2017 International Nuclear Atlantic Conference - INAC 2017
Belo Horizonte, MG, Brazil, October 22-27, 2017
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR – ABEN

ACCIDENT DIAGNOSIS OF THE ANGRA 2 NUCLEAR POWER PLANT BASED ON INTELLIGENT REAL-TIME ACQUISITION AGENTS AND A LOGICAL TREE MODEL

Gustavo V. Paiva[1] and Roberto Schirru[1]

[1] Departamento de Engenharia Nuclear - COPPE
Universidade Federal do Rio de Janeiro
Ilha do Fundão, s/n
21.945-970 Rio de Janeiro, RJ
gustavopaiva@poli.ufrj.br
schirru@lmp.ufrj.br

ABSTRACT

This work aims to create a model and a prototype, using the Python language, which with the application of an Expert System uses production rules to analyze the data obtained in real time from the plant and help the operator to identify the occurrence of transients / accidents. In the event of a transient, the program alerts the operator and indicates which section of the Operation Manual should be consulted to bring the plant back to its normal state. The generic structure used to represent the knowledge of the Expert System was a Fault Tree and the data obtained from the plant was done through intelligent acquisition agents that transform the data obtained from the plant into Boolean values used in the Fault Tree, including the use of Fuzzy Logic.
In order to test the program, a simplified model of the Almirante Alvaro Alberto 2 Nuclear Power Plant (Angra 2) manuals was used and with this model, simulations were performed to analyze the program's operation and if it leads to the expected results. The results of the tests presented a quick identification of the events and great accuracy, demonstrating the applicability of the model to the problem.

1. INTRODUCTION

Because of the potential damage that a nuclear accident can cause, the operation of nuclear power plants demands a high level of safety measures. Some of those measures are the use of highly reliable systems and the use of the defense in depth model.

Accidents like the one in Three Mile Island (Pennsylvania – USA) and Chernobyl (Pripyat – Ukraine) showed that it is also required that those who work in the nuclear power plant environment must be well qualified for the job so that human factor failures don’t happen.

One lesson learned from those accidents was that when the power plant is under an accident or transient the operators suffer from a high amount of stress and based on some studies [1], high amounts of stress can cause a reduction in the operator performance, which may cause a worse situation. In order to reduce stress on operators, systems are being developed for the nuclear power sector with the aim of simplifying the work of operators.

This paper was based on the research of Gustavo V. Paiva [2] which the objective was to create a program that identify the occurrence of accidents or transients on the power plant in real-time by applying the rules presented in the Operation Manual (OM) in the form of an Diagnostic Logic Tree (DLT) and to guide the operators by showing which chapter have the counter measures to put the power station back to the normal state.

The program proposed in this paper uses Expert System (ES) [3] to solve the rules of the DLT. Those rules were presented with a Logic Tree structure that accepts the values of “True”, “False” and “None” for the events. The addition of the “None” value is important for the program since in case of an input entry is not possible to be achieved the program must inform the user that it is not possible to be sure of what is the state of the power plant since giving a wrong information could lead to a worsening of the scenario. That tool is relevant for a system that works in real-time since any equipment can fail in reading a parameter.

Intelligent Acquisition Agents [4] were used in order to get the data of the nuclear power plant and treat them so that they could be used as input of the Logic Tree. Some examples of those treatments are: Calculate the rates that a variable change per cycle or apply techniques of fuzzy logic [5].

In some of the rules from DLT, words like “raising” or “high” are used as part of the rule. Those words are called as fuzzy values and require the use of fuzzy logic techniques. Since the power plant only offers numeric or Boolean values, in order to solve those rules, an Intelligent Acquisition Agent gets the numeric value, also known as crisp values, and with the use of a membership function, it determines how sure the program is when categorizing that crisp value as the fuzzy value evaluated. The certainty of the program in categorizing a crisp value to a fuzzy value is given by a number in the interval of [0,1] and is called as membership value.

The language used to create the program was Python 2.7.6. The reason why Python was chosen was that it is a free software, easy to be learned and because of the high use of lists, that is a strong factor of the language. A module from Python called Tkinter was used to create the user interface that gives the information to the user and allows the interaction with the program.

