17th International Mining Congress and Exhibition of Turkey- IMCET2001, © 2001, ISBN 975-395-417-4

An Outline of a Hybrid Intelligent System for Mine Safety Analysis

N.M.Lilic & LM.Obradovic
Faculty of Mining and Geology, University of Belgrade, Yugoslavia
F.Vukovic
"Pljevlja " Coal Mine, Yugoslavia

ABSTRACT: The analysis of safety in coal mines is a very complex process based on the estimation of numerous interdependent parameters. The complexity of the subject matter requires a high level of expert knowledge and great expenence. This paper discusses one of the possible approaches to solving this problem by a hybrid intelligent system for surface mine safety analysis (PROTECTOR).

1 INTRODUCTION

One of the most significant and also most complex problems encountered in surface coal mines is safety analysis. An appropriate and reliable solution to this problem İs vital for the working process in mines with surface exploitation. Contemporary mining theory already operates with a number of mathematical methods which can be used to solve mine safety problems. These methods are used in current engineering practice with the help of appropriate software products.

Numerical software products İn mine safety need expert knowledge and experience to be fully exploited. This knowledge consists of rules which heuristics experts use when they apply numerical methods. The knowledge-based approach developed by artificial intelligence (AI) offers the possibility of incorporating knowledge in software systems. Proponents of AI have devoted a major part of their efforts to the formalisation of knowledge representation and the development of mechanisms for using this knowledge, which has finally resulted in knowledge-based or intelligent systems. The so-called expert systems (ES), whose main purpose is to emulate the behaviour of an expert in a specific domain, are the most popular of these. This is due to the fact that they have been successfully implemented in different areas, ranging from medical diagnosis to complex computer network design, where they have produced very good results (Durkin, 1994).

Artificial intelligence has developed a number of methods and tools for solving complex problems. However, the complexity of some problems exceeds the potentials of single methods. A possible solution is to combine two or more AI methods into a hybrid intelligent system. This approach has been adopted in the case of PROTECTOR.

This paper describes the salient features of PROTECTOR, a hybrid system for the analysis and estimation of the mine environment, developed at the Faculty of Mining and Geology at the University of Belgrade. PROTECTOR was developed by combining neural networks and expert system technology. While the mine environment estimation methodology is implemented through an expert system, some of the related estimation parameters are determined by neural networks.

Section 2 of this paper outlines the global problem-solving strategy through a hierarchical decomposition of the main goal, the evaluation of the mine environment, and the formalisation of this strategy by means of a modified object-oriented analysis (OOA) model. The system structure and the main architectural components of the PROTECTOR system are described in Section 3. The implementation of the system İn the KAPPA-PC expert systems shell is discussed in Section 4, followed by the conclusions in the last section.

2 A FORMALISATION OF MINE ENVIRONMENT EVALUATION PROBLEM SOLVING

The analysis of safety in coal mines İs a very complex process based on the estimation of numerous interdependent parameters that are classified by means of several basic criteria for the estimation of the mine environment. These criteria are related to the following conditions: gas, dust, climate, noise, vibration and illumination.

The analysis of safety in surface coal mines in the PROTECTOR system is based on heuristics formulated by mine safety experts. The main goal of the system is the estimation of the mine environment. This global goal can be subdivided into a hierarchical structure of subgoals, where each of these subgoals can be viewed as the estimation of a set of parameters which determine the general state of mine safety and the category of danger in the mine environment. Once the parameter values are obtained, the estimation process for the general state of safety of the mine environment may begin. During this process, the importance, i.e., significance, of each particular parameter must be taken into account. The hierarchical decomposition of the main goal into subgoals, representing the problem-solving strategy, makes it easier to cope with the complexities and to coordinate the use of the knowledge incorporated in the system.

