TECHNICAL STUDY RELATED TO AN INTEGRATED KNOWLEDGE-BASED EXPERT SYSTEM FOR DECISION ASSISTANCE OF NUCLEAR FACILITY OPERATORS

ANDREI- NICOLAE AVRAM and RAMONA NĂSTASE

Institute for Nuclear Research (RATEN ICN) Pitesti, Romania
e-mail: andreiavram98@yahoo.com

Abstract

The training of personnel regarding the administration and management of a complex system represents a main theme in the security, reliability and operation of nuclear facilities. The expert system is a software application that tracks a knowledge base for managing processes, as a decision support for the operator of the complex system. The paper can be a starting point in the development of expert systems with general applicability related to the driving of industrial processes. The main objective of this paper was to identify the optimal solution for the software implementation of a prototype expert system application intended to assist Fuelling Machine (F/M) Heads Test Rig operators, having a dual purpose. The expert system is a training and improvement tool for operators to ensure security and safety in the activities of the Test Rig, and on the other hand, it represents a way of accumulating knowledge related to the specific processes carried out within the tests. Thus, the loss of information related to the actual activities during the tests will be avoided. In the perspective of the development of the prototype expert system application, in the first part an evaluation of the development environments for expert systems was made, the second part of the work containing the details related to the specific domain of system development by performing an analysis of all the tasks involved to establish a process model for the system.

Keywords: *CANDU, Fuel Handling System, Training, Expert System*

Introduction

In a CANDU nuclear power plant, a very important component is the fuel handling system (FHS). FHS contains two Fuelling Machine (F/M) Heads, that operates simultaneous on the same fuel channel, upstream supplying with fresh fuel and downstream taking over the spent nuclear fuel [1]. To ensure that the fuel loading/ unloading is carried out safely and without the risk of a nuclear accident, the handling system must be a high-performance piece of equipment. This is why it must undergo complex tests and only after meeting certain performance criteria can it be used in the nuclear power plant. Taking into account these considerations, within the Institute of Nuclear Research Pitesti, Test Rig was built.

It is known that in order to operate and maintain nuclear facilities within the designed parameters, a rapid and adequate response to a rapid change in conditions or the resolution of unforeseen situations arising during tests is required. This makes it increasingly difficult to take a quick decision, and requires qualified and experienced personnel. Staff training becomes a vital task, and an increasingly sophisticated system using larger volume of data make the use of intelligent systems and decision support systems indispensable. The purpose of these systems is to query databases and synthesize information to assist the operator in making decisions that provide solutions for system administration and management to improve operational efficiency and safety.

This is how the option for a study on the effective management and operation of the Fuelling Heads (F/M) functional tests - which are housed in the Out of Pile Testing department at the Institute for Nuclear Research Pitesti - appears. The expert system replicates human reasoning by posing questions to a control room operator, enabling it to offer recommendations and conclusions in compliance with stringent operating procedures. Typically, an expert system is a knowledge-based system that emulates expert thinking by using artificial intelligence methods and techniques to solve significant problems in a particular domain of expertise

Through the improvement and training of specialized operating personnel, the Institute for Nuclear Research (RATEN ICN) Pitesti is developing a research program to support research on technologies and equipment that can ensure security and safety in the operation of the Fuelling Machine (F/M) Head Test Rig.

Current tasks in this research direction are, as follows:

- Comparative evaluation based on the specifications provided by the manufacturers in order to choose the optimal software development tool for expert systems;
- Development of block diagrams representing the Fuelling Machine (F/M) Heads Test Rig operating procedures - requires knowledge of the manifestations of diversified systems in different operating contexts;

• Expert system mockup as a conceptual model of the expert system. After the operating procedures of the Test Rig complex system will be represented as block diagrams, the inference mechanism will be developed and the knowledge base will be populated.

Advantages of expert systems

The advantages of using expert systems as decision support for operators of nuclear facilities, as well as other fields of activity, are numerous and vary depending on the way of implementation and the field in which they operate.

Some of the advantages are:

• Low cost - they can be developed easily with the help of expert system generators, they are easy to maintain and are available to the user permanently, without additional costs;
• Increased availability - the use of the expert system does not require a qualification in the field, they do not have a limited life, they can be used on any computer and are not affected by emotionality and stress factors;
• Reduction of hazards - they can be used in dangerous environments for humans, and through decision support, they increase security and safety in operation of facility;
• Increasing quality - the system allows the transfer of knowledge and a rapid training of personnel, thus helping the proper training of operators;
• Possibility of explanation - the system obtains a complete expertise of the respective field and can offer detailed explanations for the obtained solutions, to increase the user's degree of confidence;
• Speed - can provide decisions, advice and solutions in a short time, simplifying the decision-making process;
• Preservation of knowledge - the system includes a large volume of the experts' knowledge in the field, that is safely kept;
• Flexibility - if some procedures are updated or new ones are added, the system can be continuously expanded by completing the knowledge base;
• Planning and refining methods /procedures for solving domain problems appears as an indirect benefit of developing an expert system.

