ELSEVIER

Nuclear Engineering and Design 159 (1995) 259–264

Nuclear Engineering and Design

Technical Engineering Note

ESAP, an easy-to-use expert system for the systematic analysis of operator actions within the scope of probabilistic risk assessment

G. Degen, J. Mertens, B. Reer*

Forschungszentrum Jülich GmbH (KFA), Institute for Safety Research and Reactor Technology, D-52425 Jülich, Germany

Abstract

The assessment of operator actions within the scope of probabilistic risk analyses is an important task; however, it is connected with considerable uncertainties. The aim of the expert system ESAP (an easy-to-use expert system for the systematic analysis of operator actions within the scope of probabilistic risk assessment) is the reduction of those uncertainties caused by subjectivities in using the technique of human error rate prediction (THERP) methodology. The transformation of THERP using the expert system shell knowledge engineering environment (KEE) and the input logic are briefly described as well as the calculation of error probabilities using KEE rules. The advantages of ESAP and further developments are summarised.

1. Introduction

The assessment of operator actions in the course of probabilistic risk analyses (PRA) is an important but difficult task saddled with remarkable uncertainties. On the one hand the operator models available are not suitable for an adequate description of the possible human behaviour, and on the other hand there is a lack of so-called 'hard data'. Moreover, the results of human reliability analyses are substantially influenced by the quality of the identification process of critical operator actions and by subjectivity in the course of model application. The aim of the expert system presented here is the reduction of such user-induced uncertainties. An expert system was chosen for the typical possibilities of extension and also for the typical knowledge base which grows with applications (Kosmowski, 1994).

2. Basis and design of the expert system

2.1. Operator model

The technique of human error rate prediction (THERP) was taken as the methodological basis of the development of the expert system because

* Corresponding author.

0029-5493/95/$09.50
SSDI 0029-5493(95)01089-0

this method is used in most PRA for nuclear power plants. The model application is supported by a lot of data in the Swain Handbook of THERP (Swain, 1983).

The application of THERP comprises the following principal steps:

- definition of the system failure to be analysed ('top event');
- identification of relevant operator errors (by analysing procedures and their steps);
- quantification of identified errors using basic error probabilities and correction factors to consider special influences;
- calculation of the system failure probability by connecting all error probabilities.

2.2. *Structure of knowledge engineering environment (KEE)*

The expert system shell (KEE, 1988) was used to build up an easy-to-use expert system for the systematic analysis of operator actions within the scope of probabilistic risk assessment (ESAP). The features of the model to be handled are represented in the KEE-logic by hierarchically ordered objects with properties and rules.

2.2.1. *Units and slots*

Objects are represented in KEE by units. KEE differentiates between a class unit (a set of similar objects) and a member unit (a representative of such a class unit). The classification of objects into class units or member units must be made by the person generating the expert system and depends on the desired degree of detail definition. The units are hierarchically ordered. There are no limits to the combination of links between units. Each unit can belong to an arbitrary number of super classes, and each class-unit can be divided into sub-classes and/or member units. Slots are defined for each unit to describe an object. A slot can be a property of the object (e.g. colour) and also its behaviour (e.g. a program with calculation rules).

2.2.2. *Rules*

The use of rules is another approach to simulate the behaviour of objects in KEE. A detailed description of the rules would go far beyond this context, so that in the following some important properties of the rules will be specified.

Rules are formulated in a language similar to English in order to make them comprehensible even for inexperienced users. The general form of a rule is: IF premise(s), THEN conclusion(s). KEE offers two fundamental possibilities of logical reasoning: forward chaining and backward chaining. Forward chaining begins with a new premise and determines the resultant conclusions, whereas backward chaining uses known facts in order to verify a hypothesis.

2.2.3. *KEE structure for THERP*

In the case of modelling THERP, the system failures with the corresponding procedures and steps (the wrong execution of any step leads to a system failure) are considered as objects (units). Each procedure and each step has a number of properties (slots) which are important for safe execution, e.g. personal redundancy or the amount of stress, and have special influence on basic error probabilities. Rules indicate how to calculate the error probabilities as functions of the 'properties'. The results are stored as 'properties' of the objects, too.

