CERN Summer Student report

Tanya Angelova

Supervisor: Peter Chochula

Summer 2018

Contents

I Introduction **3**

II Acknowledgments **3**

III WinCC **4**

1 Architecture **4**

2 The data point concept **5**

IV My contribution in CERN **6**

3 What is the problem? **6**

4 What is the proposed solution? **6**

V Conclusion **9**

Part I
Introduction

This summer I was part of the Summer Student Programme in CERN. I had a chance to be selected to join CERN for 8 weeks and during this time I took part of the project ALICE (A Large Ion Collider Experiment), which is a heavy-ion detector on the Large Hadron Collider (LHC) ring.

This project is one of the largest experiments in CERN, dedicated to study the physics of strongly interacting matter at extreme energy densities. The goal of the experiment is to recreate and study the quark-gluon plasma, which is believed to be the state of the Universe up to a few milliseconds after the Big Bang. For that, the ALICE collaboration uses the 10,000-tonne ALICE detector – 26 m long, 16 m high, and 16 m wide.The detector sits in a vast cavern 56 m below ground close to the village of St Genis-Pouilly in France, receiving beams from the LHC.

In this report, I present the software, used for the development of my project and then the project itself.

Part II
Acknowledgments

To begin, I want to thank my supervisor Peter Chochula to be the best supervisor in CERN. He helped me a lot during all the program.

Also, I want to thank Peter Matthew Bond and Kevin Cifuentes, who gave me great advices and helped me finish my project. And, of course, one big "Thank you!" to all the team for being so amazing, thank you for the experience, for the coffee breaks, for the talks, for the last lunch and for the present! I truly felt lucky to have the support of the coolest team at CERN.

I'm grateful to Jennifer Dembski, Eszter Badinova and Despoina Driva for the work they did to organize the whole Summer Students program, and for the chance they offered me to be a part of it.

Finally, I'm grateful to all Summer and Openlab Students at CERN, from whom I've learned so much in the past weeks.

Part III
WinCC

1 Architecture

While working on my project, I used a software, called WINCCOA. It's a SCADA system, standing for Supervisory Control And Data Acquisition. It's a software, used to connect hardware (or software) devices, acquire the data they produce and use it for their supervision. In order to do this, the software is composed of a number of Managers:

- The Event Manager (EVM) - responsible for all commuications
- The DataBase Manager (DBM) - provides the interface the the data base
- User Interface Managers (UIM) - can get device data from the database or send data to the database. It can also be run in a developement mode: PARA for parametrization of DPs (datapoints) and GEDI for the Graphical Editor
- Drivers(D) - provide the interface to the devices to be controlled
- Archive Managers - alows users to archive data
- etc.

In order to manage them, there is a software console, from where the user can reach any tool (Figure 1).

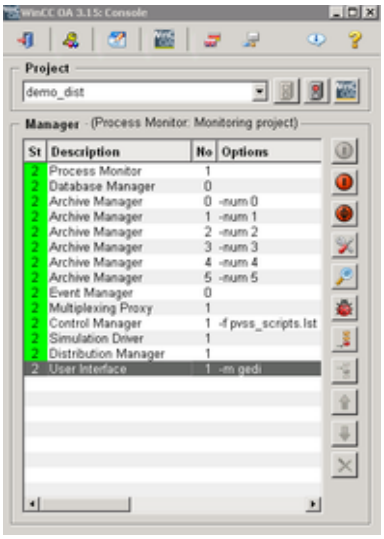


Figure 1: Console

2 The data point concept

The collected data is in the database and it is structured as Data Points(DP) of a pre-defined Data Point Type(DPT). WinCC allows devices to be modelled using these data points. The Data Point Type describes the data structure of the device and a data point contains the information. An organized datapoint type list is shown on Figure 2.