The structure of an expert system

An expert system is defined in [2] as a knowledge-based system that emulates expert thinking by using artificial intelligence methods and techniques to solve significant problems in a specific field of expertise; they are also intelligent artificial tools that in a narrow field of expertise provide justified intelligent decisions.

In general, expert systems are able to comment on the solutions and advice they have provided based on the knowledge already present in the system. So, the system has to be taught what advice and conclusions to give for each individual case. They offer the possibility to integrate new knowledge with the existing one in a flexible manner, enriching the knowledge base.

An expert system usually comprises the following two essential components:

Expert system = Knowledge + Inference

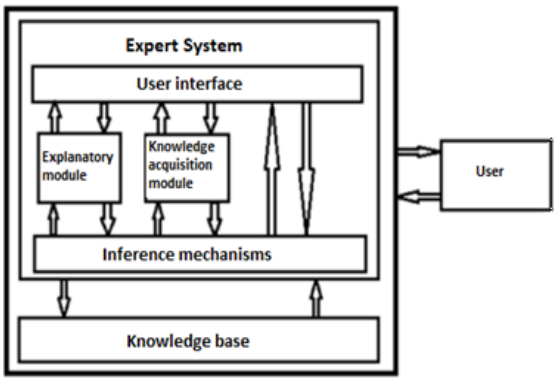


Fig. 1. Components of an integrated knowledge-based expert system

Knowledge base

The knowledge base is the working area of the system and is intended for storing specialized information, structures and rules. That is why expert systems are also called *knowledge-based systems*.

The knowledge base contains any information found appropriate by consulting various knowledge sources such as human experts, user manuals and databases. This is specific to the field in which the expert system is used. If the knowledge base is well organized and documented, it can be updated and expanded, but the model remains the same throughout the life of the system.

The knowledge base that is filled with operating procedures and instructions in accordance with the operating manual [1] is the focal point of knowledge acquisition within the study that is being conducted. It is logically organized to enable operators and maintenance personnel involved in the operation of the Fuelling Machine (F/M) Heads Test Rig technological facility to quickly acquire the knowledge they need.

Inference mechanism

The inference mechanism provides the driving force for the system and consists of algorithms to manipulate the knowledge represented in the knowledge base, control and direct problems solving by inference.

It has the following functions:
• determine which data is needed to solve the problem in question;
• obtain these data through the supporting software;
• uses the knowledge base content to make inferences;

The inference mechanism keeps performing these tasks until it is unable to or is no longer required to.

User Interface

The user interface is the component of the expert system through which the operator interacts with the system. It must provide user-friendly and intuitive graphics, presenting the user multiple dialog screens and using user-specific language.

The expert system questions the operator and based on the answers given, during the operation, the system offers conclusions and advice about the procedures that must be carried out.

The user interface makes it possible to deliver diagrams to the user in the form of questions and allows him to answer, thus the inference mechanism generates advices and conclusions extracted from the knowledge base.

Techniques and approaches

The rules on which a decision is made depend on the reasoning approaches and techniques used in the construction of the inference engine.

There are different starting points for a knowledge representation formalism before it can be considered to be suitable for encoding domain knowledge. An appropriate knowledge representation formalism:

• Should have sufficient power of expression to codify knowledge specific to the field;
• It has a clean semantic base, so that the meaning of the knowledge present in the knowledge base is easy to understand, especially by the user;
• It allows efficient algorithmic interpretation;
• It enables the user to explain and justify the solutions by demonstrating the rationale behind the questions posed to them and the steps taken to arrive at certain conclusions.

Declarative non-procedural techniques

In the case of using a procedural programming language, the program must specify exactly how to solve a problem. When using a non-procedural language, the program only specifies the goal to be achieved, leaving the system to determine the way to get there. This is the main reason that recommends using of non-procedural languages in the development of an expert system as decision support.

Object-oriented approach

This approach is a partly declarative and partly imperative programming method that can be materialized in classic procedural languages or languages specially intended for it. The basic idea is to design a program by first considering the data to be processed and classifying them by classes (objects), then defining the operations on these objects necessary to solve the problem. Thus, this approach is a "bottom-up" one, because it starts from data and detailed elements going towards general elements of processing and control.

Rule-based approach

The structure of this system includes a set of rules implemented by the developer, which have predefined results. An overview of this approach is given in [2], where the rule-based approach is defined by "if-then" coding statements (if X does Y, then Z is the result). Two important elements of rule-based techniques are "a set of rules" and "a set of facts", and by using these, developers can create a basic model of artificial intelligence [2].

Frame-based approach

Frame-based approach is a less widespread technique and can be characterized as a hybrid method from object-oriented approach in a particular form for organizing input information with control elements from rule-based systems.