Others works related to the use of Artificial Intelligence techniques for diagnose of nuclear power plant was found on the literature. In the next 3 paragraph are given examples of this works.

In 1992, BARTLETT and UHRIG [6] used an artificial neural network to diagnose 7 scenarios of a nuclear power plant. They used 27 variables each 0.5 seconds for a total of 250 seconds. The results of the research were satisfactory, but the analysis was not able to answer that a non-postulated scenario was unknown.

Alvarenga *et al* [7] used artificial neural network, fuzzy logic and genetic algorithm to diagnose 16 postulated accidents of the Angra 2 nuclear power plant. The work uses an Adaptive Vector Quantization artificial neural network to create the centroids for each accident analyzed and then uses fuzzy logic to define the influence zone of each accident and finally uses the genetic algorithm to set the centroids position on the time axis.

Douglas Salmon [8] used an ES to identify transients of a nuclear power plant using the DLT. In his work, he uses propositional language to create the rules of the system that can identify

27 scenarios for 13 different accidents. The identification process is performed in four layers where the first identify if the power plant is no longer on the normal state of operation, the second identify if the accident match one of the postulated accident proposed, the third identifies the accident and says to which chapter of the OM the user must go and the last identify the specific scenario of the ongoing accident.

This paper was written with the following structure:

The second session introduces the DLT and explains from where it comes, what are the objectives of it and how is the structure of it.

The third session explains how each of the three programs was created and how they work

The forth session shows how was the testing and simulating process

And the final session gives the conclusion of the work

2. DIAGNOSTIC LOGIC TREE

The DLT of the nuclear power plant studied is presented on the OM and as the document says, its purpose is: "To assist the operation workers in identifying plant disturbances or internal accidents that happen after an unplanned reactor power change or the initiation of a reactor Trip. It also allows a quick location of the OM chapter that describes the countermeasures needed to deal with the event." [9]

The structure of the DLT consists of two paths, but only the DLT with initiation of a reactor Trip (IRT) was used on this work:

- Disturbances or accidents that lead to the initiation of a reactor Trip (DLT with IRT). [9]
- Disturbances or accidents that lead to variations of the reactor power (DLT without IRT). [6]

For each of those paths, each element of the logic tree consists of a "true or false" question related to the state of the power plant. By answering that question, a new element will be indicated until it reaches a final event that will say which chapter of the OM must be checked in order to obtain the countermeasures of the ongoing scenario. Figure 1 shows an example of the DLT with IRT.

