EPJ Web of Conferences **251**, 04003 (2021)
CHEP 2021
https://doi.org/10.1051/epjconf/202125104003

Understanding ATLAS infrastructure behaviour with an Expert System

Ignacio Asensi[1,*], *André* Rummler[2], *Carlos* Solans[2], *Gustavo Andres* Uribe[3], and *Jose* Torres[1]

[1]Universidad de Valencia
[2]CERN
[3]Universidad Antonio Narino (CO)

Abstract. The ATLAS detector requires a huge infrastructure consisting of numerous interconnected systems forming a complex mesh which undergoes constant maintenance and upgrades. The ATLAS Technical Coordination Expert System provides, by the means of a user interface, a quick and deep understanding of the infrastructure, which helps to plan interventions by foreseeing unexpected consequences, and to understand complex events when time is crucial in the ATLAS control room. It is an object-oriented expert system based on the knowledge composed of inference rules and information from diverse domains such as detector control and safety systems, gas, water, cooling, ventilation, cryogenics, and electricity distribution.
This paper discusses the latest developments in the inference engine and the implementation of the most probable cause algorithm based on them. One example from the annual maintenance of the 15°C water circuit chillers is discussed.

1 Introduction

The ATLAS [1] Expert System [3] is a simulating and diagnostic object oriented expert system, created by ATLAS Technical Coordination to increase the knowledge base of the ATLAS experiment infrastructure, to allow easier transfer of knowledge between experts for specific domains and to help in the preparation of interventions. It describes parts of the experiment infrastructure like gas, cooling, cryogenics, ventilation, electricity distribution, and detector safety systems. It also represents their interactions with sub-detector systems (inner detector, calorimeters, Muon and magnets) resulting in a complex mesh of entities which are connected by various types of relationships. It is designed for being also used by non-experts to learn about the infrastructure and to plan interventions evaluating the possible impact on other systems of the experiment. Its requirements can be found in [3].

The ATLAS Expert System has become an important tool in Technical Coordination as part of the standard procedure conducted prior to interventions and it is used in the ATLAS Control Room in many situations [4]. The knowledge base is constantly updated tracking changes in the infrastructure during the phase-I upgrades of the ATLAS detector like the replacement of the Small Wheel of the Muon sub-detector by the New Small Wheel [2]. The

[*]e-mail: ignacio.asensi@cern.ch

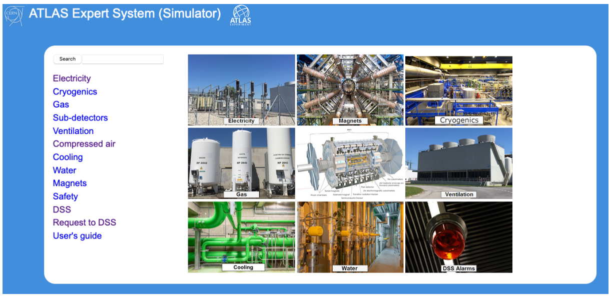


Figure 1. Expert System welcome page

update process is carried out manually following new documentation and being supervised by experts of each area.

Additionally, it can help diagnosing the cause of an unexpected situation. This could be a situation where alarms have been triggered due to the power loss of a specific system and the reason is not immediately apparent. In this case, the Expert System is an excellent tool to understand what could have caused that situation.

1.1 User interface description

The ATLAS Expert System user interface consists of a web based application which contains a virtual representation of the ATLAS experiment. The welcome page depicted in Fig. 1 contains links to general descriptions of systems such as electricity and cryogenics.

The system structure is presented to the user in form of flow chart like diagrams similar to those used in SCADA systems. Fig. 2 shows on the right a partial view of the electrical distribution page. The upper blue bar contains from left to right the menu icon, the page's title, the number of triggered alarms and affected systems, the language options, the button to reset the simulation, the search box, and simulation time constraints. Below the blue bar there are two areas which follow the convention of a yellow background for surface buildings and a blue one for underground caverns. On the left of Fig. 2 there are two boxes representing an individual system (top) and a group of systems (bottom).