Non-procedural and non-declarative techniques

The decision tree approach

A decision tree is a graphical representation of the various alternative solutions that are available to solve a given problem in order to determine the most effective courses of action. Decision trees are made up of nodes and branches - the nodes represent a test on an attribute, and the branches represent potential alternative outcomes.

Expert systems can benefit greatly from the use of decision trees when completing procedural tasks with predictive nodes in the form of decision rules. These systems allow for the detailed grading of all alternatives based on all possible options, ultimately providing results associated with each possible course of action. The combination of predictors with corresponding values provides a prediction for the target value.

The decision tree provides a highly efficient structure because it allows complex operating problems to be ordered in a logical, orderly sequence, with specific evaluations at each decision level.

The inference techniques that can be developed through decision trees are:

• *backward chaining inference (inductive)*, method shown in Fig. 2. Inference is made on the principle that if the conclusions are true, then the conditions are true;
• *forward chaining inference (deductive)*, method shown in Fig. 3. Inference is made on the principle that if the conditions are true, then the conclusions are true;
• *mixed- technique* that uses both deductive and inductive reasoning modes.

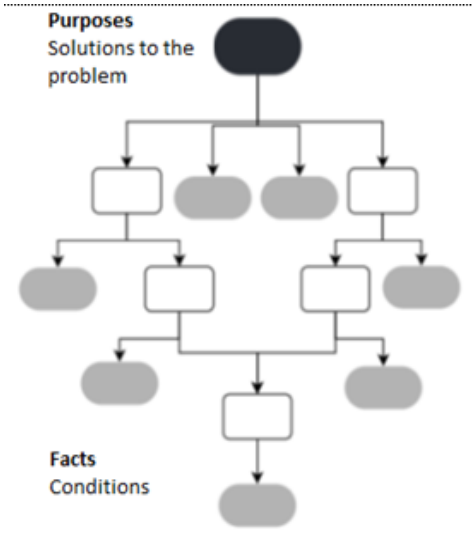


Fig. 2. Inductive inference technique

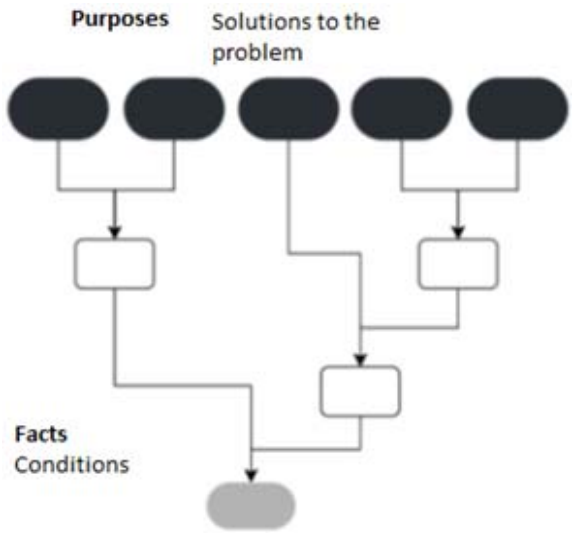


Fig. 3. Deductive inference technique

Neural network approach

This technique aims to represent tacit knowledge. A neural network is a technique in artificial intelligence that teaches computers to process data in a brain-like manner human, according to [3]. It is a type of example-based machine learning process that uses nodes or neurons interconnected in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and continuously improve.

Artificial neural networks try to solve complicated problems, such as document summarization or face recognition, with greater accuracy.

Programming languages and development tools for expert systems

Most expert systems are developed using a high-level programming language such as C, C++, Java, Python, but also in conventional languages developed especially for artificial intelligence applications, such as FORTRAN, LISP, PASCAL, PROLOG, etc. When using a high-level programming language, more attention must be paid to the implementation aspects of the system, and the resulting code can be opaque and become a very tedious task [4].

As a result, expert system generators were created that have the ability to create systems through interactive communication with programmers. This makes the task of programming much easier.

The expert system generator contains an artificial intelligence programming language at a higher level than the expert system programming language. Due to the special nature of these development environments, they are recommended for building expert systems, but not for general purpose programming. Because of this, it is often needed to develop certain functions in a procedural programming language.

However, expert system generators tend to be simpler and easier to learn than shells, because they provide both a high-level language and an environment for creating, modifying, and testing systems, including knowledge base editors, rule compilers, testing and performance tuning tools.

In Table 1, several expert system development software tools were evaluated through a comparison based on the specifications provided by the manufacturer.