The strategy for the evaluation of safety in the mine environment is formally represented using a modification of the Coad-Yourdon object-oriented analysis (OOA) model (Coad and Yourdon, 1991) In the classical model, every real world entity is represented by a class (object), consisting of its name, attributes and die methods pertaining to the procedures related to the object. In order to incorporate declarative knowledge, this model was modified by including a new (fourth) element, featuring the production (IF-THEN) rules related to an object in the model. Thus, both the procedural and declarative knowledge related to a class object could be represented. Such a modified OOA model was then used for the representation of the mine environment evaluation strategy as well as other objects in the system and their mutual relationships (Fig. 1). The inheritance relations between hierarchically connected objects representing elements of me strategy are given by full lines, while the exchange of messages between classes is represented by dotted lines. The model was the basis for the implementation of the system in an object-oriented expert system shell.

3 ARCHITECTURE OF THE PROTECTOR SYSTEM

PROTECTOR contains the "classical" elements of an expert system: a knowledge base, an inference engine, a user interface and a working memory, but also a module for the interface with routines for relevant parameter determination, the Visual Basic routines themselves, and a database used by these routines (Fig. 2).

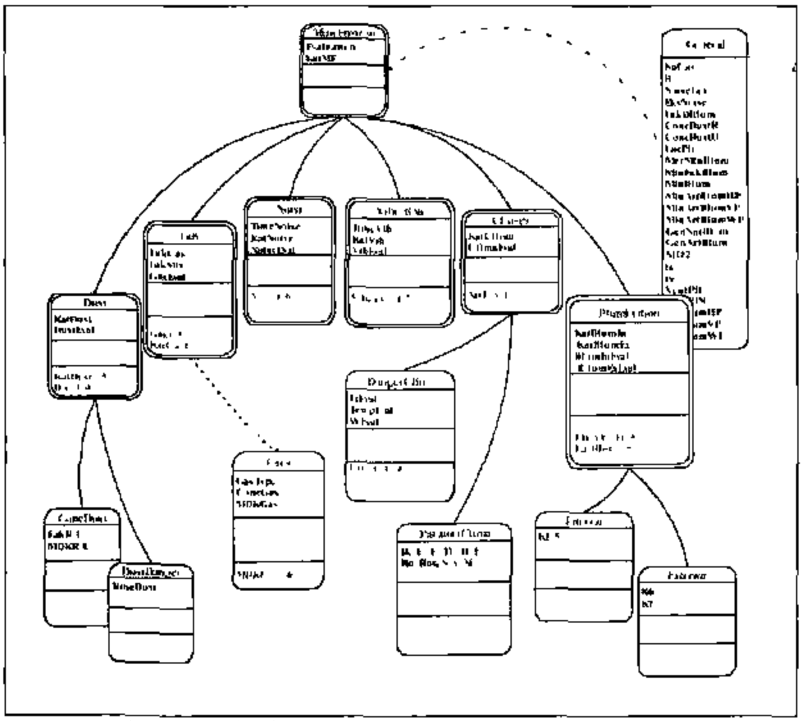


Figure 1. Modified OOA model of the mine environment evaluation

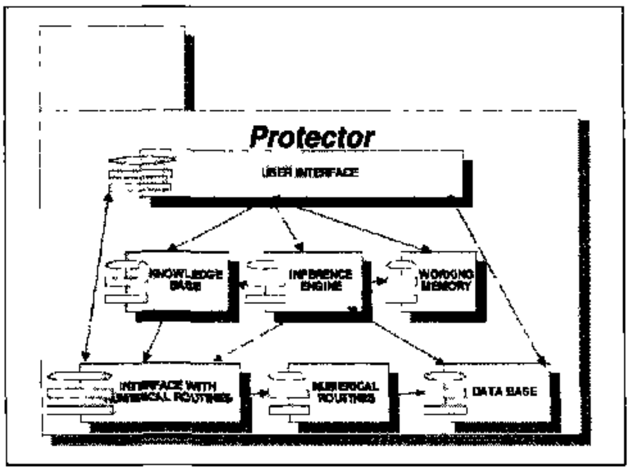


Figure 2 Architecture of the hybrid intelligent system PROTECTOR

The main purpose of the *user interface* is to provide means for a successful dialogue, i.e., an exchange of information between the user and the system. It is the user interface that enables PROTECTOR to obtain all the necessary information from the user, and that then transforms the system's results and conclusions into information the user can understand.

