ISBN: 978-3-95450-132-8 doi:10.18429/JACoW-IPAC2014-THPRI095

TESTING QUALITY AND METRICS FOR THE LHC MAGNET POWERING SYSTEM THROUGHOUT PAST AND FUTURE COMMISSIONING

D. Anderson, M. Audrain, Z. Charifoulline, M. Dragu, K. Fuchsberger, J.C. Garnier, A.A. Gorzawski, M. Koza, K. Krol, S. Rowan, K. Stamos, M. Zerlauth, CERN, Geneva, Switzerland

Abstract

The LHC magnet powering system is composed of thousands of individual components to assure a safe operation when operating with stored energies as high as 10GJ in the superconducting LHC magnets. Each of these components has to be thoroughly commissioned following interventions and machine shutdown periods to assure their protection function in case of powering failures. As well as having dependable tracking of test executions it is vital that the executed commissioning steps and applied analysis criteria adequately represent the operational state of each component. The Accelerator Testing (AccTesting) framework in combination with a domain specific analysis language provides the means to quantify and improve the quality of analysis for future campaigns. Dedicated tools were developed to analyse in detail the reasons for failures and success of commissioning steps in past campaigns and to compare the results with newly developed quality metrics. Observed shortcomings and discrepancies are used to propose additional verification and mitigation for future campaigns in an effort to improve the testing quality and hence assure the overall dependability of subsequent operational periods.

INTRODUCTION

Thousands of individual components comprise the LHC magnet powering system and each of these needs to be thoroughly commissioned and tested. The analysis of the results from these system tests are conducted manually to ensure that all of the components have been properly tested. This requires a verified individual to look at thousands of test results to determine if each one of them has passed or failed. This was a good approach to build up confidence in the results of our tests. However dealing with a large number of tests can be problematic, as in the case of the LHC, where more than 10,000 tests have to be executed. The analysis of most of these tests involves simple checks to ensure that the value of the attributes fall within defined limits.

There are well defined test procedures in place which can apply to both manual and automated testing processes.

- Are these test procedures properly covering all of the possible scenarios which could affect the systems?
- Can we somehow assert and measure the quality of these test procedures?
- The AccTesting framework has been in use since 2011, but is it enough to ensure the quality of testing procedures?

This paper will try to answer these questions based on the experience gained from working with the AccTesting framework and attempt to quantify exactly how effective the test procedures have been in the past and propose improvements for the future.

In the first section, we will give a short overview of the system architecture. In the subsequent section we will describe the measures that were taken to improve testing quality and give an outlook for the future.

ARCHITECTURE

The AccTesting Framework consists of several modules as depicted in Figure 1.

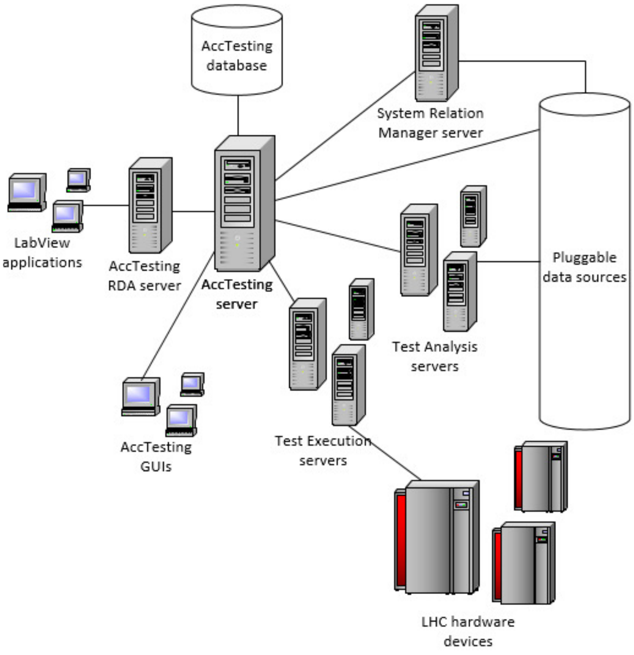


Figure 1: The architecture of the AccTesting framework. The system is extensible and can be parametrised with the data (systems information, relations, parameters and other testing lifecycle details) from different data sources.

The most important part of the system constitutes the server which coordinates actions of all system components and notifies all clients about the current status of operations. The AccTesting server was carefully designed and developed to centralize all the management of the testing lifecycle. The AccTesting server coordinates the whole testing logic and delegates a large part of the work to dedicated machines which perform specific functions like test execution or result analysis.

