2015 International Nuclear Atlantic Conference - INAC 2015
São Paulo, SP, Brazil, October 4-9, 2015
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR - ABEN
ISBN: **978-85-99141-06-9**

APPLICATION OF AN EXPERT SYSTEM FOR REAL TIME DIAGNOSIS OF THE LIMITING CONDITIONS FOR OPERATION IN NUCLEAR POWER PLANTS

Gustavo Varanda Paiva[1], Roberto Schirru[1]

[1] Nuclear Engineering Program - COPPE
Federal University of Rio de Janeiro
Ilha do Fundão, s/n
21.945-970 Rio de Janeiro, RJ
gustavopaiva@poli.ufrj.br

ABSTRACT

In the history of nuclear power plants operation safety is an important factor to be considered and for this, the use of resistant materials and the application of redundant systems are used to make a plant with high reliability. Through the acquisition of experience with time and accidents that happened in the area, it was observed that the importance of creating methods that simplify the operator work in making decisions in accidents scenarios is an important factor in ensuring the safety of nuclear power plants. This work aims to create a program made with the Python language, which with the use of an expert system be able to apply, in real time, the rules contained in the Limiting Conditions for Operation (LCO) and tell to the operator the occurrence of any limiting conditions and the occurrence of failure to perform the required actions in the time to completion. The generic structure used to represent the knowledge of the expert system was a fault tree where the events of this tree are objects in the program. To test the accuracy of the program a simplified model of a fault tree was used that represents the LCO of the nuclear power station named Central Nuclear Almirante Álvaro Alberto 1. With the results obtained in the analysis of the simplified model it was observed a significant reduction in the time to identify the LCO and the success of real-time analysis of LCO, showing that the implementation of this program to more complex models of fault tree would be viable.

1. INTRODUCTION

Since the beginning of the use of nuclear power for commercials purpose, the security factor has been of extreme importance. Systems with high reliability are used for ensure the plant safety, however, with time, was observed that combined with a high reliable system was also necessary to have a qualified team and effective forms of man-machine interaction.

At the accidents of Chernobyl and Three Mile Island, was identified some failure due to human factors, like, the failure of the cognitive capacity of the operators when those are submitted to a large number of signals and tasks in a short period of time. After those accidents, new systems were applied to the nuclear power plants to ensure the security.

This paper is based in a research made by Gustavo Paiva [1] that had as objective to create a program that use Artificial Intelligence to automate the identification of violations of the Limiting Conditions for Operation (LCO) and to keep the operators informed, in real time, of which are the required actions to be made.

To achieve the proposed objective of this work, was used an Expert System (ES) written with the Python 2.7.6 software. To represent the rules presented on the LCO was used a fault tree structure which was obtained from the CAFTA Fault Tree Analysis software created by the Electric Power Research Institute (EPRI) [2].

The reason for the use of a ES was the use of heuristic to solve the problem, the possibility of the program explain to the user how it arrived at a solution and because of how its structure works, where it is possible to change the rules and the knowledge used for the solution without having major changes to the codes used to solves the problem.

The Python software was used because of it being a free software, being user friendly and because of its potential with list which was highly used to create the codes for the program proposed in this paper.

When searching the literature for studies that use ES for diagnosis, none of those found had as objective to solve the problem of monitoring the LCO's violation, but was found studies as the one of Angeli and Atherton (2001) and Yu Quian *et al* (2003) [3] that are used for others types of industry.

The next topics of this paper were written with the following structure: 2) Explanation of what is a LCO; 3) Methodology used to achieve the study objective; 4) Introduction to the program structure; 5) Study results; 6) Conclusions obtained from the study.

2. LIMITING CONDITIONS FOR OPERATION

The LCO are presented in the chapter of the Final Safety Analysis Report (FSAR) which present the technical specifications of a nuclear power plant. Its function is to establish the minimum level of performance or the working capacity of components and systems that are required for the safety plant operation.

In order to establish if a LCO is being violated is necessary to know if it applicability conditions are the same as the reactor conditions. One of those conditions is the operation mode of the reactor where there are 6 operations mode named as: 1) Power Operation; 2) Startup; 3) Hot Standby; 4) Hot Shutdown; 5) Cold Shutdown; 6) Refueling.

The LCO structure presents the system that it is applied, its name, applicability conditions and charts with the related actions and surveillance requirements. The Figures 1 present a LCO example taken from the United State Nuclear Regulatory Commission website [4] .