Table 1. Comparative evaluation of four expert systems development software

		Basic features	Development	Functionality	Support and cost
Exsys CORVID (Exsys Inc.)	Rules	IF/THEN rules	•Non-procedural development environment	• General purpose	• Support includes technical documentation, but does not include support and software upgrades
	Logic	Systematic logic diagrams through a decision tree.	•Java based web implementation	• Counseling and supporting decisions	
	Inference and	Forward or backward chaining.			
CLIPS (Secret Society Software, LLC)	Rules	Complete environment for building rule-based and/or object-based systems.	•Procedural, declarative and rule-based programming language	•Application areas of diagnostics, advice generation, regulatory compliance, proposal configuration and decision support systems	• Support includes technical documentation and software upgrades
	Logic	Systematic rule-oriented logic, facts only, or a mixture of facts and rules.	•The inference engine decides which rules should be executed and when		•Available to the public
	Inference and control	Forward or backward chaining by combining objectives and facts in a rule.		•Shell for solving structured selection problems and supporting decisions	
FLEX (Logic Programming Associates Ltd)	Rules	Forward chaining production rules and backward chaining rules	•Allows compilation into a standalone application.	• General purpose	•Support includes basic training, technical documentation, phone and email support and software upgrades
	Logic	Systematic rule-based inference.	•Can be embedded in Java/ C*/ .NET through LPA tools	•Rich and versatile hybrid expert system development environment	
	Inference and control	Forward or backward chaining.	• Procedural programming language		
VisiRule (Logic Programming Associates Ltd)	Rules	Systematic logic diagrams through a rule-based decision tree.	• Non-procedural development environment with logic diagrams to extract, define and evaluate rules	• General purpose	•Support includes basic training, technical documentation, phone and email support and software upgrades
	Logic	Systematic rule-based logic.	•Generate executable code directly from diagram	•Application areas of diagnostics, advice generation, regulatory compliance and decision support systems	
	Inference and control	Forward or backward chaining.	•Easy user interface		

Exsys CORVID is a software application for building and implementing non-procedural expert systems. Implementation is done by developing logical blocks that define, organize and structure decision-making information.

CLIPS is an expert systems development tool by writing code with a rigorous object-oriented approach, which results in a more difficult and complex development environment.

FLEX surpasses most expert system shells by using an open architecture and allowing changes in its behaviour through a layer of access functions.

The latest development for Flex is VisiRule, a less complex AI-based software package than CLIPS, that enables the visual design of rule-based expert systems through a dedicated decision diagramming tool to extract, define and evaluate rules.

Information on specific system tools was taken from technical reports, magazine articles, user manuals, training materials, and discussions with company representatives.

Implementation of the prototype version

The purpose of the system is to provide advice and conclusions based on the answers received to the questions addressed to the user during operation. Thus, it was found that it is easy to organize the knowledge used in decision-making at each test stage in a logical diagram.

For the development of the expert system application, the construction of the decision tree with inductive chaining (see Fig. 2) represents the optimal method. The system starts from the root of the tree and then, depending on the answers given, it descends in the tree until it reaches a node corresponding to a solution.

This artificial intelligence application does not include "machine learning" or "deep learning" algorithms. Thus, the application must be learned by programmer each alternative course it can take during the operating activities, and based on the inferences it decides the next step to be performed.

For a basic model of the expert system, the main stages that must be performed during the operating activities have been established. These stages consist in the completion of the TECHNICAL PROVISION by the operator - "Starting the cold loop, with the test section equipped with 2 biological protection plugs and 12 combustible bundles", the procedure that includes all the conditions and stages that must be fulfilled during the activities of the Fuelling Machine (F/M) Head Test Rig.

Based on the TECHNICAL PROVISION, the below logic diagram was developed with the aim of implementing the prototype version of the expert system (see Fig 4). By using decision trees, all the alternatives can be graded in detail, depending on all the possible options, finally obtaining the results associated with each possible course of alternatives.

The ability of VisiRule software to visually develop decision trees and its technical attributes, according to requirements necessary for the development of the expert system based on integrated knowledge as a decision support for Fuelling Machine (F/M) Head Test Rig operators, were the main factors in the selection for the expert system implementation.