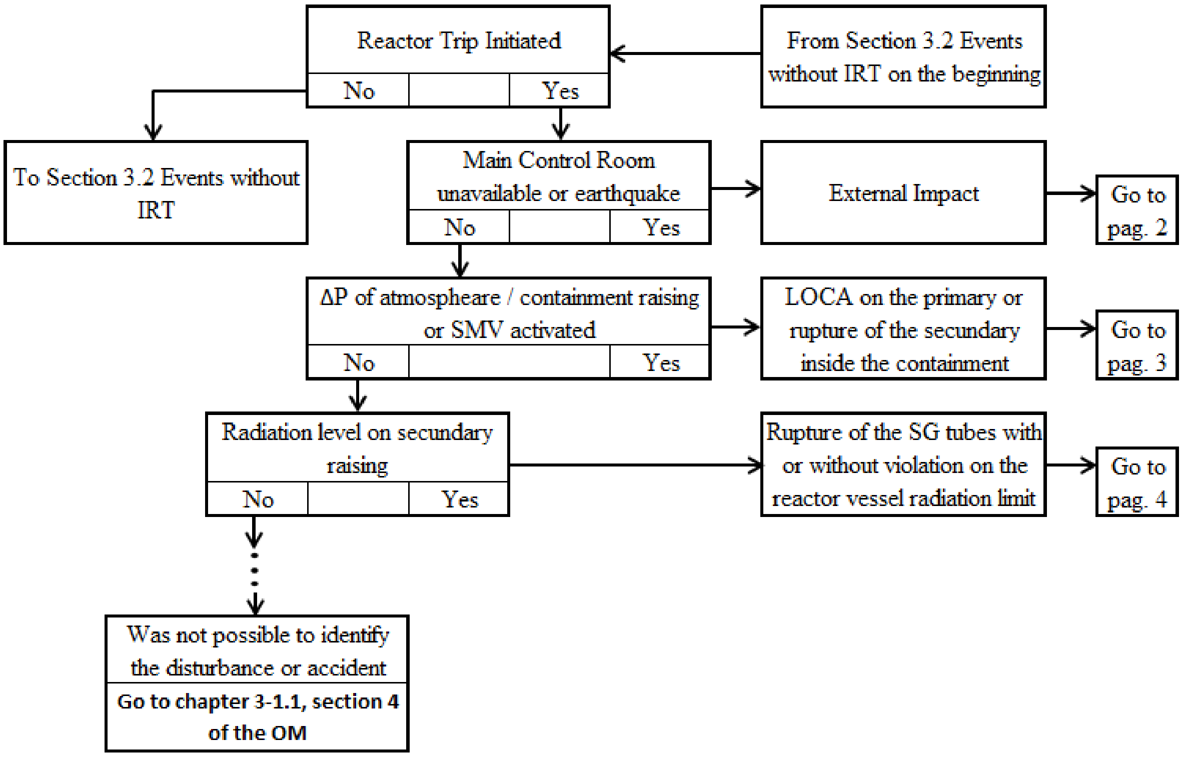

Figure 1: Example of a DLT with IR

If a scenario that isn't on the DLT scope happens, both paths are able to inform the user that the program was not able to identify what is happening to the power plant, leading the operator to the correct procedure. This means that the DLT is able to give a "Don't Know" answer if an unknown event happens.

3. METHODOLOGY

The program proposed on this paper was divided into three sub programs and in this session will introduce each of them and show how it was made and what is the objective of each one. The programs names are: 1) Fault Tree Generation and Solution Program (PGSAF); 2) Program of Acquisition and Manipulation of Input Data by Intelligent Acquisition Agents (PAMDE); 3) Graphic Interface Program (PIG).

3.1. Program of Acquisition and Manipulation of Input Data by Intelligent Acquisition Agents

The PAMDE was created in order to get the data from the power plant or given by the user and answer the rules presented on the initial events of the fault tree.

The data acquisition happens every cycle of the system and at this work, the cycle length was defined as 10 seconds. The first phase of this program is to acquire the format of the fault tree, this step is only done on the first cycle since the fault tree structure doesn't change with time. At the start of every cycle, the program checks the database in search of the structure of the rules for each event and also any relevant information needed to answer the rules.

Each initial event of the fault tree has a chain of rules that must be answered in order to get the final value of the event. The information required to solve those rules aren't always given by the power plant system with some demanding a previous treatment. The input data was divided into 4 groups and for each group was created an Intelligent Acquisition Agent (IAA) designed to treat the case related to that group. The four groups are: 1) Numeric Variables (NV): Those are the rules that use numeric variables that don't require a special treatment or that require simple operations. 2) Variation Rates (VR): In this group are the rules that require the value of a variable in multiple cycles so that it is possible to calculate the rate of change with time. 3) Boolean Variable (BV): This group covers the rules that use Boolean variables as "true", "false" or "yes" or "no". 4) Fuzzy Variables (FV): This group covers the rules that require the use of Fuzzy Logic principles to transform a fuzzy value in a crips value with the use of a membership function. There is a possibility that a rule belongs to more than one group, in that case, more than one IAA will be activated on that rule.

From the groups listed above, the VR and FV are the ones that demand special explanations of the process. For the first, the IAA identifies the variable that needs to be tracked and it records the value of that variable on the last cycle so that it becomes possible to calculate the difference between the two cycles and divide the value by the length of the cycle. If the value of the variable on the actual cycle or on the previous one is not defined, then the rate of that variable will also be. For the FV group, is required the use of the membership function of that variable to transform the crisp value into fuzzy values. The membership function is previously defined by an expert of the topic and added to the database. After using the membership function, if the biggest membership value belongs to the fuzzy value dated on the rule, then the rule is considered "True".

After the IAA treatment, it is possible to answer the rules and give a final value to them where this value can be "True" or "False" if the IAA succeeded with the treatment or "None" if it was not possible to determine because of the lack of information. As output of this program is given a list with all initial event and values that will be sent to the PGSAF.

