Using a Java Embedded Domain-Specific Language for LHC Test Analysis

M. Audrain, K. Fuchsberger, J.C. Garnier, R. Gorbonosov, A.A. Gorzawski, A. Jalal, J. Suchowski, P.C. Turcu, M. Zerlauth
(CERN, Geneva, Switzerland)

Analysis Script

```
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);
    }
}
```

Abstract

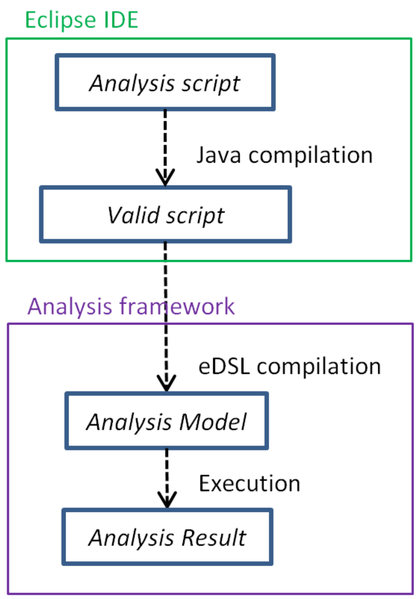
The Large Hadron Collider (LHC) at CERN requires thousands of systems to work in close cooperation. All these systems have to be tested during Commissioning and after interventions on them. Starting from the experience of Hardware Commissioning, the execution of such tests were already automated to a high degree. The remaining time in commissioning campaigns is now spent in analyzing test results, which is done manually to a certain extent. To improve this situation, a new project was launched which aims to automate the analysis of such tests as much as possible. For this purpose, a dedicated Java embedded Domain Specific Language (eDSL) was created which allows system experts to describe analysis steps in a simple way. The execution of these checks finally can produce, along with the simple decisions on the success of the tests, plots for the experts to quickly track down the source of problems exposed by the tests. This paper explains the concepts used and the future vision of this first version of the eDSL.

Requirements

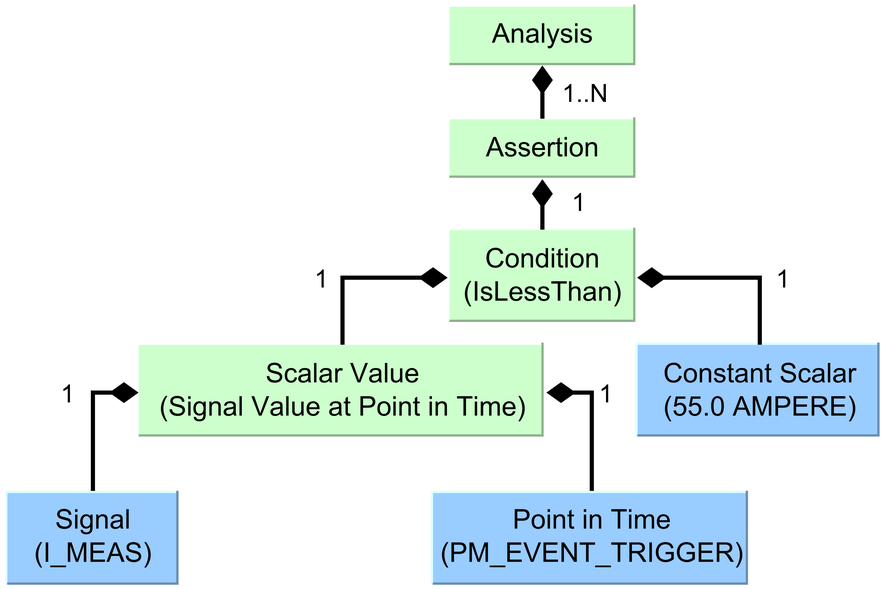
The main requirements for the Analysis framework are:

- The ability to define conditional checks called **Assertions**. These assertions have to ensure that a function of time is comparable to a value at a single moment or to another function of time over a time interval.
- The use of a graphical user interface (**GUI**) component to display the signals used in the assertions and the results of those assertions.

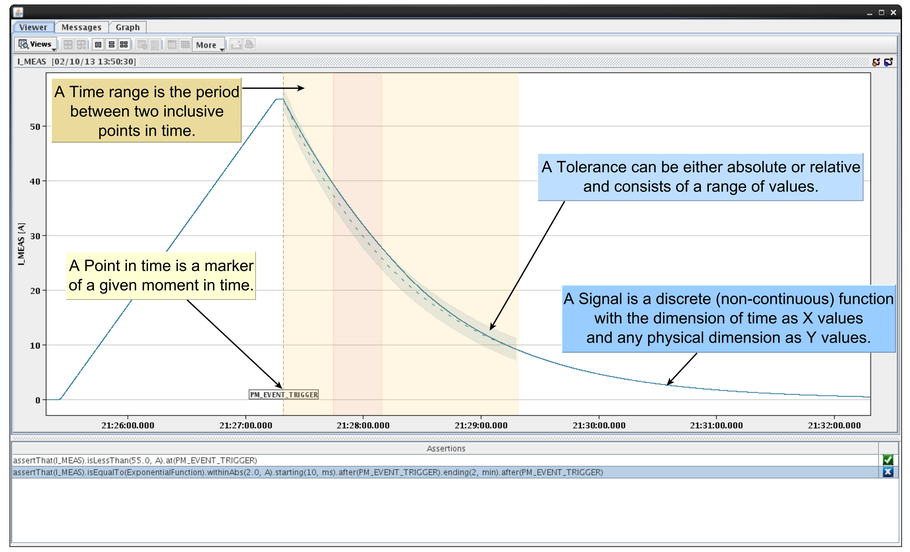
Language Processing Steps



Semantic Model



Graphical User Interface



eDSL Concepts

For the Fluent API of our analysis language, we chose the following concepts:

- **Method chaining** consists of having the current object modifiers returning the host object carrying the next element of the language syntax. Using a set of **Progressive interfaces**, one can enforce the syntax tree so the user can only use the mandatory elements provided all along the chain.
- **Expression builders** are used to decouple the fluent API building logic from the semantic model execution logic.
- The **Semantic model** can be seen as immutable domain objects populated by the language using the expression builders.
- **Object scoping** provides a template for the writing of the user analysis script. In the Java language, one can use the **Instance initializer** technique to achieve the template implementation.

Summary and Outlook

The analysis framework will be ready to perform the operational analysis for at least one prototype test during the next hardware commissioning campaign in 2014. Further analysis modules will need some more features, which are in the pipeline for 2015.

Future plans for the analysis language concern the implementation of calculation functionalities for more general analysis of operational data of the CERN accelerators. The analysis language could also be used directly in other operational software. Examples of this are: Post Mortem Modules, Sequencer Tasks or Pre/Post Operational checks of systems.