Individual systems have usually only one state and accept as user input the command to set it to enabled or disabled. This is realized by toggling the left switch in their boxes. Once an action is taken, the simulation is executed and the user is immediately presented with the consequences. Information can be also accessed through the search functionality and list-oriented interfaces which provide more detailed information about the systems and their relations.

2 Architecture updates

The Expert System has three separate components from a technical perspective as explained in [4]: the database and the python server in the back-end and the web application in the front-end.

The database used is the *ATLAS TDAQ object oriented configuration database* also called *Object Kernel Support* (OKS) [5]. This software component is expected to stay maintained during the life time of the experiment. To simulate the ATLAS infrastructure many categories of differently behaving entities are required such as racks, computers, alarms, etc. These

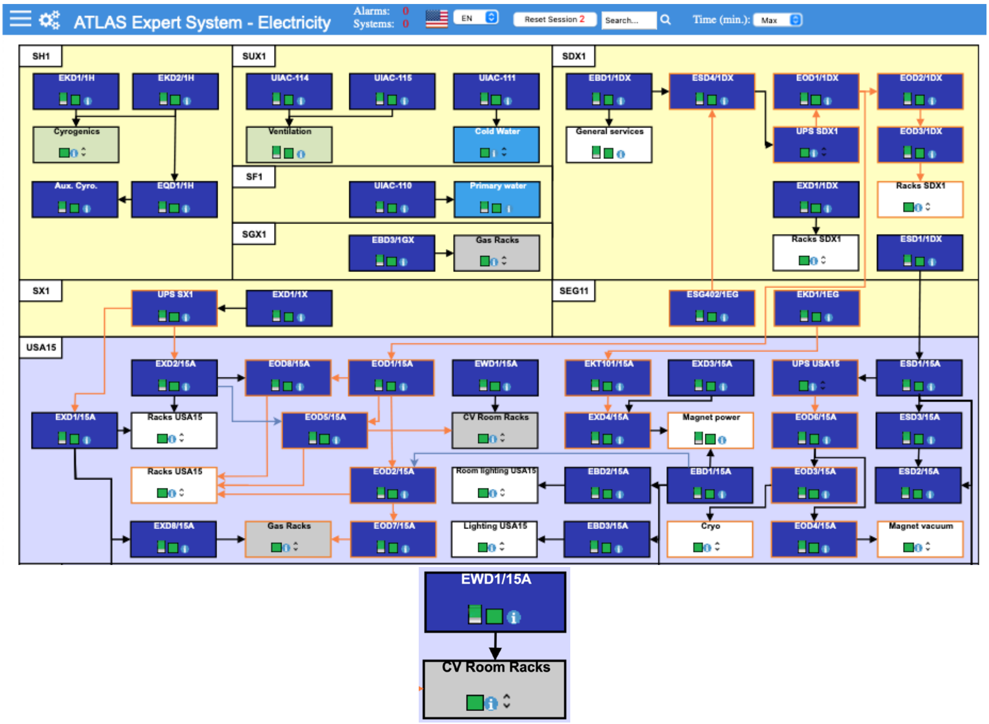


Figure 2. Top: Expert System simulation interface, partial view of electrical distribution. Bottom: example of a switchboard (upper box) and of a group of racks (lower box)

categories are defined in the database as classes and each object instantiates a class. A python server runs the inference logic with information from the database: it loads elements with their relationships and interacts with the web application receiving user input and providing scenarios as answers.

The front-end user interface is a web application built in PHP and standard JavaScript. Diagrams are built using a dynamic in-system diagram builder framework based on the MX-Graph [6] library. The python server, the web server and the client side communicate with each other using widely-used technologies like JSON, asynchronous JavaScript communication, and HTTP. Recent changes to the server architecture include the usage of the Python Networkx [8] graphs and networks library which is used for processing the mesh of systems in the database as a graph composed of nodes and edges. This allows the usage of state of the art algorithms for traversing the graph, detecting cycles, selecting parents, calculating the centrality, and detecting isolated components.