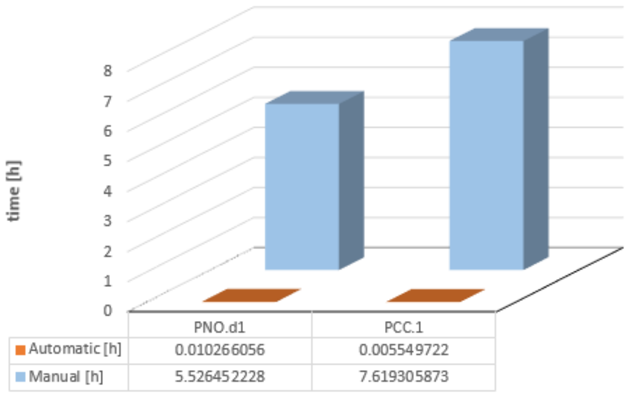
The modular design of the framework eases the system configuration, adjusting it to specific use cases and performance requirements. From the test definition point of view the most important aspect is the extendable server interfaces which enable users to plug in different implementations which satisfy even the most complex use cases. The AccTesting framework also provides external interfaces (through the AccTesting RDA server) for non-Java clients (like LabView applications) using standard CERN middleware libraries which allows other CERN applications to communicate with the AccTesting framework.

A domain specific language (eDSL) [1] has been added to the framework to allow non programmers to easily formalize analysis procedures. This eDSL simplifies the process of creating test conditions, in the form of assertions, as well as specifying what parameters should be included. All assertions are executed on test analysis servers with device data which can be provided by different data sources.

AUTOMATED ANALYSIS

Assertions which have been created with the eDSL can be used as the basis for an automated analysis module. These modules can automatically perform the analysis phase of testing and reduce the complexity of testing procedures as well as bringing large performance increases. This phase is known as Automated Analysis.

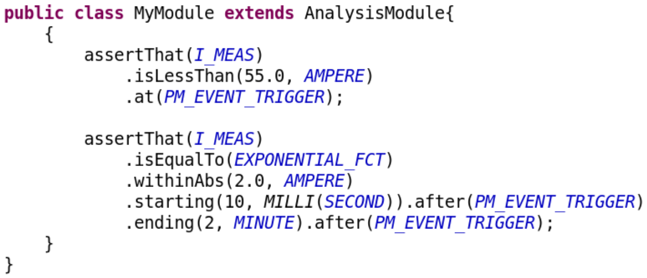
In order to quantify the performance improvements of automated analysis, two LHC magnet powering tests (named PNO.d1 and PCC.1) were run multiple times with the automated analysis module to work out an average analysis time. The average analysis times are shown on Figure 2. When performing similar tests with a manual approach, this can require many hours of waiting for analysis to be completed [2].



	PNO.d1	PCC.1
■ Automatic [h]	0.010266056	0.005549722
■ Manual [h]	5.526452228	7.619305873

Figure 2: The average automatic analysis time for PNO.d1 and PCC.1 tests.

The eDSL creates modules through an interface which was designed to be as close to natural language as possible (Figure 3). Keeping in mind that created tests are not very complex (PNO.d1 consists of 15 assertions and PCC.1 consists of 8 assertions), the usage of the automated analysis can have a significant impact on the future hardware commissioning campaigns.



```
public class MyModule extends AnalysisModule{
    {
        assertThat(I_MEAS)
            .isLessThan(55.0, AMPERE)
            .at(PM_EVENT_TRIGGER);

        assertThat(I_MEAS)
            .isEqualTo(EXPONENTIAL_FCT)
            .withinAbs(2.0, AMPERE)
            .starting(10, MILLI(SECOND)).after(PM_EVENT_TRIGGER)
            .ending(2, MINUTE).after(PM_EVENT_TRIGGER);
    }
}
```

Figure 3: An example of a simple automatic analysis module.

The automated analysis, compared to the manual approach, is much more scalable. Analysis tasks can be distributed among many different servers, depending on the needs, which can additionally shorten the total analysis time. High scalability is not possible in the case of the manual approach. Every additional person involved in the analysis process has to be properly trained to acquire the specific domain knowledge. Once a module is created it does not need to be recreated as in the case of new personnel requiring training.

Another advantage is that no human interactions are required while the automatic analysis is executing, and thus it is not subject to human errors. People performing repetitive manual tasks, such as manual result analysis, tend to become tired and are prone to make mistakes and misinterpretations. A properly developed and carefully tested module will always return exactly the same result for the same input data set. Modules can be run for the data sets from different commissioning campaigns, providing a good tool for comparisons between campaigns.

In the future new modules will be created which will allow the automation to support more complex test procedures. The eDSL functionalities will be expanded to support mathematical calculations to allow more complex and yet robust assertions to be created.

