A Unified Layer to Access Data From Data Analysis Point OF View" Icalepcs, Grenoble, October 2011.
[4] http:// http://www.hdfgroup.org/HDF5/
[5] G. Abeillé, S. Pierre-Joseph, J. Guyot, M. Ounsy "Tango Archiving Service Status" Icalepcs, Grenoble, October 2011.
[6] http://www.tango-controls.org

ISBN 978-3-95450-139-7