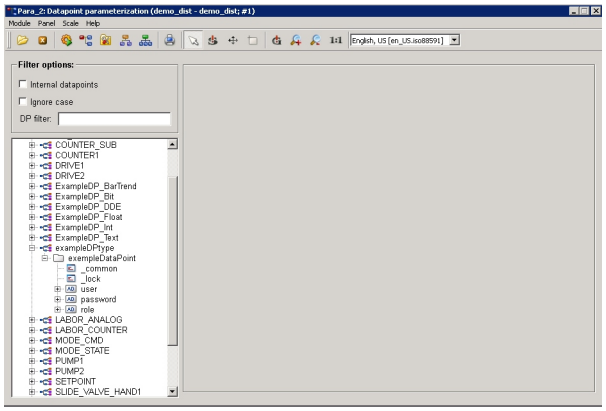


Figure 2: Datapoint parametrization

In this particular example the data point type is defined as "exampleDPtype" and the data point itself as "exampleDataPoint". We can see two types of sub-element here. The first type includes "_common" and "_lock"- these are known as "configs". There is a wide variety of configs like alert_hdl, archive, address, original ,etc. and they enable the specific behavior to be configured for the DPE (for example alarm handling or archiving) as well as to hold the attributes of that DPE - its current value or its hardware address(peripheral address config). The second type includes "user","password" and "role" - these are the parameters of the DPE and every one of them has its own configs. In this case the datapoints are only of type string, but several more data types are available - boolean, float and the dynamic arrays of the simple data types like dyn_int, dyn_float, dyn_string, etc. These dynamic arrays are used to hold a variable number of elements of the same type so the user doesn't have to give the exact number of elements from the beginning.

To access the data, hold by an element of the data point "exampleDataPoint", the user can address it as: exampleDataPoint.user or exampleDataPoint.password - the addressing scheme is always the same and it can be with a lot more levels than that.

For creating and modifying the DPTs and DPs I used the Graphical Parametrization tool (PARA), as well as programming scripts.

Part IV
My contribution in CERN

3 What is the problem?

Like we saw in the above section, the concept of a datapoint can be really wide and even complicated. Every data point could have a lot of configs and a lot of elements, every one of them of different type with some configurations themselves. So in a bigger organisation with numerous levels, it would be difficult to find the datapoint needed or a list of datapoints with a specific value of one of its configurations. The ALICE control system is configured as a large distributed system running on 100 servers. About 1 000 000 parameters are scattered across these computers. So this big distributed system needed a tool for the central system, where it can gather the required data and execute a necessary action. There have been already an idea and an old form, but the implementation wasn't functioning, it was hard to use and it had only basic functions. The purpose of this project was to enhance existing tools by configurable interface that will be used to check certain parameters.

4 What is the proposed solution?

My internship started with examining some old already partly implemented forms (Figure 3) and by starting to build a new basic one myself (Figure 4).

