A Knowledge Based Emergency Control Advisory System for the Australian Coal Industry

John Aubrey
Zoltan Nemes-Nemeth

Australian Coal Industry Research Laboratories Ltd.
14-30 Delhi Rd., North Ryde
Sydney, Australia

Abstract

The Emergency Control Advisory System (ECAS) is a knowledge-based appliance designed to assist in managing an underground coal mine emergency during the initial stages before the arrival of trained personnel, and throughout the ensuing hours. It is designed to actively guide users through the problem as well as answering specific queries. A training version for rescue personnel uses the same knowledge base, and it is anticipated that the similarity will make the product more useful in the event of an actual call-out. The system has extensive knowledge navigation facilities to allow both detailed justifications for advice, and to preview the possible consequences of actions. The system is designed to allow connection to mine-site software if the need arises. It is designed to be available both as a simple initial response system within the reach of small mine sites, or a larger more versatile system for use by rescue stations or larger mines

History

Following a major underground coal mine disaster in Central Queensland, in 1986, the Australian Government, via the Department of Resources and Energy, announced that a special round of grants would be available, specifically for Coal Mine Safety. The aim of this injection of funds was to promote Research and Development activity, in an effort to help minimise both the effect and occurrence of such disasters.

Australian Coal Industry Research Laboratories Ltd (ACIRL) in conjunction with a range of consultant sub-contractors put forward a research proposal to use "Expert System" technology to develop an emergency control and advisory system for use in disaster management for underground coal mines.

The aim of the project was twofold. Firstly, to develop an easy to use, approachable software tool that could guide, direct and assist staff in the administration and control of an underground mine disaster. Secondly, to have this system available in a "training" form, to be used as part of training courses for underground rescue teams.

Work commenced on the project in July 1986, and the delivery of fully operating systems is scheduled for June 1989. To date, a series of prototype systems have been configured and tested, involving the trialling of several different inferencing mechanisms. A single implementation form has evolved form these trails.

This "final prototype" is based around a Digital VAXstation 2000, running VMS and using NEXPERT as the inferencing engine. This configuration will form the main delivery vehicle for the system, and will contain the complete knowledge base to handle virtually any kind of underground mine emergency from initial response through to clean-up and post mortem. Smaller sub-set systems,using micro-computer hardware, will also be available for use at mine sites. These systems will contain all the necessary knowledge to handle initial response actions for the mine sites.

Introduction

There are three major problems during a mine emergency. The first is psychological : the need to remain calm and clear-headed in the face of potential loss of life or considerable financial loss. There are complex statutory regulations to remember, plus decisions to make about how best to cope with the problem in hand. A system capable of presenting the correct information in the appropriate order would assist in ensuring all essential actions are undertaken.

The second problem is making sure that nothing dangerous is undertaken by inexperienced personnel, and that time is not wasted on futile activities. A system capable of assessing the outcome of actions based on the *status quo* would address this need.

Finally there is a need for coordination, both between the various parties engaged in their diverse activities, and the different work shifts as the emergency unfolds. A system which can keep track of all the goings on and centralise information will help keep everyone informed.

ECAS is an attempt to supply these services to a minesite, using knowledge-based technology . In the words of John Hamment, the project's domain expert:

"It is not the purpose of ECAS to provide ultimate solutions to all emergencies, but to present what benefits may accrue from certain actions, what additional actions may be necessary to achieve those benefits, and finally what dangers or hazards may be associated with such actions." [HAMMENT]

In addition to this brief, the system must have certain key characteristics. It must be able to use available information to initiate and direct efficient problem-solving. Information being entered into the system must be neatly managed and easy to access. The system should also be able to answer direct questions posed by the user, and be capable of explaining and justifying its reasoning.

Representing the problem

Procedures for handling mine emergencies can often be expressed as actions to be taken based of a series of preconditions. This makes the domain suitable for a production rule system. Knowing which questions to ask is as important as being able to answer the questions. Some questions require evaluation after suggestive initial data is encountered, indicating data-driven inferencing. Others are heuristically activated, indicating meta-rules or some other strategy need to be employed. Direct queries require a classical diagnostic goal driven approach. Obviously a mixed-mode inference engine is required to cater for these differing requirements.

In ECAS the rules serve three main functions (fig 1). Primarily they encode the knowledge in such a way that the inference engine can opportunistically work its way through a mine disaster problem. They make the system work.

