

SYSTEMS & CONTROL | RESEARCH ARTICLE

QNP_SHELL: A computerized tool for improving decision-making skills for nuclear power plant operators

Hassan Qudrat-Ullah

Cogent Engineering (2015), 2: 1030829



Received: 24 August 2014
Accepted: 13 March 2015
Published: 09 April 2015

*Corresponding author: Hassan Qudrat-Ullah, School of Administrative Studies, York University, 4700 Keele Street, Toronto, Ontario, Canada M9V 3K7.
E-mail: hassnq@yorku.ca

Reviewing editor:
Zude Zhou, Wuhan University of Technology, China

Additional information is available at the end of the article

SYSTEMS & CONTROL | RESEARCH ARTICLE

QNP_SHELL: A computerized tool for improving decision-making skills for nuclear power plant operators

Hassan Qudrat-Ullah^{1*}

Abstract: Decision-making in complex systems such as nuclear power plants (NPPs) is a difficult task at best. The safety and integrity of many such high-capital cost-intensive installations depend on the operator's capability to correctly diagnose and take appropriate measures to avoid any abnormal operations of an NPP. Therefore, the role of the expert systems in the offline training programs for the operators is ever increasing. In this paper, we describe the development of an expert system, "QNP_SHELL," to assist, offline QNPP operators and plant personnel in a better familiarization to infer the anticipated and foreseen malfunctions from the observed symptoms. QNP_SHELL's inferencing mechanism is of the "Rule-based" type and to search the knowledge base it adopts the "Depth First" technique. The diagnostic performance of the trainee operators using QNP_SHELL on various accidents at QNPP has been found, through both the qualitative and quantitative evaluations, satisfactory.

Subjects: Information / Knowledge Management; Management Education; Operations Management

Keywords: nuclear power plants; operators; training; expert system; fault diagnosis



Hassan Qudrat-Ullah

ABOUT THE AUTHOR

Hassan Qudrat-Ullah is an associate professor of management science, at the School of Administrative Studies, York University, Canada. He has over 18 years of university teaching, research, and consulting experience in the US, Canada, Singapore, Norway, UK, Korea, China, Saudi Arabia, Switzerland, and Pakistan. His research focuses on: How people can make better decision in complex tasks? He has published 17 journal articles, 3 books, 8 book chapters, and over 40 papers in refereed conference proceedings. He is an editor-in-chief of *International Journal of Complexity in Applied Science and Technology*. His latest book, *Better Decision Making in Complex, Dynamic Tasks* (Springer: 2014), focuses on the design, development, and applications of computer simulation-based decision support systems. The research reported here, relating to his research on the thematic area of decision-making in complex tasks, describes the development and utility of an expert system for the training of nuclear plant operators.

PUBLIC INTEREST STATEMENT

Decision-making in complex systems such as nuclear power plants is a difficult task at best. The safety and integrity of many such high-capital cost-intensive installations depend on the operator's capability to correctly diagnose and take appropriate measures to avoid any abnormal operations of a nuclear power plant. Therefore, training of nuclear power plant operator takes the center stage among the possible initiatives for the safe and successful plant operations. It is imperative that training for this kind of complex tasks needs to be specific and focused to the needs of the particular power plant. How should we train these operators? Various tools and programs are used for this purpose. Use of computerized decision support is common. Among such tools, the knowledge-based expert systems are particular useful for the fault diagnosis training tasks. In this paper, we describe the design, development, and use of an expert system, "QNP_SHELL."

1. Introduction

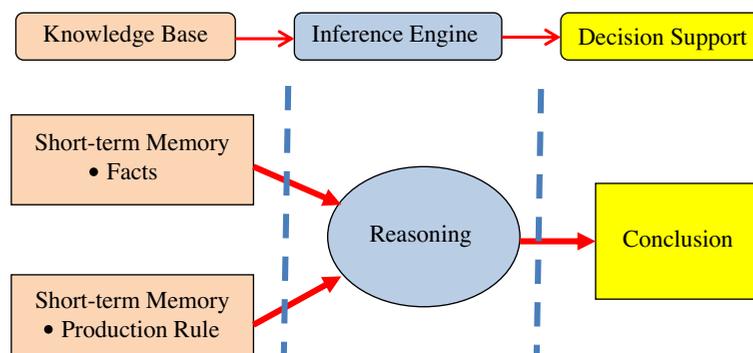
Fault diagnosis is a critical skill required for operators of high-stakes complex systems such as nuclear power plants (NPPs). To prevent accidents in NPPs, the operators are required to detect early signs of potential abnormal operations. It requires not only better understanding of operations of the NPP, but also agile diagnostic skills and judgmental decision-making abilities. To develop such skills, expert systems are extensively used in their training programs and protocols (Abu-Khader, 2009; Heo, Chang, Choi, Choi, & Jee, 2005; Lee, 2002; Moshkbar-Bakhshayesh & Ghofrani, 2013; Najdawi, Chung, & Salaheldin, 2008; Naser, 1990; Santhosh et al., 2011). On the development and utility of expert systems, the interested readers are referred to Buchanan’s comprehensive review (Buchanan, 1986; Ma & Jiang, 2011). The management of QNPP,¹ a 300 MWe power plant in the category of SMRs (Locatelli, Bingham, & Mancini, 2014), in an effort to develop indigenous capabilities, commissioned this project to develop an expert system for the training of its operators for fault diagnosis of QNPP.