With the use of the User Interface, is possible to force a value of one of the initial events and with this, the final list sent to the PGSAF will take into consideration any change imposed by the used on the previous cycle.

3.2. Fault Tree Generation and Solution Program

The goal of PGSAF is to create and solve the fault tree in each cycle by getting the information given by the PAMDE and as result, give the final value of each event to the PIG together with a list of all the alarms that started on the cycle.

The creation of this program started by studying the DLT with IRT in order to get the information needed to transform a logic tree into a fault tree and also to define which models will be used to solve the fault tree. The models used to solve the fault tree was a search from the bottom to the top together with a search on amplitude

The fault tree structure consists of the lists, one for the events and one for the logic gates. Those lists contain information as the identification of the event, its value, identification of the gate, gate logic operator and all the inputs and outputs of the gate.

With the addition of the value “None” as one of the 3 possible states of an event, it was needed to apply this new state to the truth table of each logical gate. Also the addition of a new logic gate, that was called as “Inhibit-None” was needed. Table 1 shows each gate used on this work together with the truth table containing the 3 possible states.

Table 1: Truth table for each event used on this work. (1 means "True", 0 means "None" and -1 means "False".

AND			OR			INHIBIT_NONE	
A	B	X	A	B	X	A	X
1	1	1	1	1	1	1	-1
1	0	0	1	0	1	0	1
1	-1	-1	1	-1	1	-1	-1
0	1	0	0	1	1		
0	0	0	0	0	0		
0	-1	-1	0	-1	0		
-1	1	-1	-1	1	1		
-1	0	-1	-1	0	0		
-1	-1	-1	-1	-1	-1		
NAND			NOR			NOT	
A	B	X	A	B	X	A	X
1	1	-1	1	1	-1	1	-1
1	0	0	1	0	-1	0	0
1	-1	1	1	-1	-1	-1	1
0	1	0	0	1	-1		
0	0	0	0	0	0		
0	-1	1	0	-1	0		
-1	1	1	-1	1	-1		
-1	0	1	-1	0	0		
-1	-1	1	-1	-1	1		

Once the fault tree is solved, the program searches its database to find if the final values of each event lead to an alert. If any alert is activated, it is stored in a list together with any other relevant information to that alert. As final step, the alert list and the list with all events solved is sent to the PIG program where it will be shown to the user.

3.3. Graphic Interface Program

The PIG objective is to show the user every alert that happens on the plant related to the DLT, giving him the option to interact with the interface to see the chapter of the OM related to the ongoing accident or to set the value of a variable.

The PIG structure consists of the main screen, shown in Figure 2, which is divided into 3 areas. The first, on the left side, is called Schedule of Events and is where the alerts appear on every cycle, where on the top will be the newest alerts and to the bottom the oldest. The second area is called Event Detailing and it is present on the right side of the main screen. If the user clicks on an alert on the Schedule of Events area, the description of that alert will be shown. The third area, on the bottom of the main screen, has the button that brings the input menu, where you can force a value to one of the initial events of the fault tree and that will happen on the next cycle. Also at the bottom has a set of three status indicators that show if the program is operating, if the reading of the inputs were succeeded and if the power plant is in its normal state or on an IRT.