Secondly they hold the logic behind the solution, and through the explanation facility allow justification of advice. Often advice based on complex criteria such as the chemical behaviour of gas mixtures and the vagaries of underground mine ventilation can appear counter-intuitive. The explanation facility provides an essential service by demonstrating the logic behind apparently ridiculous suggestions. One example of this is feeding more air over a mine fire. Of course the fire burns with more vigor, but in doing so does not leave incomplete combustion products, which are prone to explosions.

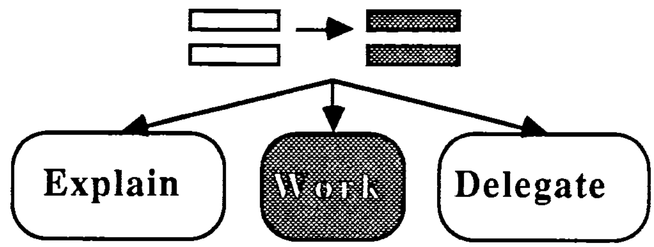

Figure 1

Finally in ECAS's case there is also a need to be able to "delegate" inferencing to humans. In some cases personnel may have to go out of reach of the command centre for periods of time. Rules which describe how to handle contingencies which may arise can be handed out in the form of advice, as well as being a working part of the knowledge base. For instance it may be necessary to remind miners in search of a suspected fire to regularly check the carbon monoxide level, and IF it goes over a certain value, THEN retreat to a safe distance OR use breathing apparatus.

In terms of explanation, rules can indicate what use required data can be put to, or what data is needed to work out a problem (fig 2). This facility is particularly necessary for the training aspect of the system.

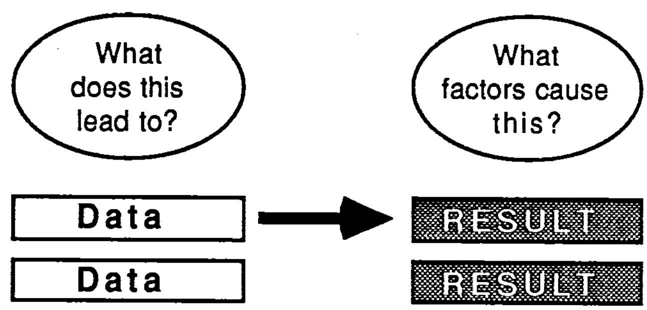

Figure 2

For instance the detection of smoke issuing from a ventilation shaft can lead to an investigation of CO levels in order to determine whether there has been an explosion, plus calls underground to see if any physical evidence can be found there. Conversely the same rule can indicate that one reason for taking CO readings would be if there is smoke issuing from a ventilation opening. The trainee can be taught both warning signs and the use of a procedure. The same rule can be used in a number of ways, depending on how the application accesses the rulebase.

A class system is used to differentiate objects on a functional basis, and to determine the firing priority of rules. Rules which warn about impending hazards must fire before ones being used to make some more general inference. This structure is necessary to assure appropriate results when new rules are added to the system, and to minimise the need to hard-wire rules to promote required behaviour.

The class structure can also be used to attach externally defined behaviors to objects, such as whether they will set up reminders, or end up in special menus to draw attention to them. Objects can be defined in terms of the problems they address, or the personnel who are most likely to be interested in them.

Disaster management

ECAS is designed to be an appliance for dealing with mine disasters. It has an interface which can include data monitoring facilities common in a mining environment, such as gas meters. It allows the user to see only the parts of inferencing that are related to the problem in hand. As much as possible the interface is designed to appear to be helping manage the disaster, with the Expert System functions being an invisible driving force. The use of menus, icons and such is designed to make interaction with the system as simple as possible.

On screen there is the "Audit Log", a timestamped transcript of what dialogue has occurred between ECAS and the user. This allows dumping to file and printer through the click of a button, providing a permanent record of events.

Initially most activity takes place through interspersed information request menus and advice boxes put up by the system. Wherever possible the user is given choices to click upon with a mouse, rather than empty space to type into.

A command menu allows the user to ask specific questions, or to volunteer or alter information. This can be done whenever the system is asking a question or is idle. If the system is idle for a predetermined time, ECAS has the ability to display dynamically marked objects which, if instantiated, can lead to further conclusions. Alternately the user may choose to ask specific questions, or review progress so far.

