Improvements to the ATLAS Expert System

Eva Lott, CERN Summer Student (EP-ADE-TK)

August 2019

Abstract

The ATLAS Technical Coordination Expert System is a diagnostic tool that aims to increase users knowledge base of the detector in the case of interventions by providing easy access to the list of affected systems in the event of a particular intervention. It describes the behaviour of elements of the experiment infrastructure such as gas, cooling, ventilation, electricity distribution, and the detector safety system. This document reports the improvements made to the expert system in order to use an XML representation of the graphical designs, based on XML graphs. The resulting implementation is easier to use, and enlarges the prospects for new functionality of the tool. Additionally, the description of the RPC was revised, and the missing elements added to the database.

1 Introduction

The ATLAS expert system [1] has broad functionality. Not only does it describe individual systems, for example the RPC (resistive plate chambers) sub-detector, but it also describes the links between the individual systems within the ATLAS Experiment [2]. This allows a user to examine what will happen to individual infrastructure, such as a specific chamber within the RPC when an electronics rack is switched off. Diagnostically this is incredibly useful as if several elements fail across the system, it is possible to understand in little time how the large dependency tree spanning the entire cavern behaves, increasing efficiency in finding a the specific cause of a fault in operation. The system also provides written description and images of infrastructure elements, making it a good substitute for verbose and often deprecated documentation.

The tasks described in this report consist of:

- Integrating a vector graphic system such that graphical representation of the dependencies between objects in the view pages can be represented at arbitrary resolution and edited implicitly, with page elements displayed over the graph dynamically changing position.
- Describing the RPC sub-detector within the expert system.

2 System Operation

The expert system has a web-based user interface, a server relying on a large `PHP` and `Python` codebase, and an object oriented database to connect elements using different relationships.

All infrastructure is stored in an XML database specific to ATLAS, known as the OKS [3]. Each object has a UID (unique identifier) and is linked to other objects it is dependent on and that depend on it, in tree-like structure. The data is inserted into the database from various older documentation files or other sources of information by a series of `Python` utilities.

The data can be searched through in a user friendly method directly using the expert system. It is displayed in custom graph tables as shown in figure 1 (left). The vertices display a switch which allows for the changing of the state of an object, a display square which shows if the given object is still functional in the session state of the simulation, and an index link which provides more information about the given object, all shown in figure 1 (middle). A menu organized by category allows navigation through the different pages organized by system, sub-detector, or location, as shown in figure 1 (right).