There are many methods available for fault detection and diagnosis in NPPs (Liu, Xie, Peng, & Ling, 2014). We can categorize them into two major types of methods that are applied for transient identification in NPP. They are: (i) comprehensive data-driven techniques [e.g. artificial neural networks, fuzzy logic-based systems, and heuristic techniques (Yong-kuo, Min-jun, Chun-li, & Ya-xin, 2013)] and (ii) model-based methods [e.g. hard computing intensive mathematical models (Angeli, 2010)]. While a fairly large number of applications of data-driven systems have been successfully applied, the application of model-driven systems is very limited [an excellent comparative review is presented in Ma and Jiang (2011) and Moshkbar-Bakhshayesh and Ghofrani (2013)]. In particular, the ability of heuristic techniques not to suffer from the local minima problem makes them suitable for the development of fault diagnosis system. Therefore, consistent with our objective “to develop an expert system for the off-line training of QNPP’s operators on fault diagnosis,” we adopted the rule-based approach. Figure 1 presents the basic structure of a rule-based expert system, the production system model for problem-solving/decision-making (Newell & Simon, 1972). Compared with traditional mathematical and numerical approaches for fault diagnosis, a rule-based expert system:

- (i) processes knowledge expressed in the form of rules and use heuristics to arrive at the conclusion,
- (ii) clearly separates knowledge from the processing program (i.e. inference engine),
- (iii) traces fully the chain of reasoning behind any conclusion reached and data-set used,
- (iv) can deal with incomplete, uncertain, and fuzzy data, and
- (v) provides flexibility to change or add new knowledge (Sydenham & Thorn, 2005). In-house availability of knowledge engineers provided additional incentive for the adoption of the rule-based approach.

In this paper, we contribute with the design, development, and assessment of such a training tool, QNP_SHELL. The main objective of the development of QNP_SHELL is to augment QNPP operators’ and plant personnel’s training by making expert systems available to them to assist them offline, to

Figure 1. The production rule-oriented problem-solving model.



become more familiar with anticipated and unforeseen transients of the plant. The principal benefit attained will be the improved productivity of fault diagnostic types of expert systems. Instead of buying an off-the-shelf expensive solution (e.g. VP-expert) or developing various expert systems to assist on offline basis in diagnosing various types of faults, a general inferencing mechanism, independent of any domain knowledge (Francis Cheong Yiu Fung, 1989; Vinod, Babar, Kushwaha, & Raj, 2002), we have designed, developed, and tested an expert system shell, QNP_SHELL.

To utilize this problem-independent expert system shell, the knowledge about various faults to be diagnosed is to be organized and coded into various files. The QNP_SHELL can then be directed during a consultation session, to access knowledge from the specified files. Therefore, the same QNP_SHELL can be used to study different types of diagnostics by directing it to consult different knowledge base files. This use of QNP_SHELL facilitates the diagnostic study and, as a result, various types of fault diagnosis are made readily accessible to the plant personnel to become better familiar with the plant transient and fault diagnoses. The utilization of the developed QNP_SHELL will help in reducing the cost and man hours on the design and development of expert systems to assist plant personnel in offline fault diagnoses (Naser, 1990; Negnevitsky, 2005). The efficiency is achieved by cutting the job down to the only requirement of coding and organizing the knowledge of specific domains into files.

In Section 2, we describe the methodology including (i) program structure and (ii) system description. The knowledge base development scheme is presented in Section 3. Section 4 presents the testing and evaluation of the developed QNP_SHELL. Finally, Section 5 concludes this paper with the highlights on potential future research.

2. Methodology

2.1. A brief account of relevant theoretical and practical perspectives

Fault diagnosis in technical systems such as NPPs in general and the design, development, and use of expert systems in particular have received considerable theoretical and practical attention over the past several decades (Angeli, 2010; Li, Upadhyaya, & Perillo, 2012; Locatelli et al., 2014). In expert systems research, regardless of the purpose of an expert system (e.g. in our case the purpose is the fault diagnosis), the fundamental organization principle is the separation of the knowledge base and the program structure (i.e. inference engine) (Buchanan, 1986). In the design process of an expert system, this fundamental principle not only results in a flexible and modular expert system, but also enhances the productivity of designers/builders of expert systems—they don't have to worry about the development of different data structures and algorithms.

In fact, rule-based expert systems have been successfully applied in fault detection, monitoring, and diagnosis of complex systems such as NPPs (Abraham, 2005; Abu-Khader, 2009; Angeli, 2010). These expert systems can explain the reasoning process (e.g. what chain of symptoms leads to a particular fault of a NPP) and deal with uncertainty, which traditional computational methods do not handle (Abraham, 2005). There are three core components of a rule-based expert system—program structure, knowledge base, and user interface (with a dedicated explanation facility).

The program structure of an expert system acts like a brain—handles users' queries and provides casual explanation of events in an effective and efficient way. Two types of inferencing mechanisms are used. Forward chaining process begins with the facts and works forward to arrive at a conclusion. Backward chaining is the process that starts with conclusions [e.g. a specific accident like loss of coolant accident (LOCA) has happened] and works backward to identify the relevant chain of facts (Abraham, 2005; Angeli, 2010; Yong-kuo et al., 2013). QNP_SHELL is based on both the forward and backward chaining processes.

The development of the knowledge base is another critical task in the development of an expert system. The most common data structures generally used for symbolic representations are production rules, semantic networks, frames, predicate logic, and hybrid (Angeli, 2010). With the objective

of achieving a relatively higher level of effectiveness and efficiency in the knowledge base of the QNP_SHELL, we have adopted a hybrid approach where production rules and predicate logic are our main data structures. The selection of knowledge acquisition and representation methods primarily depends on the purpose of the expert system (Naser, 1990; Liu et al., 2014). The knowledge acquisition process includes various techniques including interviews, protocol analysis, repertory grid analysis, and automatic induction techniques (Angeli, 2010). However, when it comes to elicitation of expert knowledge, interviews, although time-consuming, are considered as the most popular and widely used form of expert knowledge acquisition (Jiang, 2008; Milton, 2007). Given the specialized and experiential nature of the knowledge of the experts, a knowledge engineering team at QNNP employed interviews as well as utilized the existing archival data to acquire knowledge for the knowledge base of the QNP_SHELL. For knowledge representation, an exhaustive but reliable set of production rules is developed.