The PROTECTOR *knowledge base* is a formalisation of the mine safety expert's knowledge. Knowledge in expert systems basically consists of facts and heuristics which can be represented by means of rules, frames, semantic networks and other formalisms. Since knowledge is the key factor in problem solution and decision making, the quality and usability of an expert system is basically determined by the accuracy and completeness of its knowledge base. The selection of the representation formalism is very important and plays a significant role in knowledge-based design.

The problem-solving strategy is realised by the expert system's *inference engine.* This reasoning mechanism infers conclusions based on knowledge from the knowledge base and the available information pertaining to the safety problem at hand. The inference engine stores intermediate results in the *working memory.*

The PROTECTOR system was developed as a knowledge-based (symbolic) upgrade of the numerical package for relevant safety parameter determination, and it thus belongs to the category of *coupled numerical and symbolic systems* (Kowalik, 1986). The numerical part consists of routines for the determination of relevant safety parameters using neural networks and for the analysis of degree of importance. The symbolic part formalises the mine safety expert's knowledge.

Successful implementation of a coupled system requires the solving of a number of complex problems in order to obtain efficient communication between the symbolic and the numerical part of the system. In order to cope with this problem, a separation of the processes in coupled systems into independent modules is suggested. Furthermore, information interchange, i.e., communication among modules, is strictly defined and reduced to the lowest possible degree. Communication between two modules is allowed only through previously defined external links, while all implementation details remain "hidden" within the module itself. These requirements can be met successfully through the modified object-oriented approach proposed in this paper with the object as the modular unit of the system. Objects consist of attributes (structures representing their internal data), methods (procedural components), and rules (declarative components).

The object/attribute approach is often mapped into the frame/slot paradigm, which can be successfully used for its implementation (Rich, 1983). In the same way, the characteristics of an object are represented by its attributes, and the frame characteristics are represented by its slots. Slot values describe attributes of the object represented by the frame and its relations to other frames (objects) in the system. The object-oriented approach implemented as a system of frames offers a suitable formalism for the proposed decomposition of the general mine safety state evaluation problem, since they both posses a hierarchical structure. The outlined features of the knowledge base facilitate both the coordination of knowledge within the knowledge base and the communication between the symbolic and the numerical part of the PROTECTOR system.

4 IMPLEMENTATION ISSUES

PROTECTOR was developed using an expert systems shell, the KAPPA-PC applications development system. KAPPA-PC is a MS Windows application which provides a wide range of tools for constructing and using applications by means of a high-level graphical environment which generates standard C code. In the KAPPA-PC system, the components of the domain are represented by objects that can be either classes or instances within classes. The relationships among the objects in a model can be represented by linking them together into a hierarchical structure. Thus, the modified OOA model based on the strategy for evaluation of the general state of safety of the surface coal mine could be easily mapped onto the appropriate elements of KAPPA-PC.

Object-oriented programming tools within KAPPA-PC were used lo provide PROTECTOR objects with methods which specify what objects can do. First, the objects and methods for the knowledge base were constructed. Then, mechanisms were built that both specify how objects should behave and reason about the objects by using rules. Each rule specifies a set of conditions and a set of conclusions to be made if the conditions are true. The conclusions may represent logical deductions about the knowledge base or specifications of how it changes over time. Each rule is a relatively independent module, which made it possible to build the reasoning systems gradually. rule by rule

Since applications written in KAPPA-PC are intended to perform very complex tasks, a variety of graphic images in building the user interface can be used in order to observe and control the operation of an application. An important feature of KAPPA-PC is its usability for enhancing existing systems. For example, it can link into existing programs written in standard languages. Therefore, KAPPA-PC was a powerful tool for integrating the existing numerical routines into the PROTECTOR hybrid system.