2.1 Knowledge representation and inference Engine

Each type of object, which is stored in the database within a database class, has different types of relationships. When a simulation is triggered, the server calculates each object's state based on the state of its parent objects.

For example physical racks are described as objects using the class *Rack*. These objects can be related to the following nodes: power sources (*poweredBy*), computers contained inside the rack (*contains*), local batteries time span (*lifespan*), interlocks (*interlockedBy*) or

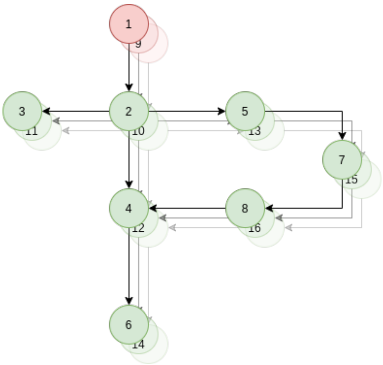


Figure 3. Depending objects breadth-first propagation order

cooling requirements (*waterFrom*). Each one of these nodes is solved applying a set of rules. The individual resolution of every node contributes to the final state of the object.

A graph representation is constructed with all the objects in the database using the Networkx *MultiDiGraph* object where the edges correspond to the relationships. When a simulation is triggered by the change of state of any element, that change is propagated over the dependent objects traversing the graph using the breadth-first algorithm as shown in Fig. 3. Due to circular dependencies between some of the dependent objects, the process is repeated as many times as required for achieving a stable state. After the simulation has stabilized it is assumed that the final state has been reached and the new state is sent to the user interface.

3 Simulation example: annual water maintenance

Annual interventions are mandatory for the infrastructure maintenance and safety. Although they are carried out every year, errors may occur due to oversights in routine procedures. Before interventions the Expert System is used to evaluate the consequences which allows to take counter measures to minimize the impact. Information about the behaviour of the system is collected during all interventions and serves to improve procedures and the accuracy of the Expert System knowledge base.

The annual maintenance of chilled water requires the stop of the 2500 kW chillers located in SUX1 that take water from the primary circuit and chill the secondary circuit down to 15°C. This mixed water is circulated down to USA15 with the help of two 22 kW pumps where it is used by the heat exchanger of the rack cooling circuit as primary cooling for the racks. It is also pumped again from USA15 to the UX15 cavern where it is used as primary cooling for four cooling stations (Muon A, Muon C, Tile, LAr). The mixed water is also pumped to SDX1 for the rack cooling.

Fig. 4 shows the water distribution page while simulating the maintenance. In the blue bar at the top it is shown that there are 41 alarms and that 5288 systems are affected in total. The two black squares on the right, FUPF1-00200 and FUPF1-00201, are expanded groups displaying the two intervened (switched off) systems, HAA-1411 and HAA-1401. Each of them is a TRANE CVGE050 centrifugal compressor that uses 625 kg of R-134a as a refrigerant for an output cooling power of 2.5 MW. On this page one can already see the impact of the intervention on rack cooling in SDX1, USA15, Muon A and C as well as LAr and Tile cooling. Rack cooling groups include many racks which are critical for monitoring and control of the detector.