Finally, besides the program structure and the knowledge base, a user interface that allows the operators or users of any expert system to interact with its inference engine is an equally critical component of an expert system. The ecological interface design (Vicente, 2002) and HCI design principles (Howie, Sy, Ford, & Vicente, 2000) provide sound theoretical basis for the development of the user interface of any decision support system albeit an expert system. Drawing on HCI design principles, the developed user interface of QNP_SHELL:

- allows user to interact with the QNP_SHELL in a systematic and flexible manner,
- and provides the rich causal explanation of the user's actions.

In the following subsections, both the development and the operational details of the program structure (i.e. inference engine), knowledge base, explanation facility, and user interface of the QNP_SHELL are presented.

2.2. Program structure

A modular approach has been adopted during QNP_SHELL program development (Angeli, 2010; Waterman, 1986). The modules developed and incorporated in the design and development of QNP_SHELL can, according to their functionality, be classified as modules that make up the inference engine, the user interface, and the explanation facility. The QNP_SHELL is problem independent and can be utilized for performing various types of diagnoses. The only restriction is that the information should be coded and organized according to a specific format. The programming language used is PROLOG (Townsend, 1989). Figure 1 presents the functional schema of QNP_SHELL.

2.2.1. The inference engine

Figure 2 presents the schematic of the inference engine of QNP_SHELL. The major modules that constitute the inference engine are (i) a top most repetitive loop, which controls the program execution, (ii) the main diagnostic module that decides the control strategy, invokes the various file manipulation modules and the inferencing modules, and (iii) the inferencing modules that process various inference conditions associated with a fault. To process each different type of inference condition, a different inferencing module is called upon. The invoking of the various inferencing modules is to be determined by the main diagnostic module and the file manipulation modules. These inferencing modules scan the information stored in different files and extract the necessary information as and when required by the main diagnostic module and the information-processing modules including the conflict resolution module. Moreover, these inferencing modules acknowledge the user's response to various queries as well as to the selection of an option from the multi-option list. Additionally, these modules update the dynamic database.

The inference engine accesses the knowledge base in a particular sequence by entering into a dialog with the user (Figure 3 presents its schematic diagram). It attempts to infer new information based on the user's response. It then uses these facts and attempts to match them with the information stored in the knowledge base files. If all the inferences about the existence of a fault stand

Figure 2. Functional block diagram of QNP_SHELL.

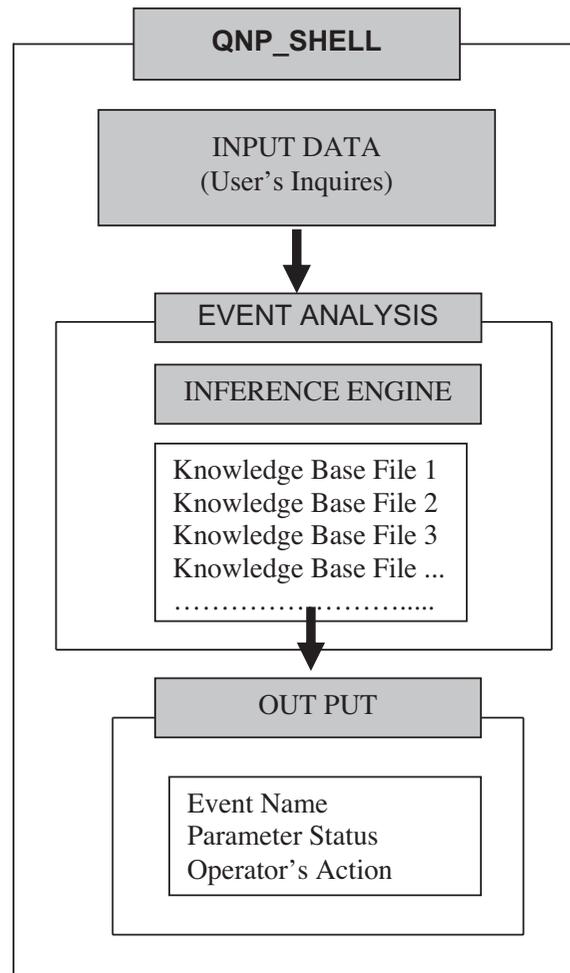
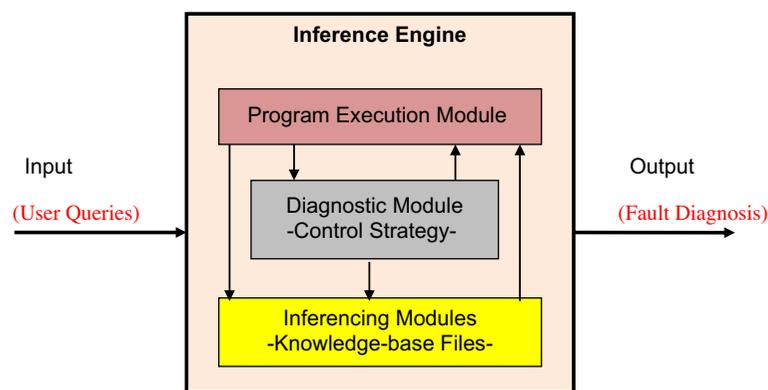


Figure 3. Schematic of the inference engine of QNP_SHELL.



true, then the concluding module decides whether the existence of the fault under consideration can be claimed with certainty or its existence can only be suspected. If the inference test to a fault fails then the next fault is selected for analysis.

2.2.2. User interface

To enhance the user friendliness of the system, a minimum of typing has been demanded of the user. Wherever possible, the available choices have been displayed as multiple options through pop-up windows.

Independent modules have been employed to control the prompt of various queries on the monitor screen and to acknowledge the user's response. Whenever the user's response is an affirmation to a query, the string-processing modules are automatically invoked. These modules transfer the query into a grammatically correct sentence. The latter is then displayed on the monitor screen in the form of symptoms. Another independent module caters for alarm generation and their display on the monitor screen.