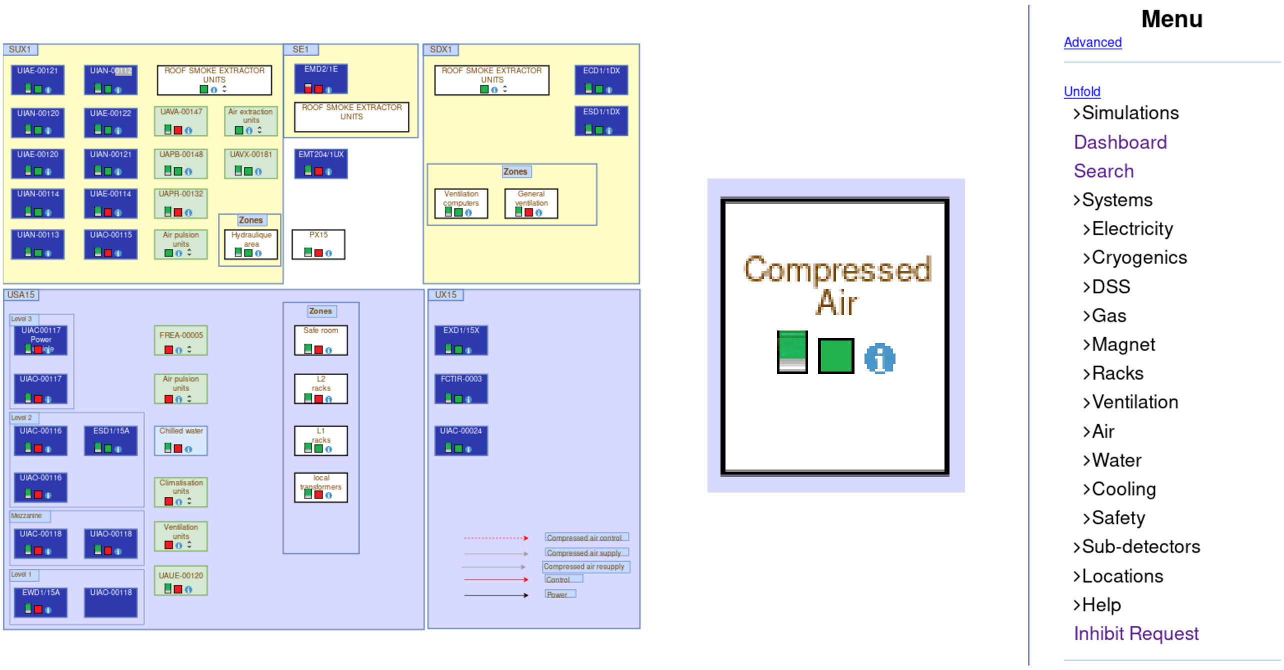

Figure 1: (Left) The ventilation page, (center) an object display, (right) categories described by the system (without the many subcategories).

3 Implementation of the `MXgraph` library

The `MXgraph` library [4] has been chosen for implementation of vector graphics inside the expert system. It is open source (using the Apache 2.0 licence), customizable, light weight, and has no dependencies. The library allows easy client side generation of vertices and edges and allows dynamic connection between them. The `draw.io` editor [5] is used to edit the vector image. The editor can not only customize the visual elements of the graph, but also has the capability to embed data into any part of the graph. This is used to encapsulate page information and elements to be received client side. For example the UID and position of an individual vertex, or alarm systems for the graph itself, as shown in figure 2.

Figure 2: (Left) Relative position and UID code can be described, (right) and data for the whole page, such as environmental alarms can be described.

The `MXgraph` from the editor is stored in XML format. The payload inside the XML is URL encoded, compressed with zlib without the header and then base-64 encoded. The client side receives this XML via an XHTTP request, then deflates and renders the diagram data using `Javascript`. The data in the page is extracted when rendering, so interactive elements such as switches, page alarms, and object types can be rendered correctly. An example of a properly rendered page using `MXgraph` is shown in figure 3.

Back-end functionality also required the ability to parse the XML server side, so a small general python wrapper class is used to allow the back-end system to store data elements from the encoded diagrams for `PHP` operation.

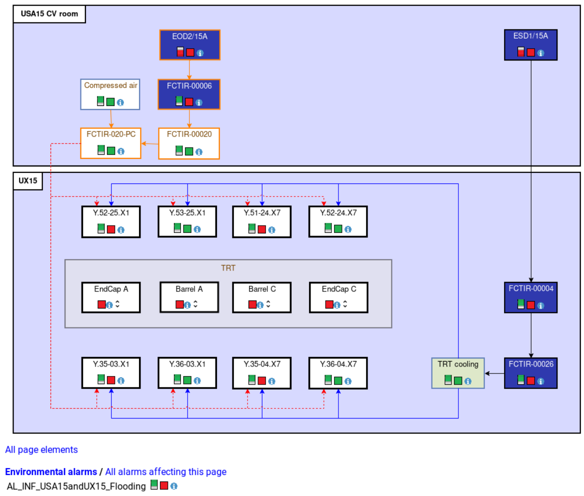

Figure 3: The page is rendered with UIDs over their vertices

4 Description of the RPC

Redesign of the RPC page required gathering all the information concerning the infrastructure of the RPC. This was done in a two stage approach: first by collecting the information available on the ATLAS DCS RPC pages; and second by meeting with the RPC expert.