There is a data browsing sub-menu for seeing what is already in the system, such as all information entered by users, or conclusions arrived at by the system. An explanation sub-menu allows reasons for decisions to be explained, definitions to be given for key objects in the system, and the systems "plan of attack" to be previewed by viewing the items queued for evaluation. Queries are usually not directed at general objects, but at specific classes of objects, depending on the type of query.

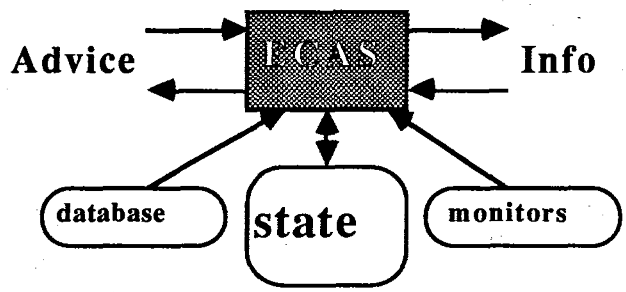

Figure 3

ECAS itself manages the connections between the interface, the inferencing, and the external databases and subprocesses (fig 3). Some information, such as who to notify automatically comes from the mine database, while others, such as gas readings can be sought from a number of sources. The user is left to supply any information which is not otherwise available.

Changing the focus of attention

In a conventional Expert System there may be little chance to interrupt the inferencing process in order to evaluate new information. Questions are asked only when rules are being tested. NEXPERT, the inferencing software used in ECAS, moves one step away from this. It allows the user to "volunteer" information randomly. Its agenda structure means that new problems can be put on the "to do" list at any stage. Rule and object hierarchies can then determine which problem will be addressed next. In the case of ECAS is was thought that this may be too passive an approach. One of the aims of the system is to ensure that things are not forgotten or overlooked. ECAS has the ability to cue crucial questions for re-prompting at given intervals. The user is confronted with the question, and the consequences of any response can alter the agenda. This means that high-priority problems can be addressed as the need arises. Timer delays are kept in a meta-slot attached to each object. They can be overwritten by the right hand side (RHS) of a rule if the need arises. Any object placed into the "TIMER" class is automatically re-prompted after its delay time has been exceeded.

Another problem arising from traditional inferencing is that answers which have been given may lose validity over time. The system needs to be able to know when conditions change, to be able to reevaluate the situation. It is optimistic to think that personnel in the heat of a disaster will remember everything they once entered into the system. ECAS keeps a list of instantiated objects in the "if change" class. These appear on a special menu as a reminder that these factors must be monitored.

During the initial stages of an emergency it is important to enter initial information as quickly as possible, and to reach some basic conclusions. It is not uncommon for certain advice to be entered as "NOTKNOWN". Some of this "NOTKNOWN" information may be necessary for crucial decisions, and ECAS highlights such objects in a separate menu, as a reminder to find out as soon as possible. Information which would be useful, but not crucial, is also displayed in a less dramatic fashion.

There is a third class of data which the system has the sense not to ask because it is obvious that the answer will not be available for some time. For example the system may advise the taking of gas samples, and will need to know when they have arrived. In this case it cues the information in a "when known" menu. It may or may not attach a reminder call to it.

Queries during inferencing

ECAS is designed to be operated at either novice or expert level. To facilitate this a number of objects are labelled to indicate there are rules available to help determine them. An expert may know the response to a question immediately, and attach a value to an object directly. A novice has the opportunity to reply "NOTKNOWN", in which case the system backward chains to attempt to find the answer. This technique means that trivial questions are asked only when required.

Some information requested by the system may require dangerous forays into the disaster area. For this reason a "why ask" facility was incorporated. This allows the user to see the use to which the information will be put, and to determine whether it is worth pursuing. If the user decides not to find out, and the information is crucial, then it will still be added to the "Needed data" menu to remind operators.

Whenever a rule is displayed for the purposes of explanation, it is possible to enquire about the way variables were determined, or the outcomes of variables set by the rule. In many systems it is not possible to trace an uninstantiated variable. In ECAS it is possible to see any cause or outcome of a given variable, whether it is instantiated or not. This allows extrapolations from the "Why ask" or "Explain" windows.

Knowledge Acquisition

Information for the system knowledge base was available from a number of sources.