2.2.3. *Explanation facility*

At the user's demand for an explanation to the current state of diagnosis, an independent module is invoked. This module displays the required information (e.g. an accurate causal chain from systems to the respective fault) to the user—a *true learning and knowledge building facility*.

2.3. **System description**

2.3.1. *The execution of the QNP_SHELL*

On the execution of the QNP_SHELL, a configuration setup module is invoked. The module either by reading a configuration file or through an interactive session with the user, selects:

- (i) The knowledge base files to be consulted during the problem diagnosis.
- (ii) The dir wherefrom the knowledge base files are to be accessed.

2.3.2. *The configuration setup*

Once the user has agreed to the files and to the disk/dir, the configuration setup module disengages itself and a repetitive loop is invoked. This repetitive loop performs the diagnosis by executing the remainder of the program for any number of consultations. The repetitive loop remains intact as long as the user's response is in the affirmative to the query that is put to him at the end of each consultation.

2.3.3. *Initialization and fault diagnosis process*

For every execution of the repetitive loop, the necessary variables are initialized and the main diagnostic module is invoked. The main diagnostic module in turn, and at first, invokes the problem set consulting module. The latter makes its access to the file that contains the problem set and reads a term in serial order. The fault as well as the associated list of inferences and confirmations is passed on to the main diagnostic module. The latter sequentially accesses the various condition sets that constitute the list of inferences and passes them to the inferencing mechanism for further analysis. The inferencing mechanism automatically caters for invoking the various inferencing modules depending upon the type of condition set under process. To analyze each different type of condition set, a separate module is triggered by the inferencing mechanism. Based upon a user's responses to the queries contained in a condition set, the inferencing mechanism determines the diagnostic path to be followed.

2.3.4. *Backtracking feature*

If during the analysis the inferencing mechanism negates the possibility of existence of a fault, then backtracking occurs. It automatically picks up the term that lies next in serial order and the above analyses are repeated.

2.3.5. *The confirmatory test*

If the inferences associated with a fault are true, then the fault under consideration is considered to have been diagnosed. At this stage, the main diagnostic module finally invokes the confirmatory test module. This module attempts to match the facts with the information in the knowledge base files. If an exact pattern matching occurs, the diagnosed fault is displayed on the monitor screen along with a message of certainty to its existence. If, however, an exact pattern matching is not observed, then the display of the fault is accompanied by a message of suspicion about its existence.

3. Knowledge base development

3.1. The overall functional architecture

A maximum of five files are required to be developed for use by the QNP_SHELL. A configuration file through which the QNP_SHELL reads the file names, where the user has stored the knowledge base. Four other files are required to be developed. These files contain the knowledge base according to the format as specified in the following sections. Depending on the nature of the domain on which the knowledge base is required, the users of the QNP_SHELL will have to develop all of the four or at least the first two of the following files.

- (1) A file containing the PROBLEM SET. Each term of this file contains a problem to be diagnosed and the associated dependencies that lead to affirmations or negation to the existence of the problem. This file is the heart of the knowledge base. Its development cannot be skipped, no matter on which domain the knowledge base is developed.
- (2) A file containing the queries to be put to the user during problem analysis. Each term of the file contains a query along with two switches, (i) option/no_option and (ii) alarm/no-alarm. These switches indicate or deny the presence of multiple options and alarms/warnings associated with each query. The associated multiple options will pop up on the monitor screen if the user has affirmed in response to the query. The user will then have to select one of the options. Likewise, the associated alarms will be displayed on the monitor screen, if the user has affirmed to the query. This file is one of the two files of primary importance and cannot be skipped.
- (3) A file containing multiple options for all such queries for which the OPTION switch has been set ON by the knowledge engineer.
- (4) A file containing alarms/warnings associated to all those queries as well as to the multiple options for which the ALARM switch has been set ON by the knowledge engineer.

3.2. Operational description

During its operation, the QNP_SHELL first makes its access to the file containing the problem set. It then picks up a problem, in serial order along with the condition-dependent sets of query numbers. The program then performs analysis on the condition set in a sequential order. The success or failure of a condition set depends on the nature of the set as well as the user's response to queries, whose numbers are contained in the set. If a condition set is at success, the system will take up the next condition set. If all the condition sets associated with a problem are at success, the problem is considered to be diagnosed. However, the system proceeds further to decide about the confirmation or suspicion to the existence of the problem. If a condition set fails, then the system may take up the next set or it may discard the possibility of the existence of the problem under consideration. While processing a condition set, the program sequentially picks up a query number from the list contained in the condition set and reads the relevant query from the file containing the primary queries. The query is then displayed and the user is supposed to respond accordingly. If the user negates the query, then depending on the type of the condition set, any one of the following situations may occur:

- (1) A query with the next query number from the list may be displayed for the user's response.
- (2) The program may move over to the next condition set for the same problem.

The program may discard the current problem and may proceed to the next problem in the problem set. If, however, the user affirms to the query, then depending on the ON/OFF conditions of the alarms and multi-option switches, the alarms and/or options may or may not be read from appropriate files.

3.3. File format for knowledge base development

3.3.1. Format for “Diagnstc.kba” containing the problem set

The file, “Diagnostics.kba” lists all the problem sets in the following format:

faults (Sno,[A list containing a problem to be diagnosed], [A complex list structure containing the primary condition sets], [A complex list structure containing the confirmation sets]).

As an example, we present the coding scheme for the diagnostic of a problem: Is charging tank lever is decreasing? The command syntax is:

faults(2,["Is Charging Tank Level Decreasing"], [and_cond([1,5]), or_cond([2,6]), and_cond([3,10])be_or([1,2,3])], [must_be([1.2,3.2,4.0]), must_not([2.3,2.0])]).

The description of this command's variables is as follows:

(i) Sno

A unique integral number associated with each of the problems diagnosed (e.g. 2 in this example). The problems in the “Diagnstc.kba” need not be arranged in an ascending/descending order of Sno.