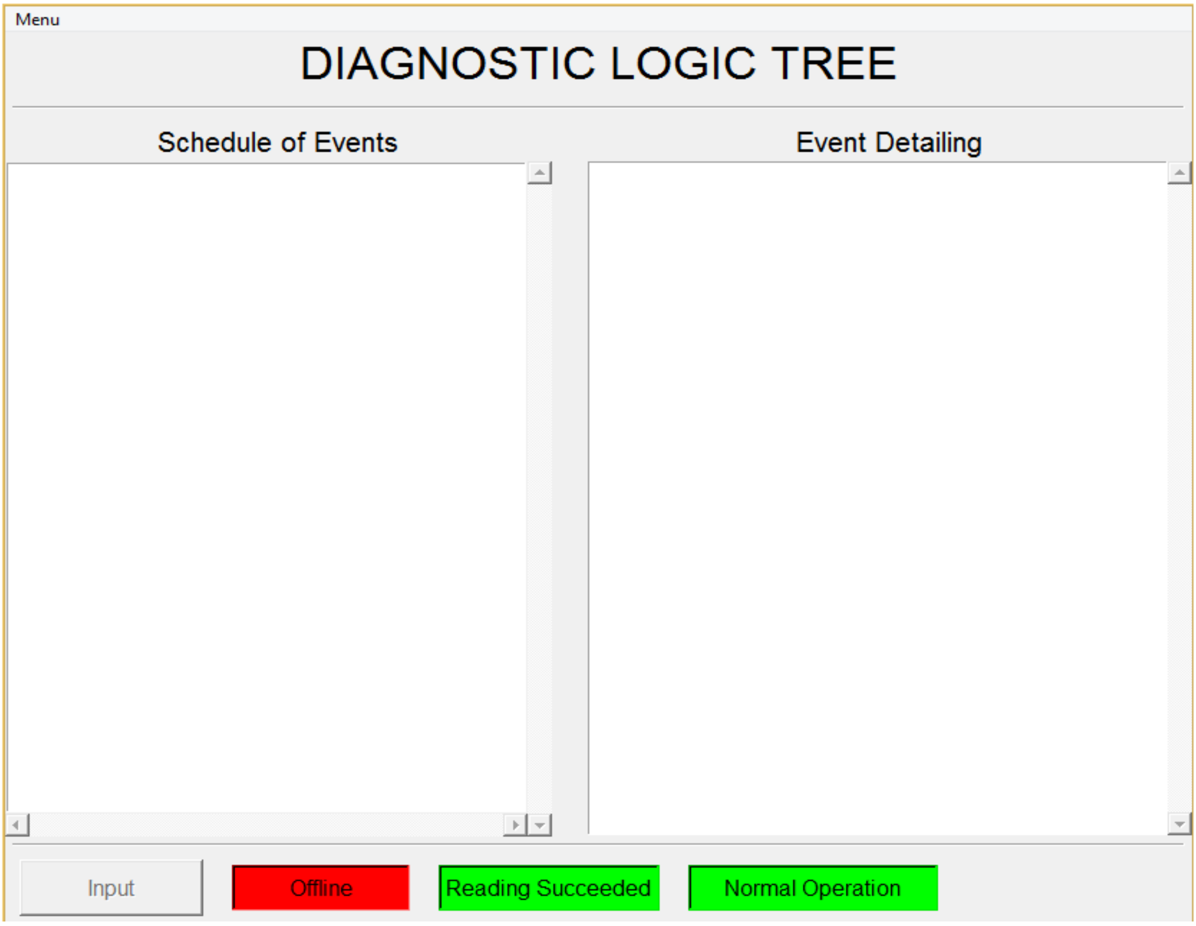

Figure 2: Demonstration of the Main Screen of PIG.

The Event Detailing area gives information as the moment that the alert happened and the identification number and the name of the alert. Others additional information as the OM chapter that must be checked and a list of the events that caused the reading error are given depending on the alert.

The reason why was decided to give the user the option to change the input is that some of the data required by the program don't come directly from the power plant computer system.

4. SIMULATIONS AND RESULTS

4.1. Simulations

By studying the fault tree, were identified five different models of alert that the program could give, those are:

- Normal operation of the power plant;
- Occurrence of one or more postulated accidents;
- Total failure on the input reading: Happens when the program lack of information to identify if there was a postulated accident, a non-postulated accident or it is running on normal state;
- Occurrence of a non-postulated accident;
- Failure to identify an accident: Happens when an IRT is activated, but because of the lack of input data is not possible to identify which accident happened.

Was decided to make one test for each of the five models and for that was used a simplified scenario of the fault tree. The studied scenario considered the part related to the LOCA accident of the DLT with IRT and used 3 variables given by a simulation executed on the Cen tral Nuclear Almirante Álvaro Alberto (Angra 2) together with 7 edited variables, in order to be possible to test each of the five models. The success on those tests implies that the progra m would succeed with the DLT with IRT, since it would only require the addition of new rule s to the fault tree.