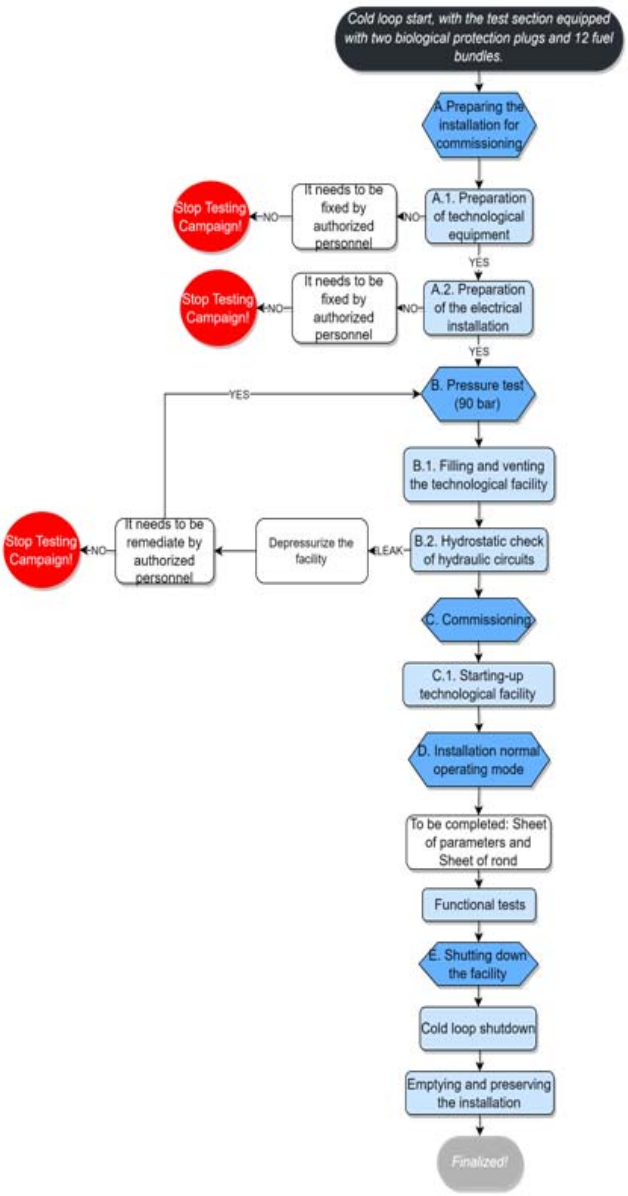


Fig. 4. Logical diagram

VisiRule decision tree

VisiRule is a graphical tool that generates executable backward chaining rules. In the Flex KSL code, the rules are denoted by the word "relation". These are often labelled as goal-driven. The logic-based engine supports complex logic that can be used to solve complex problems in a way comparable to human experts.

In the VisiRule software, the decision tree was developed based on the logic diagram shown in Figs. 5, 6 and 7. This expert system application, based on the decision tree, creates a database in an XML file (Extensible Markup Language) and includes all the procedures introduced in the branches and nodes of the tree. It can be updated and expanded, but the model remains the same throughout the life of the system.