3.1 REACTIVITY CONTROL SYSTEMS

3.1.1 SHUTDOWN MARGIN (SDM)

LCO 3.1.1 SDM shall be within the limits specified in the COLR.

APPLICABILITY: MODE 2 with $k_{eff} < 1.0$,
MODES 3, 4, and 5.

ACTIONS

CONDITION	REQUIRED ACTION		COMPLETION TIME
A. SDM not within limits.	A.1	Initiate boration to restore SDM to within limits.	15 minutes

SURVEILLANCE REQUIREMENTS

SURVEILLANCE		FREQUENCY
SR 3.1.1.1	Verify SDM to be within the limits specified in the COLR.	24 hours

Figure 1: LCO example.

The Actions chart is divided into three columns named as Condition, Required Action and Completion Time which will be explained hereafter.

The Condition column presents a list of LCO's conditions which are indexed with an uppercased letter in alphabetic order. Those conditions must be read in alphabetic order because some of them may depend of orders in to start. In case of a condition start because of an event, subsequent events that would activate that condition will not result in separated start of it unless there is a specification in the LCO that says that this may happen.

The Required Action column shows the actions that must be done if a condition is started. They are indexed with the same letter as it condition plus a number. A condition may presents more than one action and those will be connected with a logic operator as "AND" or "OR".

The Completion Time column contains the time limit to complete the required actions. The time start to run for each required actions at the moment that the condition starts. If the time limit is exceeded, the condition related to it fails and that may cause another conditions to start. In some cases the time limit may be extended. The Completion Time column may also contain logic operator and can determine a time frequency for the required action.

There is also the Surveillance Requirements chart which contains the surveillance to be made and the frequency that this must be done.

3. METHODOLOGY

The first step of this work was to study the set of rules presented in a LCO. With that study was observed which characteristics would be necessary to create a generic structure to solve the proposed problem.

The simplified structure used in this work contains the capacity to solve the logics operations "AND" and "OR" that may be presented at the LCO's Required Actions column and the ability to manipulate the chronometers used in the time count required for the Completion Time column. The generic structure is presented as a fault tree, where each element is a LCO's event.

Due to the ES characteristic of having the knowledge base separated from the inference engine, as shown at figure 2, the use of a fragment of the generic structure is able to represent the entire structure, where the only difference is the addition of new data to the knowledge base.

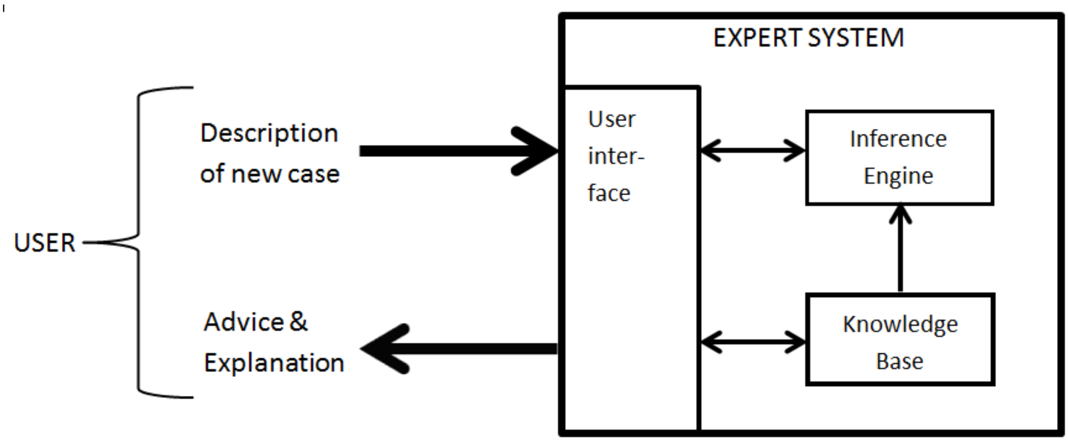

Figure 2: Representation of the structure of an expert system

The monitoring of LCO's violations requires to be done in real time. In this work, was used an interval of 10 seconds for each data input. This interval was chosen because it was considered that an event that happens at the beginning of the cycle would keep been true for the hole period.

The input data for each cycle are provided by the Sistema Integrado de Computadores de Angra – SICA (SCHIRRU e PEREIRA, 2004), that is responsible for monitoring in real time the essentials parameters to determine the safety state of the power plant in an emergency condition as well as the monitoring of it during normal operation [5].