Figure 3 shows the variables that were used in the simulation where each was evaluated in a p eriod of 60 seconds. As the program performs readings every 10 seconds, it was possible to p erform 7 analyses of the plant's state for each simulation.

```
[1, "Pressure in the Primary", 13.3803]
[2, "SMV acted", True]
[3, "Containment activity", 87.0]
[4, "PG Level", 62.5453]
[5, "emergency water supply actuated", True]
[6, "PG pressure comparator actuated", True]
[7, "Delta (P) containment / atmosphere", 39.0]
[8, "PZR level", 1.566426]
[9, "100 K / h cooling actuated", True]
[10, "Reactor Trip," True]
```

Figure 3: Example of Input data with the variables used on the simulation.

The five tests performed used the following structure: The first scenario goal was to test the case where the power plant operated in normal condition, staying this way for the 60 seconds.

At the second scenario, the sensor related to the occurrence of an IRT fails and this leads to alert of total failure of the input reading, which is solved in the next iteration. In this case, is desired that the program inform the problem on the data acquisition to the user.

The third scenario starts with the normal operation of the power plant, leading to a “Small rupture of the secondary inside the containment” that evolves into a different accident later. This test intention was to correctly show the evolution of the accident and to test if the user was able to check which chapter of the OM was related to the alarm.

On the fourth scenario the power plant starts on normal state but after 10 seconds an IRT occurs but the program can’t relate the ongoing variation pattern with any of the postulated accidents. On this test is expected that the program shows that it was not possible to identify the accident and alert the user.

The last scenario simulates the case where an IRT occurs but because of some failures on the data acquisition, it was not possible to tell which accident happens. On this case, the program must alert the user of what is happening and show which events had an acquisition failure.

Together with the five scenarios, was also tested the cases where the user would need to change the value of an initial event of the fault tree to test the input screen of the program.

4.1. Results

The tests result demonstrated that the three programs worked correctly, with the success of the data acquisition, data treatment by the IAA, the creation and solution of the fault tree and the transference of information to the user as the ability of the user to force an input. Figure 4 and 5 shows the main screen of the PIG on the 3rd and 5th scenarios simulated.