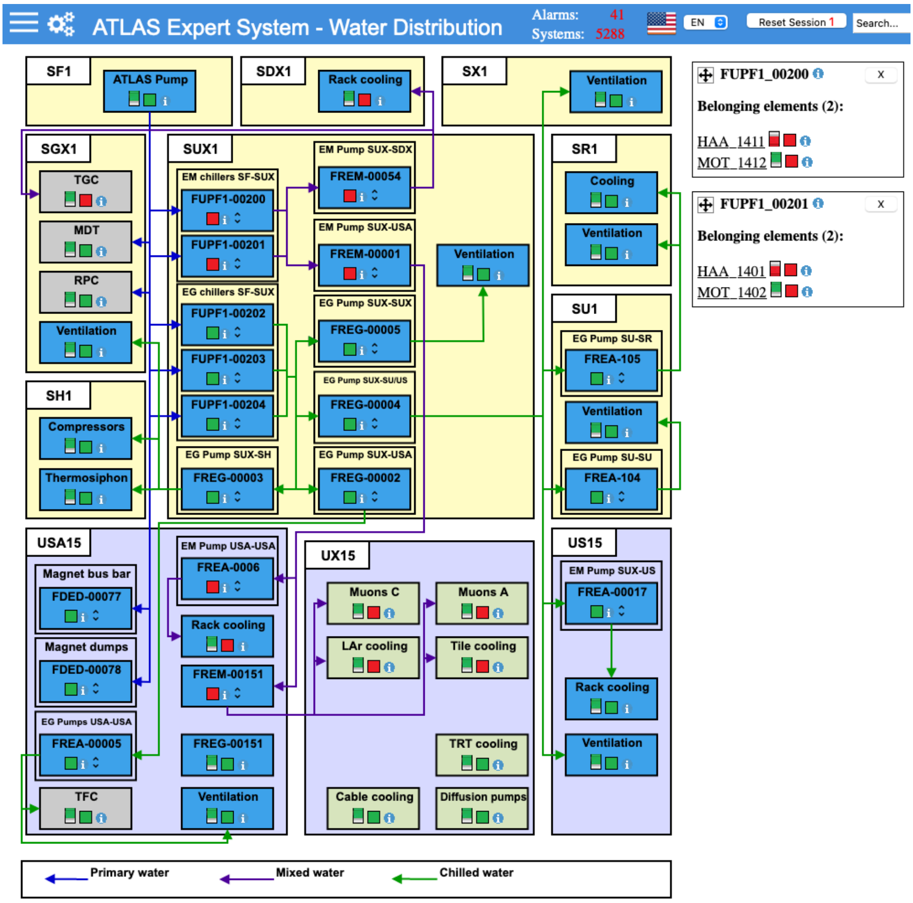


Figure 4. View of Expert System water distribution panel during simulation of annual maintenance of chilled water

This simulation is a clear example of an intervention whose impact is usually underestimated.

4 Tools

Expert System simulations allow to investigate the consequences of user proposed changes to the infrastructure. Furthermore, they allow to investigate the potential root cause for a scenario entered by the user. One can enter e.g. a list of alarms and search for the Most Probable Cause (MPC). A tree representation of the database enables to make a risk analysis using a Fault Tree approach.

4.1 Most Probable Cause

The Expert System can search the MPC for a user provided scenario traversing the graph representing the dependencies in a reversed direction. The scenario is provided to the MPC algorithm as a list of elements. In this context, objects that can cause a change of state of

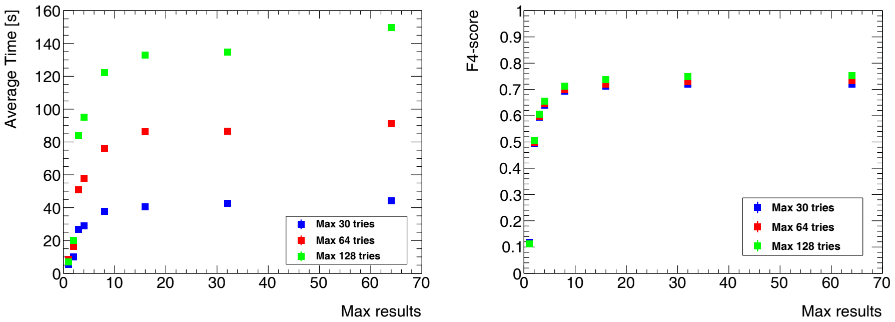


Figure 5. MPC algorithm performance. Left: Average processing time vs maximum number of results. Right: F_4-score vs maximum number of results

other objects are named parents. The cause is calculated in an exhaustive manner searching for the common parents of all the elements in the list. These parents are then filtered for those that affect all elements on the list but only those on the list.