With Python 2.7.6 software, was created a program that read the rules of the LCO in order to create a fault tree and also read the initial conditions for the events. At the next step, the inference engine solves the fault tree updating the data and starting the chronometers for each LCO conditions started. For each condition that had been running or ended at this cycle, a message is shown to the user with the LCO condition information.

4. PROGRAM STRUCTURE

To create the fault tree used in the program was necessary to determine how this would be built. At the fault tree structure, each event is classified as an object that has attributes to identify it. The figure 3 shows which those attributes are. On the next paragraphs will be shown how those attributes was implemented in the program.

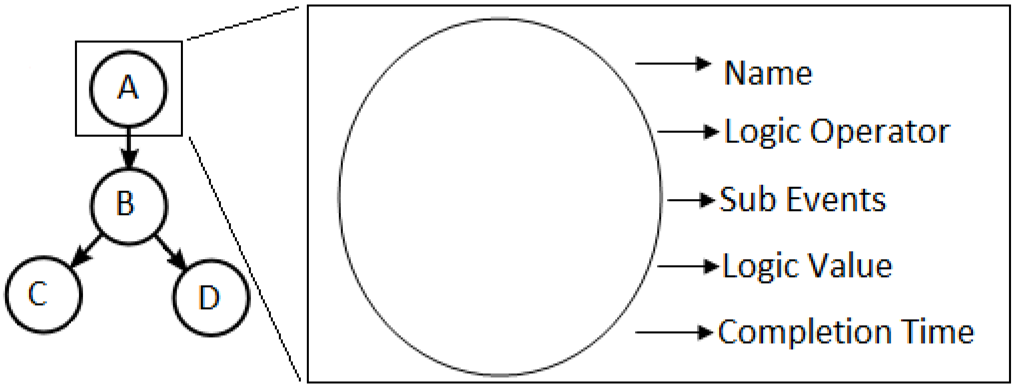

Figure 3: Determination of the attributes of each event on the fault tree.

In order to identify each event and to know each attributes are bounded to it, two lists were created for each event. One list determines the structure of the fault tree and the other has the event logic value and completion time. Those lists have the following structures:

- List 1 = [“event name”, “logic operator”, “sub events”]
- Example: List 1 = [“B”, “or”, “C,D”]

The possible logics operator are “AND” and “OR”. If an event hasn’t a logic operator then the “logic operator” attribute will be given as “leef” and the “sub events” attribute will be empty and given as “”.

- List 2 = [“event name”, “logic value”, “completion time”]
- Example: List 2 = [“A”, “F”, “60”]

The possible logic values are “T” for true and “F” for false. The completion time is presented in seconds and in case of an event that not possess completion time, the value given to that attribute is “0”.

5. RESULTS

In order to verify the functionality of the program, tests were made with some LCO that the simplified model was able to solve. The test used a fragment of the fault tree from the Angra 1 nuclear power plant made with CAFTA software, a fragment of that fault tree is presented in figure 4.