The previous arguments demonstrate the advantage of the automatic analysis over the manual approach especially when there are time constraints and the results of the analysis must be provided as soon as possible. To summarize, automated analysis provides clear improvements in terms of performance, but still the question of test procedure quality remains unanswered.

TESTING QUALITY

In order to gain trust in the effectiveness of the test procedures, it is necessary to find metrics for measuring the testing quality of past and future campaigns. These metrics should give an insight into how appropriate the procedures are for the situations that they are applied to.

There are several ways of expressing testing quality, one way would be to check the consistency of the result from previous campaigns which were performed manually against the results of the same campaigns using automated analysis. A separate consistency checking tool was created which checks the analysis accuracy of past campaigns against the thresholds set. Analysis consistency checks were performed on results from campaigns between 2011 and 2013. The consistency of past results were all over 90%. The biggest part of the inconsistencies (almost 90%) were caused by very strict criteria which did not involve such factors as high circuit resistance or signal noise. The rest of the inconsistencies were caused by data errors (6% - tests were passed manually), missing language functionalities (5%) and human errors (1%). The missing language functionalities have since been added to the eDSL.

While conducting these consistency checks the automatic analysis helped to identify one faulty circuit which had not been identified previously. The automatic analysis also confirmed three other failures which were previously known from the manual analysis.

Automatic Analysis has shown that it is an effective fault finding process, being able to identify problems which had previously gone unseen, as well as saving time and manpower. New language functionalities are already in place, or being worked on, which will enhance the capabilities of the automated analysis by satisfying more complex test scenarios. When manual analysis is performed the human is able to take into account extra criteria such as signal noise or high circuit resistance. To achieve this through automatic analysis some of the criteria have to be altered, doing this brings the consistency levels for previous campaigns up to over 99%.

The consistency between automatic and manual approaches is not the only possibility. Other approaches for testing quality would be to use the experience that we have gained from previous campaigns as a benchmark. One example of this is to create test data which simulate typical faults that have occurred previously such as the following:

- Noise Faults (Putting a random noise factor of x percent on one signal)
- Coherent noise (Put a random, but coherent noise on different signals)
- Spike Faults (Put a spike at a random location, exactly at the trigger position, or anywhere else that is necessary)
- Unusual decay behaviour (A decay which grows up instead of actually decaying, or a straight signal instead of an exponential decay)

This would build trust in the test procedures. An overall picture of the use cases which are covered by the test procedures could be built, for example if the test procedure can detect a typical fault on past campaigns then it could be assumed that similar faults in the future will also be detected. This concept could be extended, for every fault discovered it could be possible to build a new assertion which protects the system from this fault in future tests.

These concepts can be expressed as a number, the number of known faults which are covered by the testing procedures. This number becomes a metric and can derive the effectiveness of the testing procedures. This can reveal weaknesses in the testing procedures where the procedures need to be improved as well as what kind of faults the system is protected against already.

CONCLUSION

When working with such a large number of systems and their complex test procedures, as those found in the LHC magnet powering system, experience has shown that automation is an effective, and often necessary, step to achieve sufficient levels of reliability and safety. The AccTesting framework has already been in use in previous campaigns since 2011 and has proven its reliability [3]. The methods described in this paper show that new extensions of the AccTesting framework has had a positive impact on previous campaigns and future campaigns will benefit from the increased reliability and time saving.

By using past knowledge and experience that has been gained through previous test campaigns it is possible to create metrics which measure the quality of testing procedures. Correct metrics can guide the design of future testing procedures and planning of future hardware commissioning campaigns in order to provide safer machine operation.

Well defined and reliable test procedures can ease the hassle associated with hardware commissioning campaigns by allowing staff to focus on results which need their attention. As a result of the time saved, hardware commissioning could potentially be completed in a much shorter time period, allowing more time for upgrades and above all bringing more time for experiments in the accelerator. Hardware commissioning campaigns could be run more often than they have been in the past which would further increase the reliability and safety of all of the systems.

REFERENCES

[1] M. Audrain et al., “Using a Java Embedded DSL for LHC Test Analysis ”, THPPC079, proc. of ICALEPCS 2013, San Francisco, CA, USA.

[2] K. Fuchsberger et al., “Automated Execution and Tracking of the LHC Commissioning Tests”, THPPP009, proc. of IPAC 2012, New Orleans, LA, USA.

[3] D. Anderson et al., “The AccTesting Framework: An Extensible Framework for Accelerator Commissioning and Systematic Testing”, THPPC071, proc. of ICALEPCS 2013, San Francisco, CA, USA.