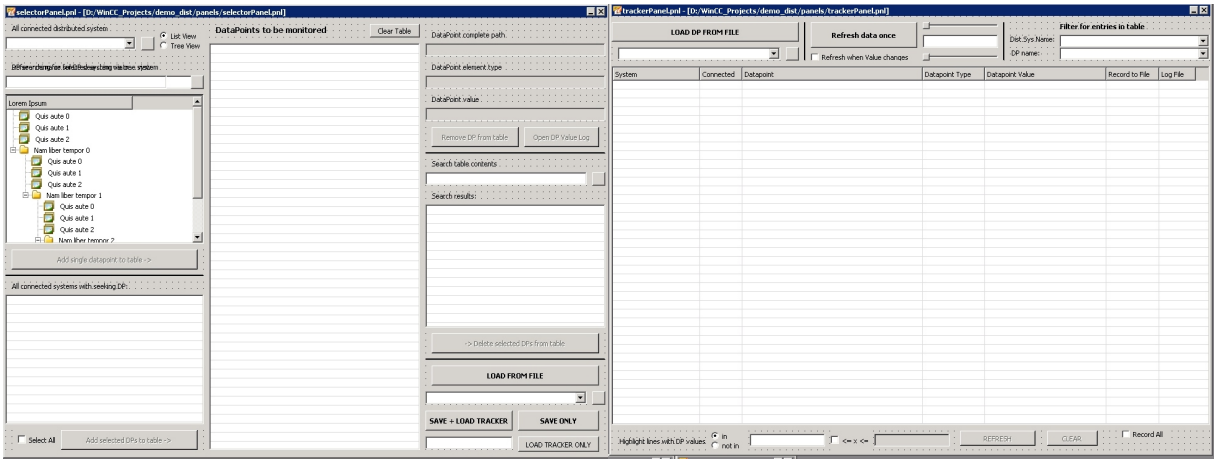


Figure 3: Old forms