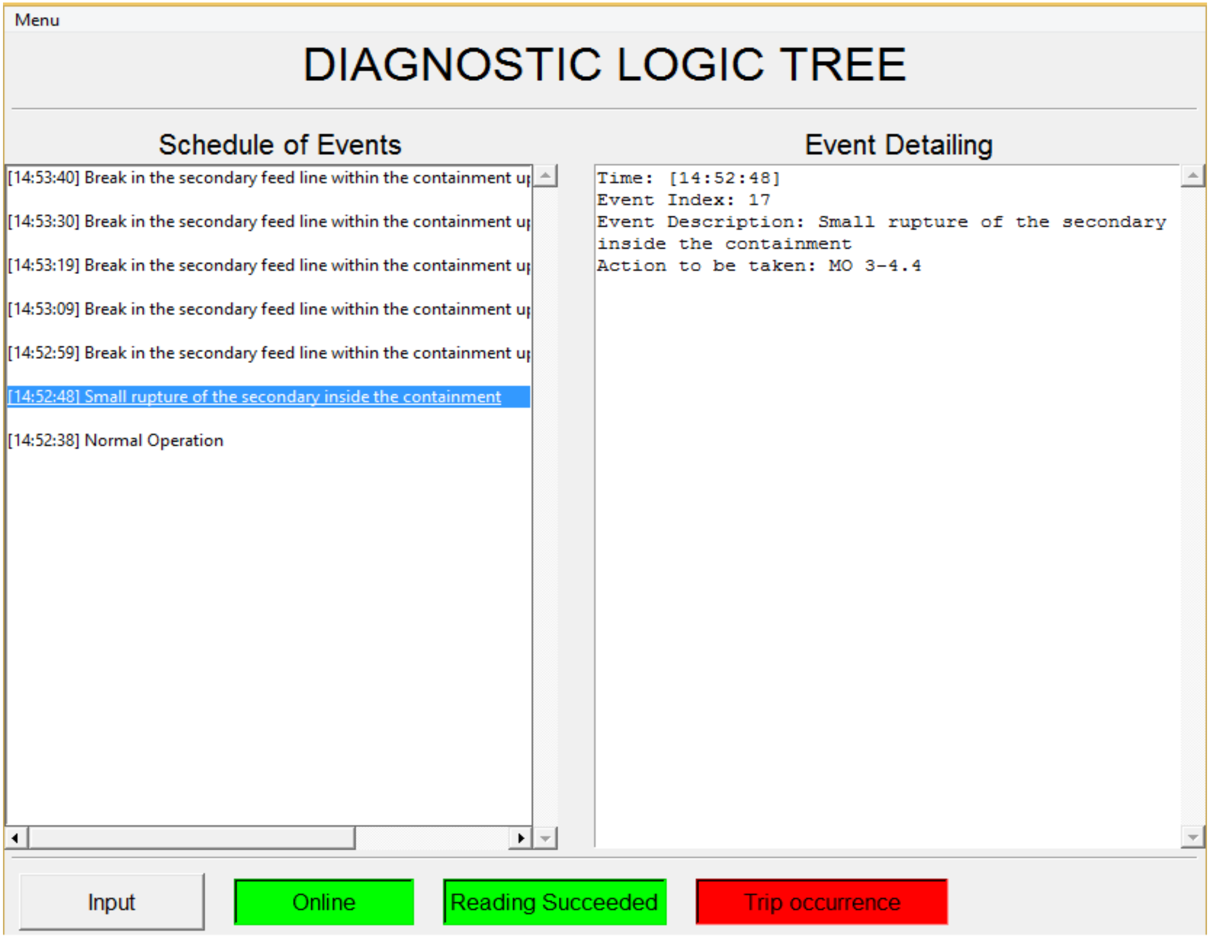

Figure 4: Results of testing the 3rd scenario.