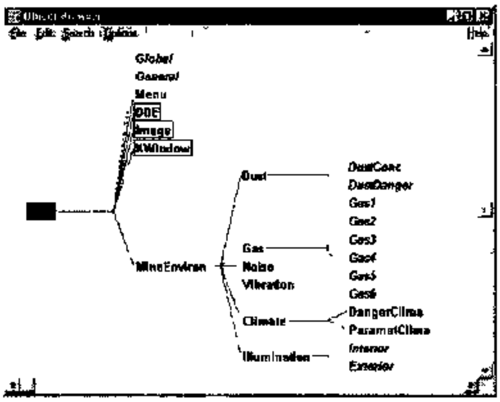
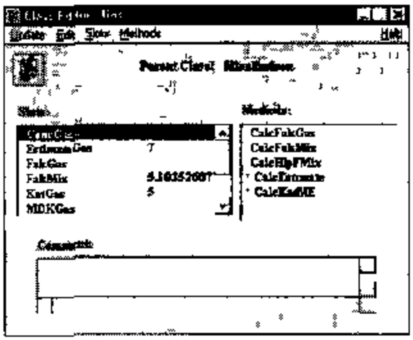


Figure 3. The ICAPPA-PC Object Browser for PROTECTOR

The classes and objects of the modified OOA model were transformed to classes and instances in the KAPPA-PC system as shown in the system's object browser (Fig. 3). The object browser also shows the classes that KAPPA-PC generates for each application, such as Root, Image and Kwmdow



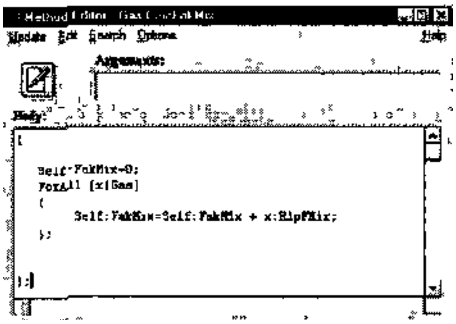
ConcGas
EstimateGas 7
FakGas
FakMix 5.16352607
KatGas 5
MDKGas

CalcFakGas
CalcFakMix
CalcHlpFMix
CalcEstimate
CalcKatME

Figure 4 The Gas Class

Classes/instances are described using the class/instance editor, while slot facets are defined by means of the slot editor. Slots represent class attributes, while methods in the class/instance editor account for both methods and IF-THEN rules related to a class in the modified OOA model. For example, consider the Gas class given in Fig. 4. The class has a parent class MineEnviron and six slots. Five methods are listed. The first three are numerical procedures (CalcFakGas, CalcFakMix, CalcHIpFMix) used for calculating the value of attributes FakGasa and FakMix, and the remaining (CalcKatME, CalcEstımate) contam rules for the evaluation of the mine environment on the basis of several parameters and their mutual relationships.

The methods are described by the method editor. The method editor for the CalcFakMix method is given in Fig. 5.



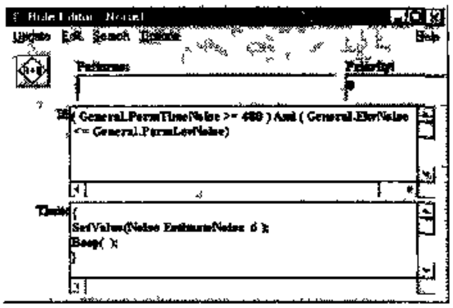
```
{

  Self:FakMix=0;
  ForAll [x|Gas]
  {
      Self:FakMix=Self:FakMix + x:HlpFMix;
  };

};
```

Figure 5 The CakFakMix

Since all rules in the system do not have to be related to particular objects, KAPPA-PC offers the possibility of specifying rules independently, using a rule editor as shown m Fig 6

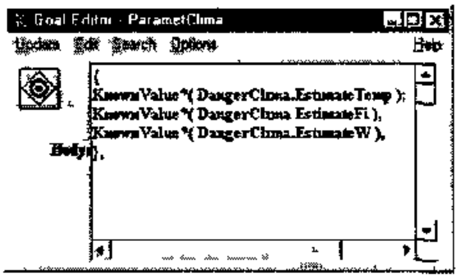


```
( General.PermTimeNoise >= 480 ) And ( General.EkvNoise
<= General.PermLevNoise)
```

```
{
SetValue(Noise EstimateNoise 6 );
Beep( );
}
```