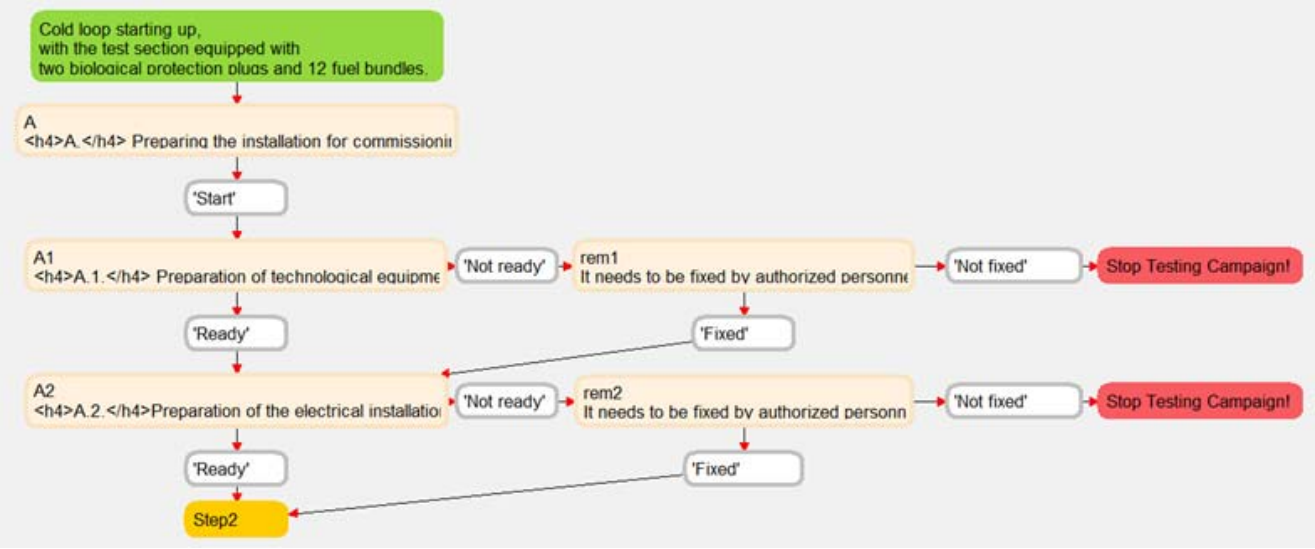


Fig. 5. Decision tree - step 1

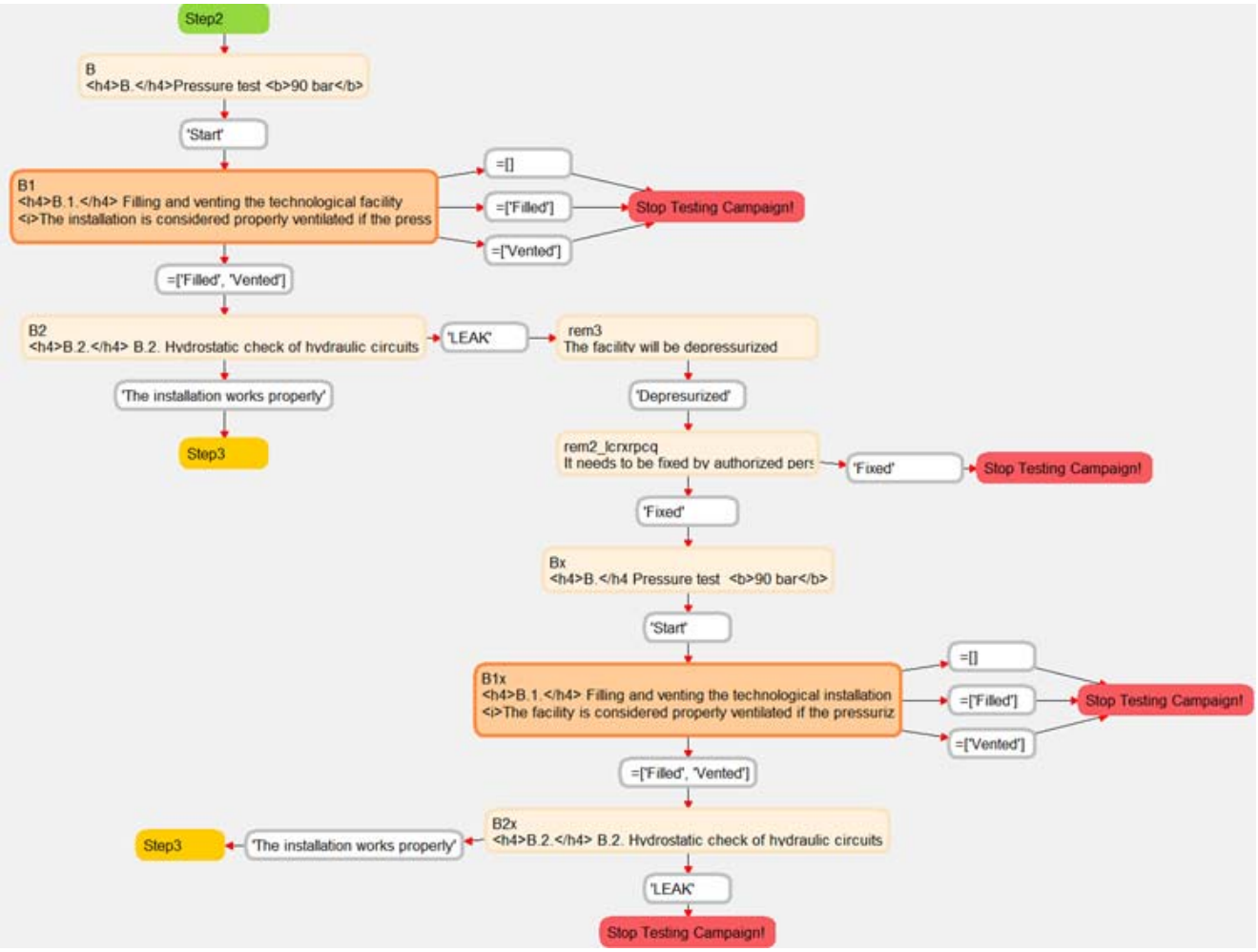


Fig. 6. Decision tree - step 2

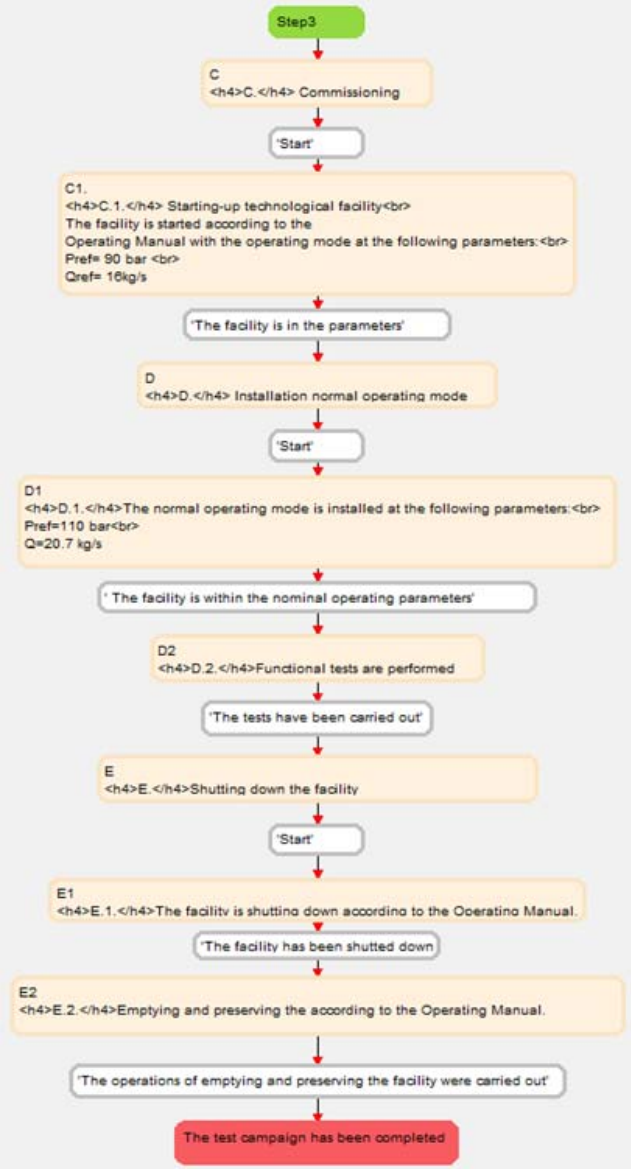


Fig. 7. Decision tree - step 3

Based on the decision tree, the application creates a script in Flex KSL language that can be seen for Step 1 in Fig. 8, and constitutes the inference mechanism that interprets and determines, based on the interaction with the user, which data are needed from the knowledge base.

This script works based on the relationships between "Questions" and "Answers". Each contains an ordered set of steps that must be used to accomplish a specific task. These could be seen as a scenario of the events taking place.

VisiRule allows the export of Flex KSL code in HTML format, a format that will be used in the end as part of the user interface. By generating the Flex KSL code in HTML format, the inference mechanism keeps its mode of operation, but is implemented in JavaScript language.

The user interface has been developed in HTML (HyperText Markup Language) and CSS (Cascading Style Sheets) and is also connected with an explanatory module - explanation generation subsystem, which, guided by the inference engine, allows the user to follow the path to reach to a solution.

```
do ensure_loaded( system(vrlib) ) .
relation 'Cold loop starting up, with the test section equipped with two
biological protection plugs and 12 fuel bundles.'( Conclusion ) if
  q_A( Conclusion ) .

relation q_A( Conclusion ) if
  the answer to 'A' is _ and
  check( 'A', =, 'Start' ) and
  q_A1( Conclusion ) .

relation q_A1( Conclusion ) if
  the answer to 'A1' is _ and
  check( 'A1', =, 'Ready' ) and
  q_A2( Conclusion ) .

relation q_A1( Conclusion ) if
  the answer to 'A1' is _ and