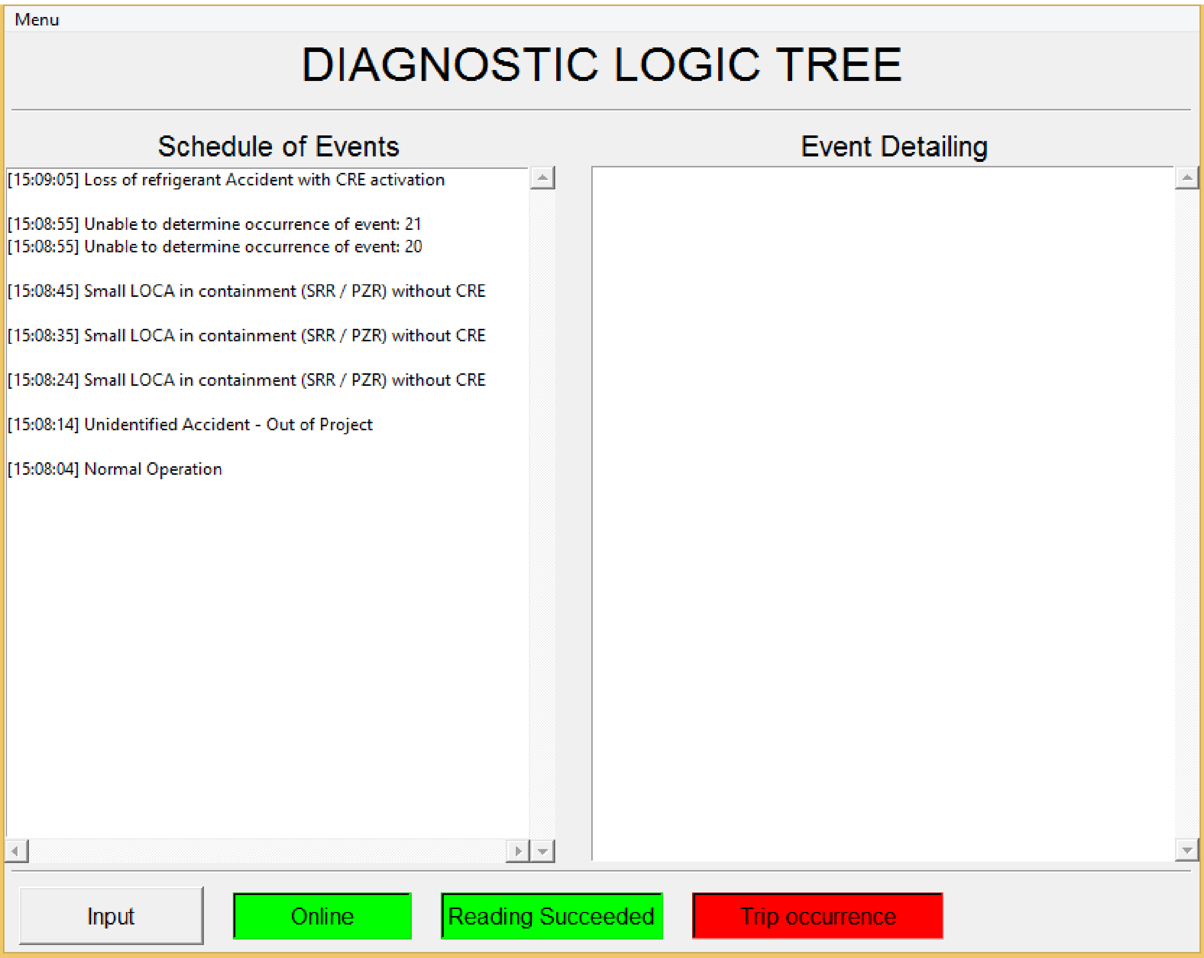

Figure 5: Results of testing the 5th scenario.

3. CONCLUSIONS

The objective of this work was to create a program that could simplify the work of operators by executing the accidents diagnose in real time. By automatizing this process this can reduce the stress of operators, reducing the probability of a failure caused by human factors.

The results demonstrated on the previous section shows that the programs correctly informed the user of the occurrence of accidents, achieving its objective.

One notable point of this work is the addition of the value "None" as one of the 3 possible values on the fault tree. With this addition, the program becomes able to inform the user that it is not able to determine the current state of the power plant.

Another important factor of this work is the capability to get the data from the nuclear power plant computer system, where on this work was used the SICA from Angra 2, and also allows the user to input any value required.

With the characteristic of the ES where the rules database is separated from the Inference Engine, it is possible to apply this program to others power plants with different rules where it will require minor changes to the program and the change on the database.

ACKNOWLEDGMENTS

The authors acknowledge CNPQ (National Research Council, Brazil) and Petrobras.

REFERENCES

1. Kreutzer, J., DeLuca, J., Caplan, B., *Encyclopedia of Clinical Neuropsychology*, Springer-Verlag, New York, USA (2011).
2. Paiva, G. V., "Diagnóstico de Acidentes da Usina Nuclear de Angra 2 Baseado em Agentes Inteligentes de Aquisição em Tempo Real, e em um Modelo de Árvore Lógica", M. Sc. Tese, COPPE/UFRJ, Rio de Janeiro, Brazil, 2017.
3. Jackson, P., *Introduction to Expert System*, Addison-Wesley, Rochester, USA, (1998).
4. *Applications of Intelligent Agents*, N. R. Jennings, M. Wooldridge, available in: https://inf.ufes.br/~tavares/dsba/artigos/jennings.pdf, accessed in: July of 2017.
5. Passino, K. M, Yurkovich, S., *Fuzzy Control*, Addison-Wesley, Columbus, USA, (1998).
6. BARTLETT, E. B., "Detecting Faults in a Nuclear Power Plant by Using a Dynamic Node Architecture Artificial Neural Network", Nuclear Science and Engineering, v. 116, pp. 313-325, 1994.
7. ALVARENGA, M.A.B., MARTINEZ, A.S., SCHIRRU, R., "Adaptive Vector Quantization Optimized By Genetic Algorithm For Real-Time Diagnosis Through Fuzzy Sets". Nuclear Technology, v. 120, pp. 188-197, Dec. 1997.
8. SALMON, D. R., Sistema Especialista Baseado em Níveis Progressivos de Diagnóstico para Identificação de Acidentes em Usina Nuclear PWR. M.Sc. Tese, UFRJ/COPPE, Rio de Janeiro, RJ, Brasil, 2013.
9. Manual de Operação da Usina – Identificação de Acidente – Árvore Lógica de Diagnósticos, Eletrobras Eletronuclear, 2010.