The common parents are selected running the breadth-first algorithm starting from each object provided by the user and filtering the parents which are present in all the results. Furthermore, the MPC can be executed in a non-exhaustive way filtering the parents that affect all but not exclusively the elements listed by the user. It is intended for the users who might not have the full picture of the affected systems. The MPC algorithm demonstrates the best performance if before the processing the objects are ordered by the score of their contribution. The score is given by eigenvector centrality, a measure of the influence of a node in a network.

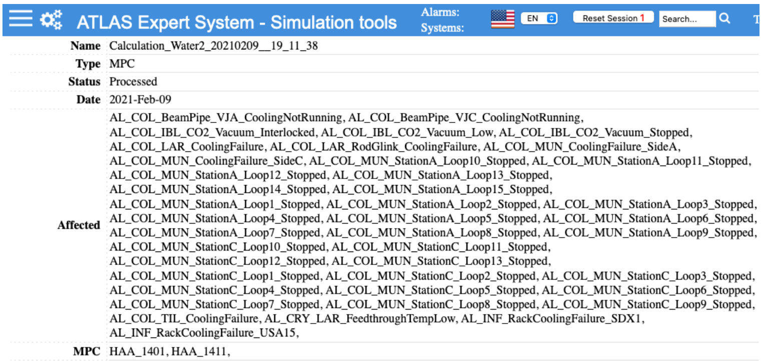
The MPC algorithm uses two parameters. First, the maximum number of attempts which is the number of parents that will be processed. Second, the number of results shown to the user. Fig. 5 reveals on the right of the plot a stabilization of the F_4-score showing a value of 0.7 after 8 results. The F_4-score is a measure for the quality of the results.

The F_4-score is calculated by Eq 1 where β is equal to 4. Precision is the number of correctly identified positive results divided by the number of all positive results including those not identified correctly. The recall is the number of correctly identified positive results divided by the number of all samples that should have been identified as positive.

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{1}$$

The left part of the plot in Fig. 5 indicates that the number of attempts does not strongly affect the maximum results and consequently also not the quality of the results. Therefore, increasing the maximum number of results would increase the processing time without significantly improving the quality of the results. A number of 8 maximum results and 30 tries has been established as the best parameter set for the algorithm in terms of time vs accuracy with an average time of 37 s and a F_4-score of 0.7.

Fig. 6 shows the MPC tool output after entering the list of 41 alarms which were triggered during the annual maintenance of the chilled water production system. The result is calculated to be HAA-1411 and HAA-1401 which correctly reflects the real root cause. The process is more time consuming compared with the normal simulation, around 10 minutes compared to typically a few seconds. It is a huge improvement compared to the brute force approach of the Fault Tree Analysis (FTA, explained in the following paragraph) which requires several