(ii) A list containing a fault or a group of faults to be diagnosed

The problem to be diagnosed may contain a single statement describing the problem or it may be a multi statements problem. Each single statement is to be considered as an element of the list. The only limitation is that the number of characters in each statement should not be more than 76.

(iii) A complex list structure containing the condition sets

The third variable is a complex list structure. Any one, or all, or any combination of the following conditions can be used. Each of the following condition sets should contain a list of query number as its first value. An empty list is also a valid entry. The following are some examples of the permissible condition sets along with their description:

(1) AND condition

Its format is: and_cond ([A list of integral numbers]). For example,

and_cond([1,15,10,15]).

If any query is associated with the query number contained in the list and the user has responded with negation, the “and_cond” will fail. This means that the problem to which the “and_cond” is associated has been discarded from analysis. For any of the “and_cond” set to be true, all the queries in the list must be affirmed.

(2) OR condition

The format of this condition is: or_cond ([A list of integral numbers]). For example,

or_cond([1,2,6,9]).

It means, no matter what the user has replied, the system will proceed to process the next query number contained in the “or_cond” set. The decision will be taken after analyzing the complete list contained in the “or_cond” set. If the user's response to all the question numbers contained in the list was in negative, then the “or_cond” will fail, which means that the problem to which the “or_cond” is associated has been discarded from analysis. For an “or_cond” to be true, at least one of the queries associated with the corresponding query numbers in the list must be affirmed.

(iv) A complex list structure containing the confirmation sets

The fourth variable of the command is also a complex list structure. An empty list is a valid entry.

3.3.2. Format for "PRIM.KBA" containing the primary queries

The following is the general format for of this file:

prim (Qno,[List containing a query], Option_choice, Alarm_choice)

As an example, we present the command syntax of a query, "Is there an increase in temperature?":

```
prim(1,["Check the Channel Temperature recorder", "Is there an increase in temperatures"],  
"no_option", "alarm")
```

The description of this command's variables is given below.

(i) Qno

A unique integral number associated with each of the queries to be put to the user.

(ii) List containing a query

A string list containing a query along with its description, if required, is to be inserted for the user's information. The following conditions must be met during file development:

- The query to be put to the user must be the last element of the list.
- An individual element of the list must not contain more than 77 characters.

(iii) Option choice

Option_choice = "option"/"no_option"

A flag, associated with each query which indicates or denies the presence of further options corresponding to the query. In the "PRIM.KBA" file, if the flag "option" has been associated with none of the queries, then the file "SEC.PRO" containing the secondary options need not be developed.

(iv) Alarm choice

Alarm_choice = "alarm"/"no_alarm"

A flag associated with each query which indicates or denies the availability of alarms associated with the primary query, irrespective of the fact whether secondary options are associated with the query or not. If none of the queries in "PRIM.PRO" has been associated and none of the secondary options is associated, the file "ALARM.PRO" containing the alarms need not be developed.

3.3.3. Format for SEC.KBA containing the secondary options

The file, "SEC.KBA" deals with the secondary options. The format of this file is as:

sec (Qno, Information-description, [A list of options], alarms_for ([A list containing the option numbers to which alarms are associated]))

Here is an example of this command syntax.

```
sec(2, "The charging tank level decrease is", ["fast", "gradual", "slow"], alarms_for([1,2]))
```

The variables of this command syntax are described below.

(i) Qno

The same unique query number (an integer) which is associated with the corresponding query in "PRIM.KBA".

(ii) Information description

This is a string providing explanation of the options which are to be selected by the user from the following defined list.

(iii) A list of options

A list of options that will be displayed to the user and the user will have to select one. The number of characters contained in the option list must be such that the total length of the string should not exceed 76 characters.

(iv) Alarms for ([A list containing the option numbers to which alarms are associated])

The term “alarms_for ([...])” contains a list of integers corresponding to the nth option contained in the option list to which certain alarm/warning is associated. If the alarm is not associated with any of the secondary options, the knowledge engineer will have to enter an empty list “[]” within the term “alarms_for(...)”.

4. Testing and preliminary evaluation of QNP_SHELL

For the utility and performance assessment of QNP_SHELL, we applied a multi-method approach—in addition to the application of the Turing test, the performance of the trainee operators was assessed through statistical validation procedures.

4.1. Applying the Turing test

4.1.1. Selection of the testing event

LOCA is the worst kind of accident scenario that one can expect in a NPP (Apostolakis, Kafka, & Mancini, 1988). LOCA accident occurs where there is a leak or break in the primary coolant loop of a PWR reactor such as our QNPP, a 300 MWe plant. If LOCA accident is not identified and mitigation actions are not taken in time, catastrophic consequences including core melt and release of radioactivity to the atmosphere (if containment systems also fails) will happen. Therefore, we decided to test QNP_SHELL on LOCA. The information on LOCA diagnostics, from the written form of the standard procedures, as made available by QNPP management, was coded and organized into four different files. These files were then utilized to have a demonstration of the working QNP_SHELL.

4.1.2. Performance evaluation of QNP_SHELL on LOCA at QNPP

The purpose of this expert system is to advise you, the user, to identify LOCA types in a PWR, when high-pressure injection signal alarm appears in the control room of QNPP.

4.1.2.1. Event definitions based on manual procedures of QNPP.

- 3.1 LOCA-1 is medium LOCA in the main loop (QMP 14.21.2)
- 3.2 LOCA-2 is leakage due to pressurizer safety/relief valve open (inadvertently) (QMP 14.23.1)
- 3.3 LOCA-3 is leakage due to steam generator (SG) tube rupture (primary leak into secondary side) (QMP 14.31)
- 3.4 LOCA-4 is leakage due to steam line break outside the containment building (QMP 14.11)
- 3.5 LOCA-5 is leakage due to steam line or feedwater line (inside the containment building) without SG tube rupture (QMP 14.6-14.10)
- 3.6 LOCA-6 is leakage due to steam line or feedwater line (inside the containment building) with SG tube rupture (QMP 14.21.4)
- 3.7 LOCASG is an additional logic signal to confirm one of the above break types.