3. Realisation of the THERP application using KEE

3.1. *Problem structure in KEE*

Each system failure, each procedure and each step are represented in KEE as a unit. Since a system failure consists of several procedures which are again divided into a number of steps, a natural hierarchy is obtained for a system failure. The class of procedures are divided into procedures prior to the accident (PP) and procedures during the accident (PD). System failures and steps are also combined in superclasses. In total, the hierarchy shown in Fig. 1 is obtained.

3.2. *Example for demonstration*

An emergency cooling test of a research reactor is taken as an example to demonstrate the

THERP application using KEE. In the case of a primary circuit leak, the cooling water running in a pump sump is delivered back into the reactor vessel. For test purposes the leak is simulated by a special connection leading into the pump sump. The system failure (top event) here is 'emergency cooling unavailable due to error during test'. For the identification of relevant errors the procedure 'test of emergency cooling' has to be analysed. The procedure consists of 40 steps divided into groups. The omission of step 37 (bend installing of emergency cooling plug) was identified as relevant by the analyst.

3.3. Input

The user must enter the hierarchy for each system failure, i.e. for all procedures and their steps. He must tell the system which steps lead directly to system failure, which combinations of steps lead indirectly to system failure and which of these failure paths are used for further calculation. Moreover, a number of questions must be answered for each procedure and each step. The total input is controlled by a menu.

3.3.1. Entering the hierarchy

In Fig. 2 the menus are shown for entering the unit-hierarchy of the above example. A defined sequence must be observed in entering the hierarchy. A procedure cannot be entered unless it is known to which system failure it belongs. Moreover, the higher-level procedure must be defined for every new step. The menu always permits only those operations which are possible at this time and is started by a LISP program. This program also generates new units to extend the hierarchy.

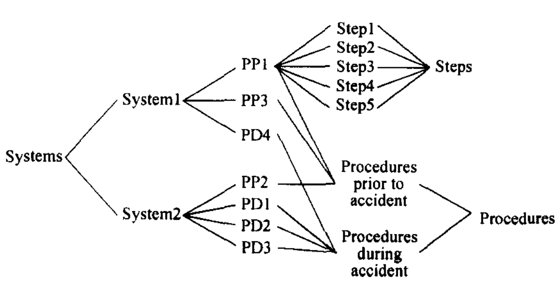

Fig. 1. Unit hierarchy of THERP in KEE.

For example, entering the text 'Emergency cooling unavailable' (see Fig. 2) results in generating a new system unit and in adding this unit as a subclass to the superclass 'systems' (see Fig. 1).

3.3.2. Data input for procedures

A number of questions must be answered for each procedure (see Fig. 3). As soon as a procedure is selected using the menus, this data input is basically possible. If data have been entered already, the user is informed accordingly and can decide whether or not he will change these data. The actual data input for the procedure begins with the output of the first question and the associated possible answers. After having selected an answer (using the mouse), further questions and associated answers appear as a function of the first answer.

3.3.3. Data for steps

If a procedure is defined and steps have been entered for this procedure, the data for these steps can be entered. The data for the steps are entered in two sections. First of all, the step sequence must be entered and the steps of importance for further calculations (i.e. the step belongs to an important failure path) must be identified. The catalogue of questions must then be completed for each of these steps; Fig. 4 gives an example. Before starting the actual data input, the program is asked whether data are already available. When all important failure paths are entered, the catalogue of questions is completed for each step in this failure path. This input is the same as described for the data input procedures.

3.3.4. Calculation of the probability and output of results

The failure probability of a procedure can be calculated after complete data input. The median and uncertainty factor of basic error probabilities, modified error probabilities, correction error probabilities, and the error conditions of all steps of a procedure are calculated using KEE rules. The rules are ordered in groups by a KEE hierarchy and the groups are started by a LISP program. For instance, one rule belonging to the questionnaire and modifying the basic error probability has the following form.