* Statutory regulations indicate procedures which must be undertaken. They also indicate the correct delegation of authority, and mandatory notifications. These are consistent throughout an Australian state, though small details may vary, depending on the physical characteristics of a mine, e.g. how much methane is likely to be exuded from the local coal seam.

* Mine rescue techniques are the domain of district mine rescue stations. These include a body of knowledge about rescue equipment including gas analysers and communications. Some men at each mine are usually trained at the local rescue station, but during an emergency men from throughout a district may be called in to assist.

* Laws of physics and chemistry play an important part in any disaster, particularly the behaviour of gases, and their potential explosibility.

While regulations may determine what needs to be done, the most effective way of implementing policy may require the personal expertise of a seasoned mine manager. In particular this knowledge determines the order in which activities are most safely and effectively undertaken.

To collate this varied information a retired mine manager and Rescue Station Superintendent was employed to act as information coordinator and domain expert.

Implementation

It soon became apparent that the project needed to be divided into a number of components (fig 4). A small initial actions module needed to be available to virtually any minesite. The delivery system would have to be within the financial reach of small operations, and had to be deliverable at a PC level. This system would have to be comprehensible to the most inexperienced of operators. Only a few choices would be offered, and data-browsing features would be minimal. It was anticipated that this module would consist mostly of a question and advice session, driven by ECAS on the basis of incoming information.

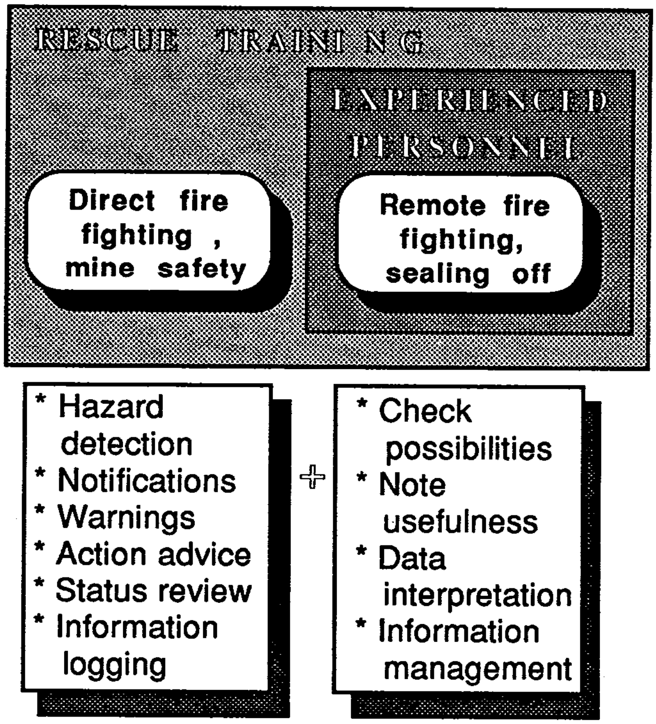

Figure 4

The management modules to be used by experienced personnel needed a far more powerful interface, capable of navigating the knowledge base more freely. This module also needed an ability to manage the increased amount of information being accrued as the emergency proceeded. It was anticipated that this system would be used by the mine rescue team, both for their training, and for deployment during an emergency. It was envisaged that the rescue team could be informed about the disaster so far by the audit log, which could be transmitted by modem to the team before they set out. Upon arrival they could transfer the state of the working memory in the local System into their own version and continue management on the larger system.

Methodology

The original concept was to create a general purpose knowledge base, then do "knowledge queries" upon it. ECAS would translate user requirements into expert system and other requests, and report back to the user (fig 5).

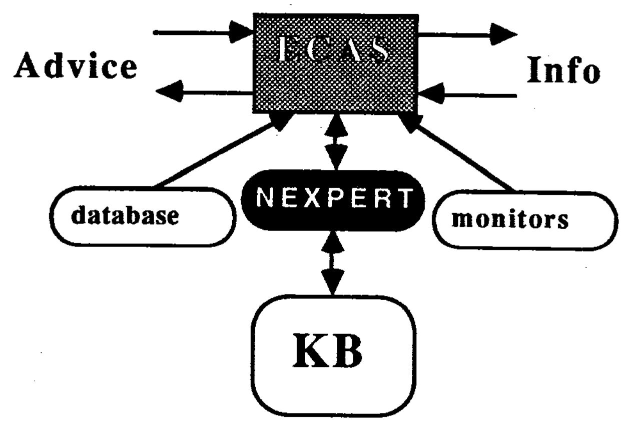

Figure 5

To do this it was necessary to create structures, much as a database has fields and relationships. Atoms correspond to different categories - advice, setting of states, information, and control features (fig 6). Obviously the first task is to hide the control features from the user, or to have meaningful translations for them, such as "Look up the mine personnel database" instead of "DBload...."