Please note that corresponding to each of above-defined LOCA types, there are separate procedures for safety actions (called “QMP” Manual Procedures in this PWR), which the operator is required to follow after confirming the leakage type. For the benefit of the user, the QMP procedure numbers are also given in the expert system with each break definition.

4.1.2.2. User's queries within QNP_SHELL expert system.

ASK	HPINSG:	"Is there high pressure injection signal?"
CHOICES	HPINSG:	yes, no
ASK	PRLVL:	"Is pressurizer level <2.85 m?"
CHOICES	PRLVL:	yes, no
ASK	PPCL:	"Is pressure in primary coolant loop <110 bar?"
CHOICES	PPCL:	yes, no
ASK	PDECA:	"Is diff. pressure of equipment compartment w.r.t. atmosphere >30 mbar?"
CHOICES	PDECA:	yes, no
ASK	NASG:	"Is N16 activity behind SG 1 or 2 becomes >limit?"
CHOICES	NASG:	yes, no
ASK	NBSG:	"Is N16 activity behind SG 3 or 4 becomes >limit?"
CHOICES	NBSG:	yes, no
ASK	DIRIC:	"Is direct ionizing radiation level (inside containment > limit?"
CHOICES	DIRIC:	yes, no
ASK	PRLRT:	"Is pressure level in pressurizer relief tank >12 bar?"
CHOICES	PRLRT:	yes, no
ASK	PRSVON:	"Is pressure safety/relief valve open?"
CHOICES	PRSVON:	yes, no
ASK	LOCASG:	"Is corresponding LOCASG logic alarm signal available?"
CHOICES	LOCASG:	yes, no

4.1.2.3. Rules formulations within the QNP_SHELL expert system (only few rules are presented here as an example).

Rule 1	IF	HPINSG = yes AND PDECA = yes AND NBSG = yes AND PRLRT = yes
	THEN	LOCA type is LOCA-1 (Ref. Definition 3.1, above)
Rule 3	IF	HPINSG = yes AND PDECA = yes AND NBSG = yes AND PRLRT = yes AND PRSVON = yes
	THEN	LOCA type is LOCA-2 (Ref. Definition 3.2, above)
Rule 11	IF	HPINSG = yes AND PDECA = yes AND DIRIC = yes AND LOCASG = yes
	THEN	LOCA type is LOCA-6 (Ref. Definition 3.6, above)

In a computer-simulated LOCA accident condition, operators using QNP_SHELL expert system were able to identify the type of LOCA correctly. Not only were these operators able to run down the plant to "cold shutdown and depressurized" condition, a required condition as specified by the emergency procedures of QNPP, their dynamic behavior during the control of the accident was judged as

“excellent” by the group of experienced plant operators. Therefore, the performance of the QNP_SHELL has been found satisfactory by the experienced plant operators at QNPP, through a procedure known as the Turing test (Spring, 1993).

4.2. Statistical validation of QNP_SHELL

The utility and effectiveness of QNP_SHELL on the diagnostics performance of its users were also assessed on a range of events through a quasi-experimental manner. The management of QNPP has an intensive training program for the operators. In a six-week modular program, it includes classroom instructions based on QNPP’s operating manuals (three weeks), exercises and written tests (one week), and simulator-based fault diagnostic practice and learning (two weeks). Compared with the group of trainee operators who did not have access to QNP_SHELL (i.e. prior to the induction of QNP_SHELL), this new group with access to QNP_SHELL had only one week of classroom instructions. The performance of both groups was measured on their ability to correctly identify the faults. First, they had to do a written test consisting of multiple-choice questions on the symptoms of the fault and then their performance was tested on QNPP’s simulator that fully replicates QNPP. All the participants were tested on seven events, as listed in Table 1. The order of the events was random. The diagnostic accuracy of the trainee operators, assessed through the written tests, is the same for both groups. Although the total number of errors (i.e. how many faults were not correctly identified by the trainee operator) by Group 1 (those without QNP_SHELL) was higher (i.e. eight versus four) than Group 2 (with QNP_SHELL), the overall diagnostics performance on the seven events did not differ statistically ($F = 1.61$; $p = 0.21$). We ran several t -tests for all the events separately and found no statistical difference between both groups (please see the t -test results in Tables 2 and 3). Since all the trainee operators correctly identified LOCA, the statistical testing for LOCA was not required.

Table 1. Trainee participants and list of diagnostic events

	Group 1 (without QNP_SHELL)	Group 2 (with QNP_SHELL)	Diagnostic performance events
<i>N</i>	15	13	LOCA: loss of coolant accident FLBA: feed line break of loop A FLBB: feed line break of loop B SGTRA: steam generator A’s tube rupture SGTRB: steam generator B’s tube rupture SLBA: steam line break of loop A SLBB: steam line break of loop B
Age (average)	24 years	25 years	
Gender	12 males; 3 females	11 males; 2 females	
Instruction	3 weeks	1 week	
Education	Postgraduates	Postgraduates	

Table 2. Diagnostic performance on FLBA, FLBB, and SGTRA events

Event	FLBA		FLBB		SGTRA	
	Group 1	Group 2	Group 1	Group 2	Group 1	Group 2
Mean	0.13	0.08	0.13	0.076923	0.07	0
Variance	0.12	0.08	0.12	0.076923	0.07	0
Observations	15	13	15	13	15	13
Pooled variance	0.10		0.10		0.04	
<i>df</i>	26		26		26	
<i>t</i> Stat	0.465731		0.465731		0.928571	
$P(T \leq t)$ one-tail	0.322643		0.322643		0.180826	
<i>t</i> Critical one-tail	1.705618		1.705618		1.705618	
$P(T \leq t)$ two-tail	0.645286		0.645286		0.361652	
<i>t</i> Critical two-tail	2.055529		2.055529		2.055529	