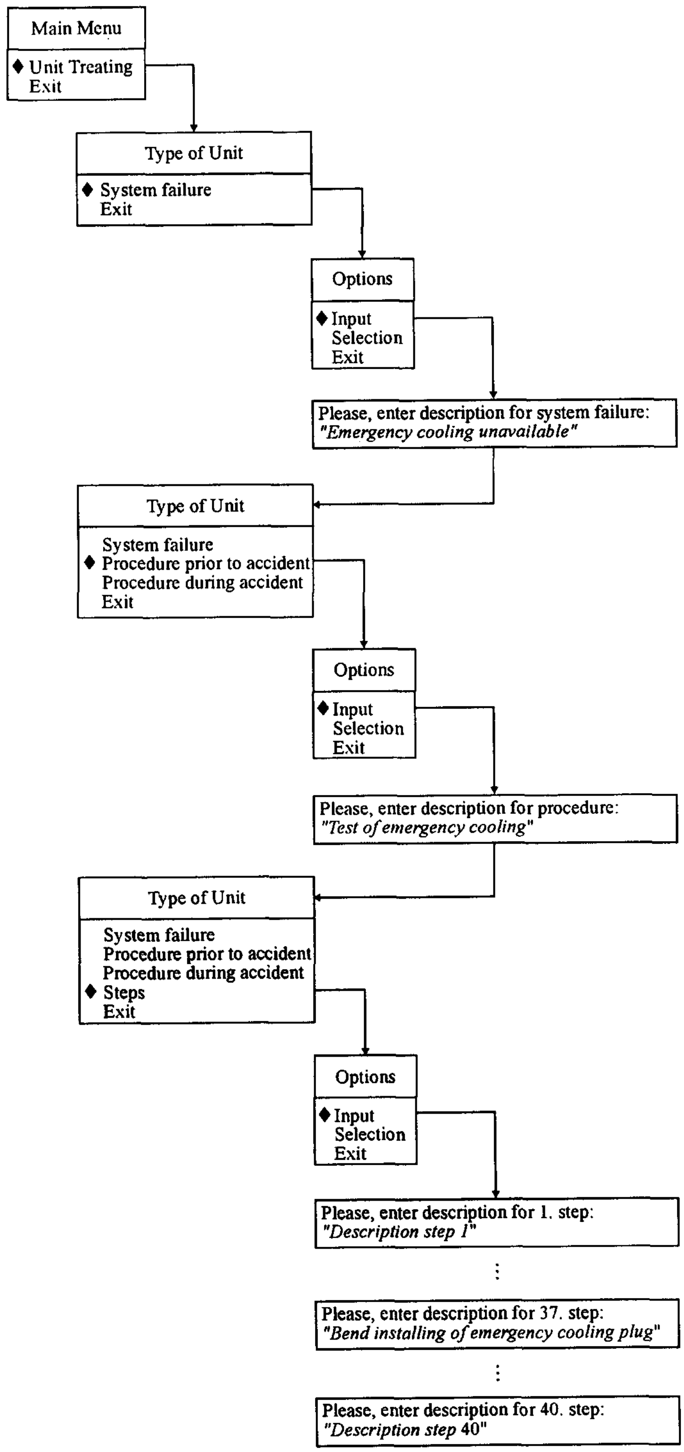

Fig. 2. Menu for entering the unit hierarchy: ♦, selected item; *Text*, input by user.

<table>
<tr><td colspan="6">The procedure is performed by an ...</td></tr>
<tr><td colspan="3">experienced operator</td><td colspan="3">inexperienced operator</td></tr>
<tr><td colspan="6">The procedure is known to the operator due to ...</td></tr>
<tr><td rowspan="3">an oral instruction</td><td colspan="5">a written document</td></tr>
<tr><td colspan="5">Is the written document available as a check list?</td></tr>
<tr><td colspan="3">yes</td><td colspan="2">no</td></tr>
<tr><td colspan="6">At which level of stress is the procedure performed?</td></tr>
<tr><td colspan="2">too low</td><td>optimal</td><td colspan="2">high</td><td>extremely high</td></tr>
</table>

Fig. 3. Example of questions to be answered for each procedure.