The information from the RPC experts was provided in the form of generated text files that contained the list of racks, crates, boards, and channels that instrument every chamber of the RPC. The RPC is composed of the following:

- 16 sectors, 8 octants, 2 sides (A and C) and 2 layers (BML, BOL)
- Phi segmentation: 4 quadrants, 4 DCS PCs, 4 CAEN mainframe, 4 generators in US15
- LPVS (48V): 24 PS with 2 channels, 3 PS per octant, in UX15
- Z segmentation: 3 barrels, 3 HV lines
- HVPS (10kV): 51 PS with 6 channels, in UX15
- 5 Gas racks: C2H2F4 (94.7%), C4H10 (5%), SF6 (0.3%)

This information was cross referenced, processed into `JSON` maps containing the relationships between crates and chambers, and crates and branches, then inserted into the expert system. The number of objects required to describe the RPC are: 500 chambers (SubDetector), 100 crates, 20 racks. Typically, one rack contains 5 crates, and one crate powers 20 chambers.

The naming convention for the RPC chambers is the following. Given a chamber name (BIL4A01): The first letter corresponds to the region (B for barrel, E for end-cap). The second letter corresponds to the layer (I for Inner, M for Middle, O for Outer, E for Extra layer in the barrel to end-cap transition). The third letter corresponds to the station type (S for small, L for large, F for those between the feet, G for those inside the foot, M for those close to the calorimeter rails, R for those below the calorimeter rails, and E for those mounted on the elevator shafts). Next digit corresponds to the index from 0 to 8 (counted along Z for the barrel starting at Z=0, and radially from the beam-pipe for the end-cap). The next letter corresponds to the detector side (A for Z>0, B for Z=0, C for Z<0). Finally, the last two digits correspond to the sector number (1 to 16). With the data mapped by the system it is far easier to have an understanding of the sprawling dependencies, and build a geometric picture of the RPC system in regards to a dependency graph, which in turn will improve control and understanding of conditions within the cavern.

5 Conclusions

The expert system has been demonstrated to be a crucial tool in examining the extensive infrastructure of ATLAS in a digestible way.

A new functionality of the expert system is to be able to use the `MXgraph` XML representation for the images. This simplifies the process of creating a page, and opens a door to new functionality in the back-end of the system. It is possible to implement a complete custom graphics editor that would read the database contents.

The description of the RPC embodies the importance of having an expert system for ATLAS. Information gathering is one of the more complicated and time consuming tasks, congregating data into an accessible system increases efficiency for the many who will require retrieval in future. The RPC description has been incorporated into the expert system up to the level of a single chamber, with more than 1000 new elements in the database. This results in the increase of the knowledge base of ATLAS, meeting one of the goals of the expert system.

6 Acknowledgements

The author wishes to thank Mr. Ignacio ASENSI TORTAJADA and Dr. Carlos SOLANS SANCHEZ for the opportunity of participating in this internship.

References

[1] I. Asensi Tortajada, C. Solans, A. Rummler, G. Salukvadze, and K. Reeves, "ATLAS Technical Coordination Expert System," Tech. Rep. ATL-SOFT-PROC-2018-028, CERN, Geneva, Nov 2018.

[2] The ATLAS Collaboration, "The ATLAS experiment at the CERN large hadron collider," *Journal of Instrumentation*, vol. 3, pp. S08003–S08003, aug 2008.

[3] R. Jones, L. Mapelli, Y. Ryabov, and I. Soloviev, "The OKS persistent in-memory object manager," *IEEE Trans. Nucl. Sci.*, vol. 45, pp. 1958–1964, 1998.

[4] "MXGraph library version 4." `https://github.com/jgraph/mxgraph`. Accessed on 2019-08-27.

[5] "draw.io editor." `https://www.draw.io`. Accessed on 2019-08-27.