Table 3. Diagnostic performance on SGTRB, SLBA, and SLBB events

Events	SGTRB		SLBA		SLBB	
Group	Group 1	Group 2	Group 1	Group 2	Group 1	Group 2
Mean	0.07	0	0.07	0.08	0.07	0.08
Variance	0.07	0	0.07	0.08	0.07	0.08
Observations	15	13	15	13	15	13
Pooled variance	0.04		0.07		0.07	
<i>df</i>	26		26		26	
<i>t</i> Stat	0.928571		-0.10129		-0.10129	
$P(T \leq t)$ one-tail	0.180826		0.460047		0.460047	
<i>t</i> Critical one-tail	1.705618		1.705618		1.705618	
$P(T \leq t)$ two-tail	0.361652		0.920094		0.920094	
<i>t</i> Critical two-tail	2.055529		2.055529		2.055529	

These diagnostics performance assessment results were shared with the trainee operators and feedback was provided in two debriefing sessions, one for each group. Two expert operators delivered these debriefing sessions. All the participants were instructed to review QNPP's manuals and prepare for the diagnostics assessment through the QNPP's simulator. The expert operators emphasized the recognition and memorization of each symptom related to the propagation path of each event.

Again, in the simulator-based diagnostic assessment of the trainee operators, the same measure, *diagnostic accuracy*, was used. Indeed, one can argue that the timing measure (i.e. how fast an operator can identify a fault) should have been assessed as well. In fact, when we looked at the computer logs of all the participants, they all had completed each event's diagnosis in less than 80s. The expert operators at QNPP termed such a timing performance quite satisfactory. How about their statistical performance on simulator-based diagnostic testing? Because participants prepared well and were eager to perform—soon they have to apply for licensing, both the groups performed successfully—all of trainee operators identified each of the events correctly.

We recognize the limitations of these evaluations (e.g. small sample size and only one performance measure) of QNP_SHELL. Nevertheless, with the reported results of our qualitative (i.e. Turing testing) and quantitative (i.e. statistical validation) assessments of the diagnostic performance of the trainee operators, we are confident about the continued utility and success of QNP_SHELL at QNPP and elsewhere (e.g. other NPPs of our client). In fact, as the quality of training is the leading factor that shapes the actual performance of the NPP operators (Alvarenga, Frutuoso e Melo, & Fonseca, 2014), we can expect improved decision-making skills by the QNP_SHELL trained operators.

5. Concluding remarks

The role of human operators in increasing the efficiency and availability of complex systems such NPPs is critical. Therefore, the operator's training on the accurate identification of the root causes of faults is indispensable. The developed framework system allows the builders of expert systems to focus on the development of knowledge bases without having to develop a new program structure that otherwise is needed to process them. Utilizing this framework, an expert system for fault diagnosis training of the power plant operators at QNPP has been developed as a part of indigenous capability development program of our client. The developed, QNP_SHELL-based, expert system was tested on a range of events (e.g. LOCA, SGTR, FLB, and SLB) that can occur in a PWR reactor. Using the analysis and advice of QNP_SHELL expert system, the operators were able to (i) identify the correct type of event, and (ii) execute the relevant emergency procedures in time—a *raison d'être* of any diagnostic system. Therefore, the developed expert system can be applied in diagnosing an accident situation like LOCA and act as an additional confirmatory aid to plant operators.

Additionally, the modular architecture of QNP_SHELL allows users to study any kind of fault diagnosis over any domain. The only limitation is that the symptoms associated with the faults to be diagnosed can be broken down into a set of questions.

The QNP_SHELL can diagnose a fault or a group of faults having the same unique symptoms. Once the diagnosis has been performed, it can further analyze the confirmation or suspicion to the existence of the diagnosed results. Provision has been kept to further enhance the capabilities of the QNP_SHELL to accommodate the analysis of the group of faults that may have the same symptoms but different confirmation requirements. Furthermore, the capabilities of the QNP_SHELL can be enhanced to base the predictions about the existence of fault on probabilistic evaluations. In the next phase of the research project, we intend to develop a web-enabled version of QNP_SHELL enabling simultaneous multi-site training sessions for the NPP operators.

Acknowledgments

The author would like to thank the two respected anonymous reviewers for their useful comments and critique. Also, an earlier version of this paper (a short paper of five pages) was accepted for presentation and publication in the proceedings of the conference, ISMS2012, Malaysia.

Funding

The author received no direct funding for this research.

Author details

Hassan Quadrat-Ullah¹

E-mail: hassnq@yorku.ca

ORCID ID: <http://orcid.org/0000-0002-8018-2363>

¹ School of Administrative Studies, York University, 4700 Keele Street, Toronto, Ontario, Canada M9V 3K7.

Citation information

Cite this article as: QNP_SHELL: A computerized tool for improving decision-making skills for nuclear power plant operators, Hassan Quadrat-Ullah, *Cogent Engineering* (2015), 2: 1030829.

Cover image

Source: Author.