Figure 6 Noise 1

The problem-solving process m PROTECTOR unfolds by means of the KAPPA-PC backward-chaining inference engine Goals to be satisfied by backward chaining are defined by means of the goal editor The goals in PROTECTOR pertain to the estimation of different parameter values A decomposition of the goal ParametCluna into subgoals is illustrated in Fig 7



```
{
KnownValue?( DangerClima.EstimateTemp );
KnownValue?( DangerClima EstimateFi ),
KnownValue?( DangerClima.EstimateW ),
},
```

Figure 7 The ParametClima Goal

Goals can also be generated and modified within methods This feature enables the creation ol new rules and the modification of existing ones dynamically during system operation

The interface developed for PROTECTOR m KAPPA-PC fully exploits the GUI (graphical user interface) technology available lor MS Windows applications (Fig 8) It enables straightforward and easy manipulation of input data and control over parts of the problem-solving piocess It also offers suggestions and recommendations to the user which can contribute to the improvement of the overall performance of the mme system

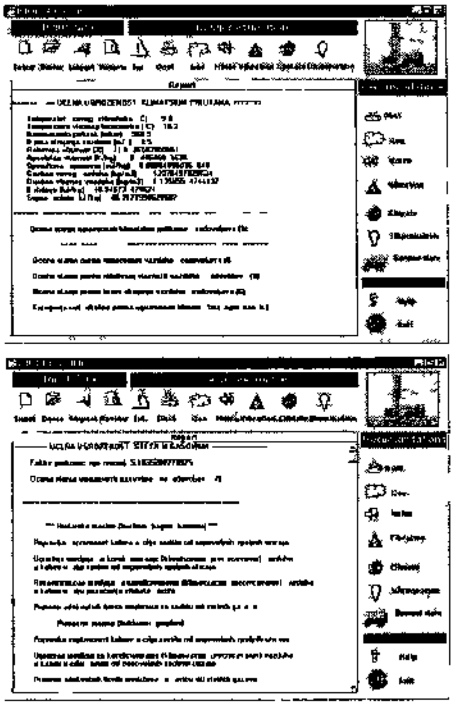


Figure 8 The PRO TECTOR mteriat-t. with esnmalıon and suggestions for improvements

5 CONCLUSION

In this paper, first, a hierarchical decomposition of the safety state evaluation analysis problem was proposed As a consequence, the solution of this complex problem, which is based on the estimation of a number of mutually dependent parameters, is greatly facilitated Then, a possible approach to the solution of this problem through the development of an expert system was outlined The aim of such a system is to generate an overall evaluation of the general state of safety of the mine as well as suggestions for the improvement of particular charactenstics of the mine system

The approach to the realisation of the hybrid system was based on a symbolic upgrading of existing numerical routines It has been argued that the object-oriented frame structure represents a natural approach, considering the coupled numerical-symbolic nature of the system and the hierarchical decomposition of the problem PROTECTOR was developed by integrating neural networks and expert system technology While the mme environment estimation methodology is implemented through an expert system some ot the related estimation parameters are determined by neural networks. The conceived system structure was realised in the KAPPA-PC expert system shell, resulting in PROTECTOR.

REFERENCES

Coad, P. & Ydurdon, E. 1991., *Object-Oriented Analysis,* Second Edition, Prentice Hall, Englewood Cliffs, New Jersey.

Durkin, J. 1994. *Expert Systems. Design and Development,* Macmillan Publishing Company, New York.

Hartman, H., (ed.), 1992. *SME Mining Engineering Handbook,* Society for Mining, Metallurgy, and Exploration, Inc Littleton, Colorado, 2nd Ed. VoL 1.

Kowalik, J.S., (ed), 1986. *Coupling Symbolic and Numerical Computing in Expert Systems,* North-Holland, Amsterdam

Lilié N,, Gagic D. & Stankovic R. 1996. A Knowledge-Based Approach to Coal Mines Safety Analysis, Minesafe International 1996, Pert.

Rich, E. 1983. *Artificial Intelligence,* McGraw-Hill, New York.