  check( 'A1', =, 'Not ready' ) and
  q_rem1( Conclusion ) .

relation q_A2( Conclusion ) if
  the answer to 'A2' is _ and
  check( 'A2', =, 'Ready' ) and
  Conclusion = 'Step2'  .

relation q_A2( Conclusion ) if
  the answer to 'A2' is _ and
  check( 'A2', =, 'Not ready' ) and
  q_rem2( Conclusion ) .

relation q_rem2( Conclusion ) if
  the answer to rem2 is _ and
  check( rem2, =, 'Fixed' ) and
  Conclusion = 'Step2'  .

relation q_rem2( Conclusion ) if
  the answer to rem2 is _ and
  check( rem2, =, 'Not fixed' ) and
  Conclusion = 'Stop Testing Campaign!'  .

relation q_rem1( Conclusion ) if
  the answer to rem1 is _ and
  check( rem1, =, 'Fixed' ) and
  q_A2( Conclusion ) .

relation q_rem1( Conclusion ) if
  the answer to rem1 is _ and
  check( rem1, =, 'Not fixed' ) and
  Conclusion = 'Stop Testing Campaign!'  .

group group1
  'Start' .
question 'A'
  '<h4>A.</h4> Preparing the installation for commissioning' ;
  choose one of group1
  because " .

group group2
  'Prepared', 'Not ready' .
question 'A1'
  '<h4>A.1.</h4> Preparation of technological equipment' ;
  choose one of group2
  because " .

group group3
  'Remedied', 'Not fixed' .
question rem1
  'It needs to be fixed by authorized personnel' ;
  choose one of group3
  because " .

group group4
  'Remediated', 'Not fixed' .
question rem2
  'It needs to be fixed by authorized personnel' ;
  choose one of group4
  because " .

question 'A2'
  '<h4>A.2.</h4>Preparation of the electrical installation' ;
  choose one of group2
  because " .
```

Fig. 8. Flex KSL source code for step 1

The diagrams are delivered to the user in the form of questions in HTML format shown in Figs. 9, 10 and 11. JavaScript engine asks questions and generates advice and conclusions extracted from the knowledge base.