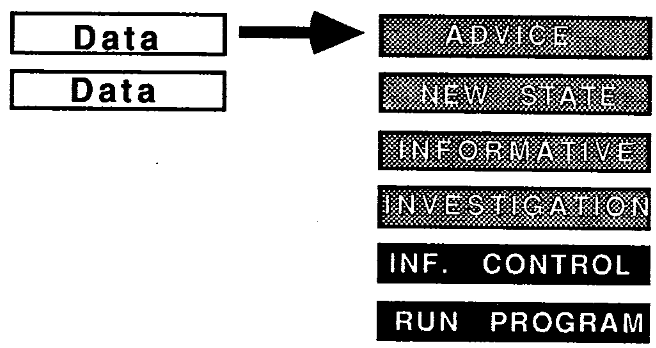

Figure 6

The choice of expert system software was simplified because there were very few products capable of fulfilling the rigorous demands of the system. The software required both forward and backward chaining, device portability, and the ability to customise the interface. NEXPERT offered all this, and its callable interface architecture also allowed customisation of the inferencing. Some parallel prototyping was done in PROLOG, to evaluate the usefulness of some of the anticipated functions. While this functioned well on a small scale, it was thought that the C-based NEXPERT would provide a much more comfortable and speedy development environment, and be easier to integrate with other mine-site software.

The inference procedure is an opportunistic mixture of forward and backward chaining(fig 7). Data driven inferencing uses initial data to stack an agenda with useful rules. These are resolved, sometimes through backward chaining. As each of these fires or fails, it too adds to the agenda, thereby plotting a course through the knowledge base. Warning and informational rules are free to fire if they are triggered by the accumulating data. Object classes determine the priority of rules, and dictate which will be fired first.

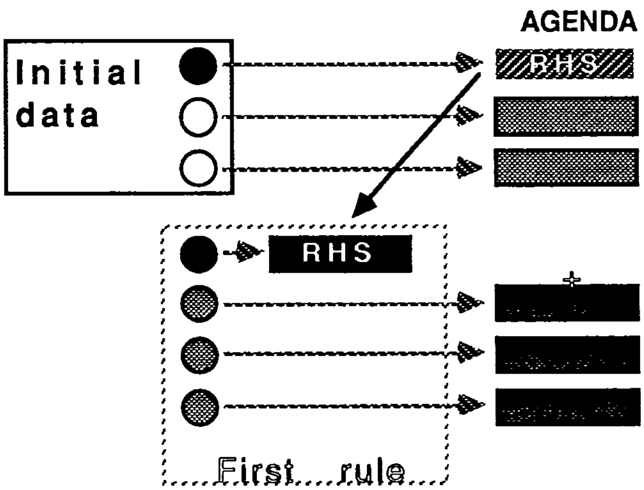

Figure 7

The user has the opportunity at any time to examine what is on the system's agenda by selecting an item from the "Explain" sub-menu.

Initial Problems

The traditional "Expert System" interface was the first problem encountered. The "Volunteer" and "Suggest" options available in the standard interface were daunting to the personnel who tested the initial prototype. This showed the necessity of creating a more intuitive interface which more closely modelled the way mine emergencies are handled. Explanations were supported only in the graphic network, which was difficult to operate, let alone comprehend.

Most of the menus offered were overloaded with options. The initial system had too many choices in the menus. People at the beginning of a problem were confronted with final solutions, or worse, control variables. Correcting this involved 1) partitioning of the knowledge base to bring in rules and objects only when relevant and 2) a classification system for objects allowing them to be selectively displayed.

Rampant , disordered forward chaining caused advice to appear in counter-intuitive order. Originally all rules had identical priorities, and fired in forward chaining mode in order of writing. Some framework had to be created using classes to allow the inference engine to fire "important" rules first.