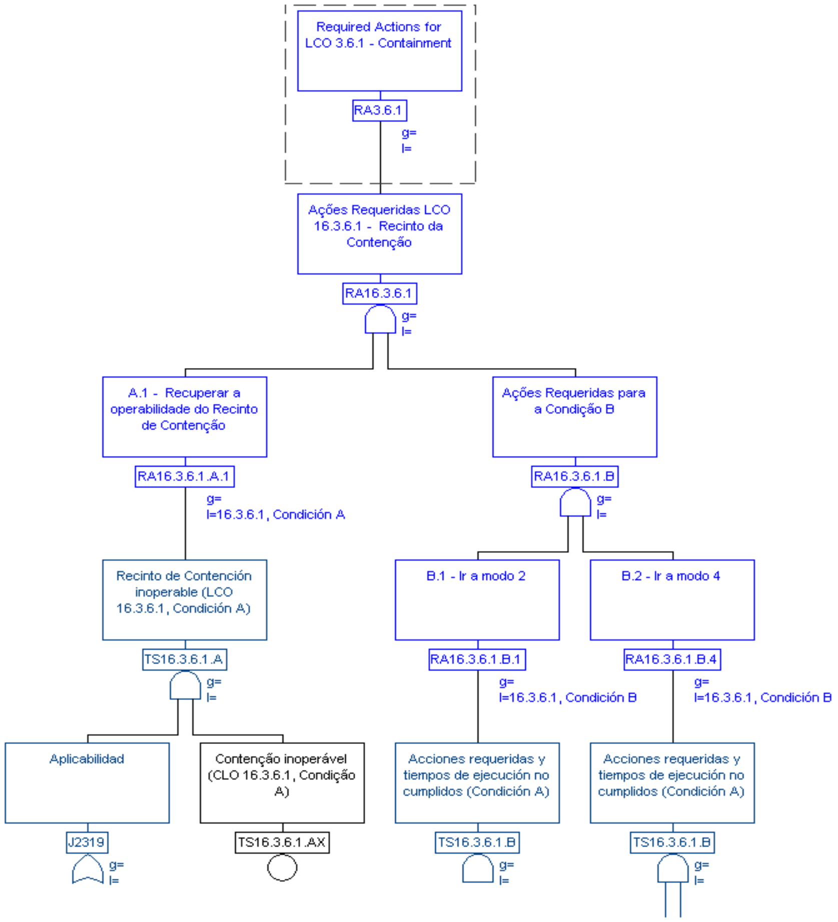

Figure 4: Fragment of the fault tree used to test the program.

After performing the tests, was possible to note the accuracy in the solution of logical operations, the data acquisition worked properly, the program succeeded at the management of the competition time and it was able to inform the user of an event of LCO start-up and failure in real time. An example of output data is given in the figure 5.

```
The LCO 16.3.6.1 condition A.1 started at 00:00:00
Time to conclude the LCO 16.3.6.1 A.1 00:00:06

time running 0 seconds

[LCO, Time, Time to conclusion]
[]

Atention! Time to conclusion of the LCO RA16.3.6.1.A.1 finished

The LCO 16.3.6.1 condition B.1 started at 00:00:06
Time to conclude the LCO 16.3.6.1 B.1 00:00:36

The LCO 16.3.6.1 condition B.4 started at 00:00:06
Time to conclude the LCO 16.3.6.1 B.4 00:01:12

________________________________________________________________________________
time running 10 seconds

[LCO, Time, Time to conclusion]
[['RA16.3.6.1.A.1', 10, 6], ['RA16.3.6.1.B.1', 4, 36], ['RA16.3.6.1.B.4', 4, 72]]

________________________________________________________________________________
time running 20 seconds

[LCO, Time, Time to conclusion]
[['RA16.3.6.1.A.1', 20, 6], ['RA16.3.6.1.B.1', 14, 36], ['RA16.3.6.1.B.4', 14, 72]]
```

Figure 5: An example of the output data given by the program.

Due to the ES characteristic of having the knowledge base separated from the inference engine, the fact that the test succeeded for a fragment of the fault tree represents that if a test with the same restrictions be made to a different tree, the program will also work properly.

Although any tests was made comparing the time used by the operators to manually check the violation of the LCO and the time used by the program, is believed that a significant reduction was acquired. Since no other study with a similar solution to the problem was found, was not possible to compare it results with others works.

6. CONCLUSION

Since the results obtained for the simplified fragment of the fault tree was positive, it is possible to considerer that with the entire fault tree the results will be the same, since the only requirement will be the addictions of new rules and facts and that can be done without major changes at the ES.

Another fact obtained at the end of the study is that the choice to use an ES was good because of it characteristic of having the knowledge base separated of the inference engine and the choice of using Python software as the programming language was welcomed because of it potential with list modules and it friendly environment.

The results of this work shows that the idea of this study have potential to be a tool to support the operation of nuclear power plants and contribute to it safety.

ACKNOWLEDGMENTS

We would like to acknowledge CNPq (National Research Council, Brazil) and FAPERJ (Foundation for research of the state of Rio de Janeiro) for its support to this research.

REFERENCES

1. PAIVA, G. V., "*Aplicação de um Sistema Especialista para o Diagnóstico em Tempo Real das Condições Limite de Operação em Usinas Nucleares".* Projeto de Graduação, POLI/UFRJ, Rio de Janeiro, RJ, Brasil, 2015.
2. "Computer Aided Fault Tree Analysis System (CAFTA)" http://www.epri.com/abstracts/Pages/ProductAbstract.aspx?ProductId=000000000001015514.
3. "Expert Systems: From Expert's Knowledge to Real-Time Systems" http://www.tmrfindia.org/eseries/ebookv1-c4.pdf.
4. "ML13220A943 - AREVA Design Control Document Rev. 5 - Tier 2 Chapter 16 - Technical Specifications - Tech Spec 3.1 " http://pbadupws.nrc.gov/docs/ML1322/ML13220A943.pdf.
5. NICOLAU, A. S., *Algoritmo Evolucionário de Inspiração Quântica Aplicado na Otimização de Problemas da Engenharia Nuclear.* Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2014.