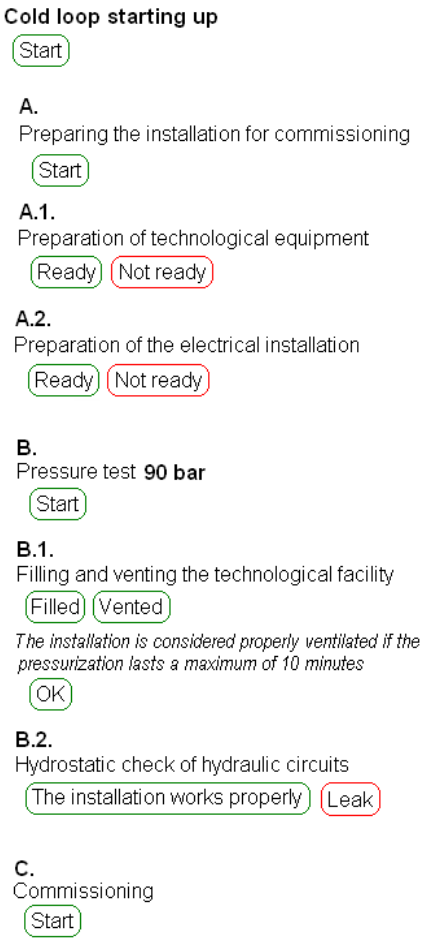


Fig. 9. User interface - step 1

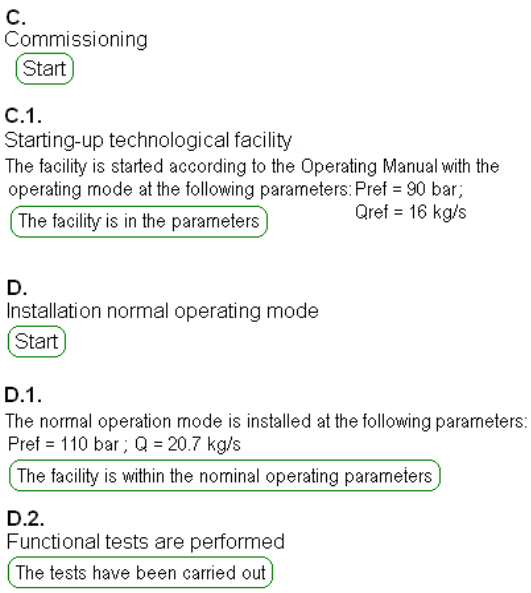


Fig. 10. User interface - step 2

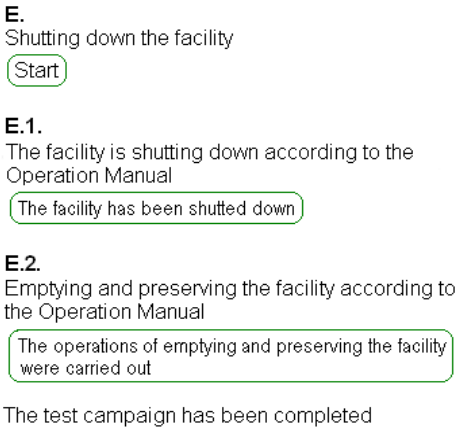


Fig. 11. User interface - step 3

Conclusions

In order to capitalize the results obtained by achieving the objectives of the work, it is necessary to continue the development of the expert system prototype by including all the issues related to the operation of the Fuelling Machine Head Test Rig.

Due to the complexity of the operating system, the knowledge for the task may not be adequately represented, and there is a need for further interaction and discussion with the Fuelling Machine Head Test Rig operators who are willing to share their expertise.

The training of personnel in the nuclear field is known to be vital, and the expert system includes a large volume of the experts' knowledge in the field and is kept safe, thus allowing the proper training of the operating personnel.

Therefore, it must be highlighted the importance of one of the research objectives mentioned earlier in the paper, namely, development of a virtual assistant to support the operator in decision-making and provide solutions for the administration and management of systems in order to increase efficiency and operational safety.

References

[1] Badescu, L. (2015) "Operating manual for the MID Loop", INR internal report, RI-10761, Institute for Nuclear Research, Pitesti, Romania (available in Romanian).
[2] Buchanan, B.G., Duda, R.O. (1983) "Principles of Rule-Based Expert Systems", *Advances in Computers*, Vol. 22, pp. 163-216.
[3] Kartam, N., Flood, I., Tongthong, T. (1995) "Integrating knowledge-based systems and artificial neural networks for engineering", *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, Vol. 9(1), pp. 13-22.
[4] Buydens, L., Van Leeuwen, H., Wehrens, R. (1993) "Expert System-Development tools", *Data Handling in Science and Technology*, Vol. 13, pp. 121-151.