IF
  (?step belongs to the classes steps and *procedure*)
  (slot ans-2.5 in *procedure* has the value 'optimum')
  (slot ans-2.1 in *procedure* has the value 'unexperienced')
  (slot ans-1.1 in ?step has the value 'high')
THEN
  (modify the median and uncertainty factor of probability slot of ?step)

Of course, this rule is not represented here in the formal KEE language but in a more readable form. ?step is a variable which consecutively has as its value all names of the units belonging to the class of steps and to the class *procedure* (name of the current procedure unit). This variable bond is achieved by the first condition in the IF part. The other conditions ask for the values of individual slots, 'ans-2.5' giving the answer to the question for the operator's stress level, 'ans-2.1' the answer to the question for the operator's experience and 'ans-1.1' the answer to the question for the cognitive level required to perform the current step. If all slots have the corresponding values, the median and uncertainty factor of the modified probability are calculated for the current step using a LISP program. After processing all rules, the other parameters (lower bound, upper bound, mean and variance) of the probabilities determined by rules and all parameters of the probabilities of the error type contribution, the step failure probability, the failure path probability and the procedure failure probability are calculated by a LISP program and/or active values.

Furthermore, the program generates a data set containing the parameters of the procedure failure probability for all procedures which had to be calculated. This data set can be printed out on request.

4. Summary and scope

Table 1 summarises the analysis steps of THERP, their realisation in ESAP and the respective advantages using ESAP.

The following further developments are planned or are under discussion:

— implementation of a knowledge base increasing with every THERP-session
— extension of THERP to consider time dependencies;
— documentation of the results of all parameter calculations;
— support to identify relevant steps;
— quantification of non-stochastic uncertainties of decision processes regarding the modelling and assessment of operator actions.

Based on existing experience in developing ESAP, the conclusion can be drawn that it will be more difficult to include some support for the user to identify relevant steps than to refine details regarding the quantification of error probabilities.

<table>
<tr><td colspan="6">Does the performance of the step require recognition of an information signalized by a reading board?</td></tr>
<tr><td rowspan="7">no</td><td colspan="5">yes</td></tr>
<tr><td colspan="5">Is it possible not to perform the step by watching the wrong reading board?</td></tr>
<tr><td rowspan="5">no</td><td colspan="4">yes</td></tr>
<tr><td colspan="4">Is the considered reading board positioned within a group of similar boards?</td></tr>
<tr><td rowspan="3">no</td><td colspan="3">yes</td></tr>
<tr><td colspan="3">How is the considered reading board distinguished from adjacent reading boards</td></tr>
<tr><td>only by labelling</td><td>additionally by arrange-ment in a functional group</td><td>additionally by clearly drawn seperating lines</td></tr>
</table>

Fig. 4. Example of questions to be answered for each step.

Table 1
Analysis steps of THERP, their realization in ESAP, and the resulting advantages

Analysis steps of THERP	Realization in ESAP	Advantages using ESAP
Identification of operator actions	– Menu-driven input – Storage as unit-hierarchy	– Stable storage of structure – Simplicity of extensions and changes
Identification of relevant steps	– Complete list of steps and demand for selection	– Possibility of checking the results – Simplicity of changes
Data selection and input	– Answering of questions presented in the right order – Storage of the answers in slots	– Completeness – Support by default values and comments – Possibility of checking the results – Parameter calculations
Calculations	– Carried out automatically by program – Storage of the result in slots	– No lists needed – No mistakes in calculations

References

KEE (knowledge engineering environment) User's Guide, Version 3.1, Intellicorp, Inc., K3.1-UG.-1, May 1988.

K. Kosmowski, G. Degen, J. Mertens and B. Reer, Development of advanced methods and related software for human reliability evaluation within probabilistic safety analyses, Jül-2928, Forschungszentrum Jülich GmbH, 1994.

A.D. Swain and H.E. Guttmann, Handbook of human reliability analysis with emphasis on nuclear power plant application, Final Rep. NUREG/CR-1278, 1983.