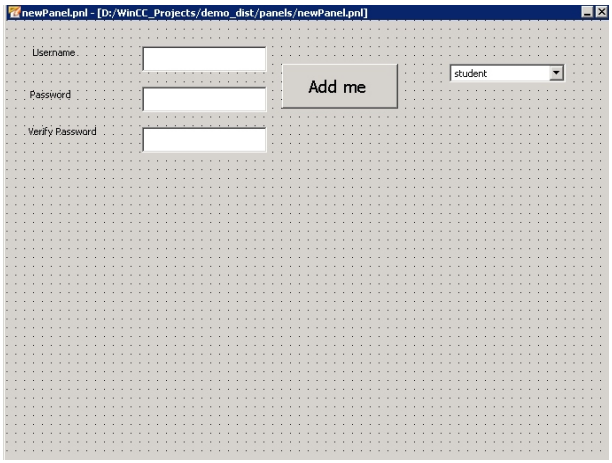
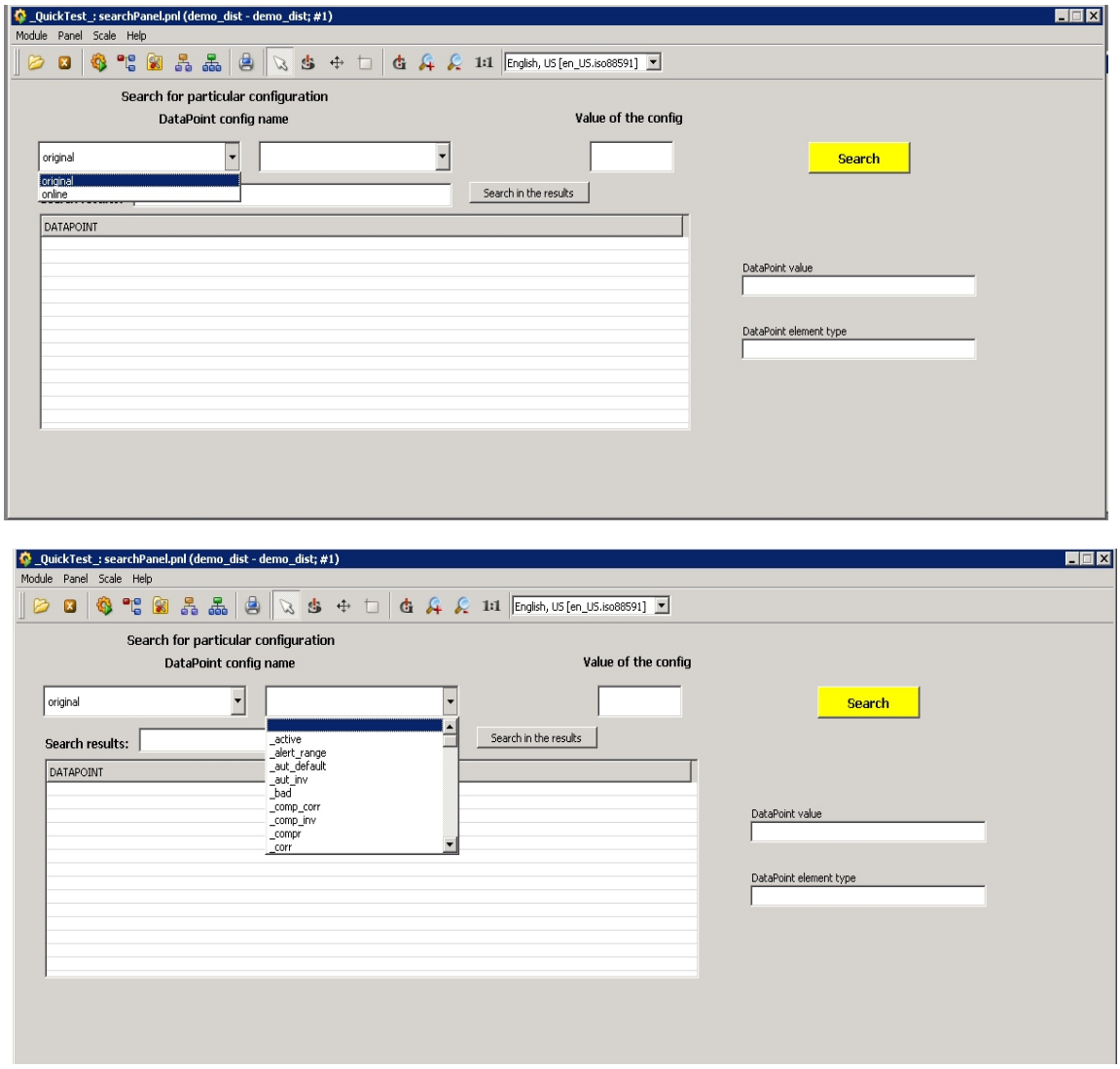