Name	Calculation_Water2_20210209__19_11_38
Type	MPC
Status	Processed
Date	2021-Feb-09
Affected	AL_COL_BeamPipe_VJA_CoolingNotRunning, AL_COL_BeamPipe_VJC_CoolingNotRunning, AL_COL_IBL_CO2_Vacuum_Interlocked, AL_COL_IBL_CO2_Vacuum_Low, AL_COL_IBL_CO2_Vacuum_Stopped, AL_COL_LAR_CoolingFailure, AL_COL_LAR_RodGlink_CoolingFailure, AL_COL_MUN_CoolingFailure_SideA, AL_COL_MUN_CoolingFailure_SideC, AL_COL_MUN_StationA_Loop10_Stopped, AL_COL_MUN_StationA_Loop11_Stopped, AL_COL_MUN_StationA_Loop12_Stopped, AL_COL_MUN_StationA_Loop13_Stopped, AL_COL_MUN_StationA_Loop14_Stopped, AL_COL_MUN_StationA_Loop15_Stopped, AL_COL_MUN_StationA_Loop1_Stopped, AL_COL_MUN_StationA_Loop2_Stopped, AL_COL_MUN_StationA_Loop3_Stopped, AL_COL_MUN_StationA_Loop4_Stopped, AL_COL_MUN_StationA_Loop5_Stopped, AL_COL_MUN_StationA_Loop6_Stopped, AL_COL_MUN_StationA_Loop7_Stopped, AL_COL_MUN_StationA_Loop8_Stopped, AL_COL_MUN_StationA_Loop9_Stopped, AL_COL_MUN_StationC_Loop10_Stopped, AL_COL_MUN_StationC_Loop11_Stopped, AL_COL_MUN_StationC_Loop12_Stopped, AL_COL_MUN_StationC_Loop13_Stopped, AL_COL_MUN_StationC_Loop1_Stopped, AL_COL_MUN_StationC_Loop2_Stopped, AL_COL_MUN_StationC_Loop3_Stopped, AL_COL_MUN_StationC_Loop4_Stopped, AL_COL_MUN_StationC_Loop5_Stopped, AL_COL_MUN_StationC_Loop6_Stopped, AL_COL_MUN_StationC_Loop7_Stopped, AL_COL_MUN_StationC_Loop8_Stopped, AL_COL_MUN_StationC_Loop9_Stopped, AL_COL_TIL_CoolingFailure, AL_CRY_LAR_FeedthroughTempLow, AL_INF_RackCoolingFailure_SDX1, AL_INF_RackCoolingFailure_USA15,
MPC	HAA_1401, HAA_1411,

Figure 6. Screenshot of the MPC result for a potential scenario occurring during the annual maintenance of the chilled water circuit

days of computation time for the entire database. The results are stable and reproducible. The state of other elements of the database does not influence the speed of the algorithm, and the same result is obtained each time. In order to speed up the simulation further, the result could be stored in a pre-computed cache of expected scenarios, that could be presented to the user with very little latency. This option is being explored at the moment.

The search for the MPC can be used to understand many situations in the control room and by safety system experts in the early steps of critical situations, when time is essential and the cause of a failure is not well understood.

4.2 Fault Tree Analysis

The FTA can estimate the probability of failure of every object [3]. The principal components of a given system can be deduced by reducing the reliability of a different node each time. This algorithm consists mostly of calculating the probability of failure for each element in the tree and of sorting the results afterwards in a meaningful way. It either does not take into account the coincidence of 2 elements. The algorithm is computationally intensive. Although this is a problem that can be solved with the computing power available today.

5 Conclusions

The ATLAS Technical Coordination Expert System describes and simulates the most relevant parts of the ATLAS infrastructure. It has been proved that it describes the detector's behaviour with a high degree of accuracy as it has been used extensively for preparing and evaluating the impact of interventions during LS2. Furthermore, the Expert System has been updated with the information collected during maintenance interventions and integrating upgrades of the infrastructure during LS2. The recent development of the Most Probable Cause tool extends the area of application of the Expert System to the analysis of ongoing events. The probable cause of failure is deducted quickly and with high reliability.

References

[1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST, 2008. https://cds.cern.ch/record/1129811

[2] ATLAS Muon Collaboration, *New Small Wheel Technical Design Report*, 2013, https://cds.cern.ch/record/1552862

[3] I Asensi et al, EPJ Web Conf 214 (2019) 05035 https://doi.org/10.1051/epjconf/201921405035

[4] I Asensi et al, JACoW Publishing Proc ICALEPCS 2019 (2020) 106 - 109

[5] R Jones et al, TNS 45 (1998) 1958-1964, https://cds.cern.ch/record/370902

[6] MXGraph, 2020 [software] version 4.1.0 Available from https://github.com/jgraph/mxgraph

[7] Glance Project 2021 [software] version 1.6.3 Available from https://atlas-glance.cern.ch/atlas/aces

[8] Networkx, https://networkx.org/

[9] Albert Myers, *Complex system reliability*, Springer 2010, http://link.springer.com/10.1007/978-1-84996-414-2