Note

1. At the request of our client, we have used anonymous names, QNP_SHELL and QNPP, as referenced in this paper.

References

- Abraham, A. (2005). Rule-based expert systems. In H. Sydenham & R. Thorn (Eds.), *Handbook of measuring system design* (pp. 909–919). New York, NY: Wiley.
- Abu-Khader, M. (2009). Recent advances in nuclear power: A review. *Progress in Nuclear Energy*, 51, 225–235. <http://dx.doi.org/10.1016/j.pnucene.2008.05.001>
- Alvarenga, M. A. B., Frutuoso e Melo, P. F., & Fonseca, R. A. (2014). A critical review of methods and models for evaluating organizational factors in human reliability analysis. *Progress in Nuclear Energy*, 75, 25–41. <http://dx.doi.org/10.1016/j.pnucene.2014.04.004>
- Angeli, C. (2010). Diagnostic expert systems: From expert's knowledge to real-time systems. In P. Sajja & R. Akerkar (Eds.), *Advanced knowledge based systems: Model, applications & research* (Vol. 1, pp. 50–73). Kolhapur: Technomathematics Research Foundation.
- Apostolakis, A., Kafka, P., & Mancini, G. (1988). *Accidence sequence modeling*. London: Elsevier Applied Science.
- Buchanan, B. G. (1986). Expert systems: Working systems and the research literature. *Expert Systems*, 3, 32–50. <http://dx.doi.org/10.1111/exsy.1986.3.issue-1>
- Francis Cheong Yiu Fung, F. (1989). Framework for building rule-based machine diagnostic expert systems. *Knowledge-Based Systems*, 2, 228–238. [http://dx.doi.org/10.1016/0950-7051\(89\)90067-1](http://dx.doi.org/10.1016/0950-7051(89)90067-1)
- Heo, G., Chang, S., Choi, S., Choi, G., & Jee, M. (2005). Advisory system for the diagnosis of lost electric output in nuclear power plants. *Expert Systems with Applications*, 29, 747–756. <http://dx.doi.org/10.1016/j.eswa.2005.06.002>
- Howie, E., Sy, S., Ford, L., & Vicente, K. J. (2000). Human-computer interface design can reduce misperceptions of feedback. *System Dynamics Review*, 16, 151–171. [http://dx.doi.org/10.1002/\(ISSN\)1099-1727](http://dx.doi.org/10.1002/(ISSN)1099-1727)
- Jiang, H. (2008). Study on knowledge acquisition techniques. In *The Proceedings of 2nd International Symposium on Intelligent Information Technology Applications* (pp. 181–185). Washington, DC.
- Lee, M. (2002). Expert system for nuclear power plant accident diagnosis using a fuzzy inference method. *Expert Systems*, 19, 201–207. <http://dx.doi.org/10.1111/exsy.2002.19.issue-4>
- Li, F., Upadhyaya, B. R., & Perillo, S. R. P. (2012). Fault diagnosis of helical coil steam generator systems of an integral pressurized water reactor using optimal sensor selection. *IEEE Transactions on Nuclear Science*, 59, 403–410. <http://dx.doi.org/10.1109/TNS.2012.2185509>
- Liu, Y., Xie, C., Peng, M., & Ling, S. (2014). Improvement of fault diagnosis efficiency in nuclear power plants using hybrid intelligence approach. *Progress in Nuclear Energy*, 76, 122–136. <http://dx.doi.org/10.1016/j.pnucene.2014.05.001>
- Locatelli, G., Bingham, C., & Mancini, M. (2014). Small modular reactors: A comprehensive overview of their economics and strategic aspects. *Progress in Nuclear Energy*, 73, 75–85. <http://dx.doi.org/10.1016/j.pnucene.2014.01.010>
- Ma, J., & Jiang, J. (2011). Applications of fault detection and diagnosis methods in nuclear power plants: A review. *Progress in Nuclear Energy*, 53, 255–266. <http://dx.doi.org/10.1016/j.pnucene.2010.12.001>
- Milton, N. (2007). *Knowledge acquisition in practice: A step by step guide*. London: Springer Verlag.
- Moshkbar-Bakhshayesh, K., & Ghofrani, M. B. (2013). Transient identification in nuclear power plants: A review. *Progress in Nuclear Energy*, 67, 23–32. <http://dx.doi.org/10.1016/j.pnucene.2013.03.017>
- Najdawi, M., Chung, Q. B., & Salaheldin, S. (2008). Expert systems for strategic planning in operations management: A framework for executive decisions. *International Journal of Management and Decision Making*, 9, 310–327. <http://dx.doi.org/10.1504/IJMDM.2008.017412>
- Naser, J. (1990). *Expert systems applications for the electric power industry*. Singapore: Taylor & Francis.
- Negnevitsky, M. (2005). *Artificial intelligence: A guide to intelligent systems* (2nd ed.). Englewood Cliffs, NJ: Addison Wesley.

- Newell, A., & Simon, H. (1972). *Human problem solving*. The University of Michigan: Prentice-Hall.
- Santhosh, T. V., Kumar, M., Thangamani, I., Srivastava, A., Dutta, A., Verma, V., ..., Ghosh, A. K. (2011). A diagnostic system for identifying accident conditions in a nuclear reactor. *Nuclear Engineering and Design*, 241, 177–184. <http://dx.doi.org/10.1016/j.nucengdes.2010.10.024>
- Spring, G. (1993). Validating expert system prototypes using the Turing test. *Transportation Research Part C: Emerging Technologies*, 1, 293–301. [http://dx.doi.org/10.1016/0968-090X\(93\)90003-X](http://dx.doi.org/10.1016/0968-090X(93)90003-X)
- Sydenham, H., & Thorn, R. (2005). *Handbook of measuring system design*. New York, NY: Wiley. <http://dx.doi.org/10.1002/0471497398>
- Townsend, C. (1989). *Introduction to turbo prolog*. San Francisco, CA: Sybex.
- Vicente, K. J. (2002). Ecological Interface design: Progress and challenges. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 44, 62–78. <http://dx.doi.org/10.1518/0018720024494829>
- Vinod, S., Babar, A., Kushwaha, H., & Raj, V. (2002). Symptoms based diagnostic system for nuclear power plant operations using artificial neural networks. *Reliability Engineering & System Safety*, 82, 33–40.
- Waterman, D. (1986). *A guide to expert systems*. Boston, MA: Addison-Wesley.
- Yong-kuo, L., Min-jun, P., Chun-li, X., & Ya-xin, D. (2013). Research and design of distributed fault diagnosis system in nuclear power plant. *Progress in Nuclear Energy*, 68, 97–110. <http://dx.doi.org/10.1016/j.pnucene.2013.06.002>



© 2015 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format
Adapt — remix, transform, and build upon the material for any purpose, even commercially.
The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.
You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



Cogent Engineering (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com