Figure 4: My own first form

Inspired by these old forms that were already developed, we decided to create an application for searching through the datapoints of all connected systems by config's name and its desired value. During the implementation of the idea, we realized that there are many different configs with different structures and it would be difficult to construct one generalized searching system for all of them. That's why we decided to make a list of most frequently used configs, organized in two drop-down lists. The user has the opportunity to choose between "original" and "online" value of a long list of configs. The next field is dedicated to the particular value desired. For example ,if we want to find all datapoints in the system, which online invalid value is 1, we choose "online" from the first drop-box menu, "_invalid" from the second and we write the value 1 in the field, called "Value of the config". We receive a list of names of all datapoints, satisfying the imposed requirement, including in which distributed system is it. From this list, we can choose the datapoint needed. On the right we have two fields, indicating the value and the element type of the marked datapoint.



(a) Choosing online or original value and the config

Figure 5: Choosing the config's name

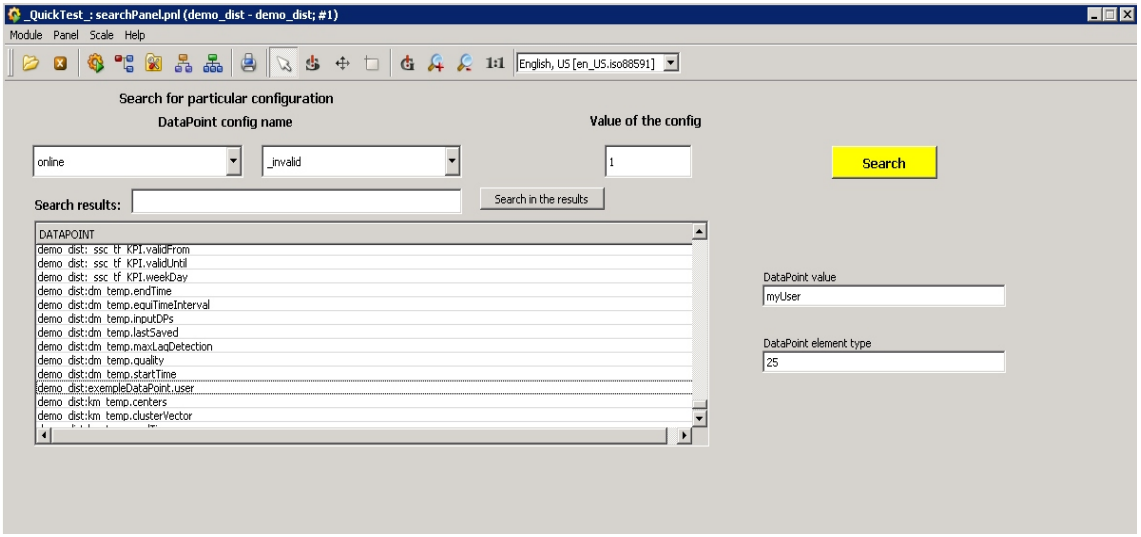


Figure 6: Visualisation of the results

After running some tests, I saw that the list can be really long and this will make the searching of a particular datapoint or a group of datapoints complicated. So we decided to add one field more, where we can search through the results, which was indeed very useful in the end and enabled easier manipulation.

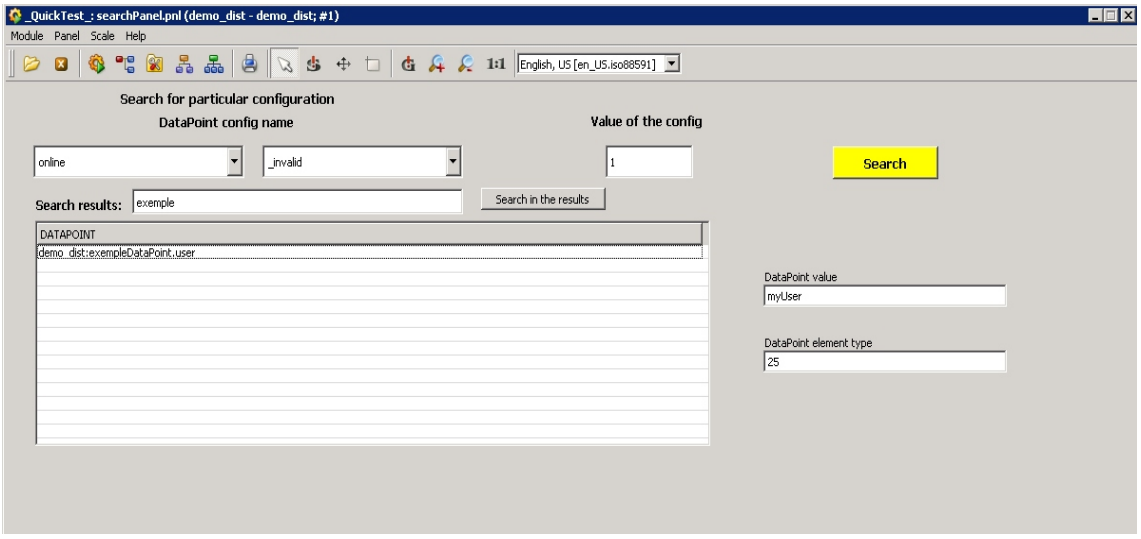


Figure 7: Searching throught the results

Part V
Conclusion

In conclusion, I can say that this project was certanly very interesting and really rewarding. Thanks to my great team, I learned a lot about CERN, ALICE project and WINCCOA. The implemented application and the idea could be further developed. It's possible to add more configurations, with different constructions to improve the search, also it's possible to add more characteristics of the selected datapoint. In addition the application needs some ways to apply changes to the datapoint's components and to perform an action with the datapoint (adding to another application, deleting from the system, etc.). The whole summer student experience was one-in-a-lifetime opportunity to work in an amazing environment, to meet interesting people and to get to know the beautiful region of Geneva. Thank you for choosing me to be part of CERN's world.