Rules which had "NOTKNOWN" values in them tended to get lost in the system. It was seen to be necessary to highlight such objects and to try to get the required information and refire the rules when possible.

Due to the "Hypothesis" system incorporated into NEXPERT, there was no easy way to follow a forward chained trail back to its origins in the rule network. A special browsing routine had to be written to facilitate this in the ECAS interface.

Explanations had to be parsed to make them more legible, and even so, features put in to control inferencing needed to be hidden or effectively translated. Enough room had to be allowed so that some of the particularly verbose object names could be displayed in their entirety.

Customising ECAS

ECAS is a rule-driven system, and the questions it asks are generally required to satisfy a rule. To function properly, the system must have its queries answered. Where the answer comes from depends on the particular mine. On simple sites, most responses may have to come from the user. However ECAS is designed to attempt to gain information from computerised sources if available. Using the "order of sources" metaslot, it is possible to specify a search path for data. External programs and databases can be searched for, and if they are not available, the system will prompt the user. This means that the system functions similarly no matter how many peripheral information sources are available.

To function correctly in different locations and in mines of varying scale, ECAS must be able to cope with a range of mine-specific parameters. To accommodate this, the system is designed to take advantage of a local mine database. This can include :

* The physical characteristics of the mine , e.g. layout, ventilation

* Local procedures for evacuations etc

* Catalog of facilities, e.g. whether liquid nitrogen is readily available

The use of a database minimises the requirement to alter the rulebase. In general the system can decide whether some action is necessary, and can leave the fine points of the procedure to the database.

Access to a database can speed up the system, if questions normally presented to the user can be directly answered from the database. This will particularly aid the novice user, who may not have a good working knowledge of the minesite.

The type of hardware is also variable. While the initial response system is designed to be universal available, some larger facilities may desire to have a larger system with better integration with minesite technology. From a practical point of view the use of more powerful hardware will allow more adventurous integration with other mine software, though the fine connections will have to the customised to each individual software application.

The training system will need to have additional facilities to help in simulating predetermined emergencies, and perhaps to allow recording of scores for individual classes, etc. However the basic knowledge should be identical in each configuration.

Conclusions

All mine personnel who have seen the ECAS prototype in operation have been favourable impressed, and have indicated that they see a future for this type of Expert System product at the minesite. Mine rescue and emergency control is a knowledge-intensive domain where, fortunately, there is little chance to put training to the test. The ability to have this knowledge available in a usable form when the critical need arises is seen not only as utilitarian , but also reassuring. It is hoped that having a familiar system used in training available during an emergency will mean that fewer rash errors are made initially, and afterwards that experienced personnel will be freed to devote their energies to more complex activities, and be able to delegate the more mechanical management duties to the system.

Acknowledgements

The following bodies are acknowledged for their help and assistance in the work of this project.

National Energy Research, Development and Demonstration Council, the funding body for this project.

University of Wollongong and Southern Mines Rescue Station, sources of expert knowledge in the field of underground mine emergencies.

Bellambi Coal Company, a place to trial the ideas.

Digital Equipment Corporation (Australia) Pty Ltd, help and assistance in developing the initial prototype.

The Australian Underground Coal Mining Industry for their continued help, assistance and enthusiasm.

Inquiries

If you have any queries regarding ECAS please contact
John Aubrey *or* Zoltan Nemes-Nemeth
ACIRL
PO Box 83, North Ryde
Australia 2133

Bibliography

Aubrey, J. & Nemes-Nemeth, Z. (1988) "*Developing an Emergency Control Advisor System*" Proceedings DECUS Australia Symposium.
Buchanan, B.G. & Shortcliffe, E.H. (1984) "*Rule-Based Expert Systems*" Addison-Wesley
Englemore, R. & Morgan, T. (1988) "*Blackboard Systems*" Addison-Wesley
Hayes-Roth, F. & Waterman, D.A. & Lenat, D.B. (1983) *Building Expert Systems* Addison-Wesley
Strang, J. & MacKenzie-Wood, P. (1985) "*A manual on Mines Rescue, Safety and Gas Detection*" Weston & Co.
Harmon, P. & King, D. (1985) "*Artificial Intelligence in Business*" Wiley Press

References

Hamment, J. (1987) "*Development of an Expert System for Hazard/Emergency Control and Training*" Project Knowledge Acquisition Report #1