TRIAL OF THE "HAZID" TOOL FOR COMPUTER-BASED HAZOP EMULATION ON A MEDIUM-SIZED INDUSTRIAL PLANT

S.A. McCoy and P.W.H. Chung
GRADIENTS Research Group, Department of Computer Science, Loughborough University, Loughborough, Leics. LE11 3TU.

The HAZID knowledge-based system was developed as part of the EC-funded "STOPHAZ" project, whose brief was to investigate the use of software tools for improving safety in chemical process plants. HAZID models the behaviour of a plant qualitatively, so that the potential hazards of equipment failures can be detected. The models in HAZID are based around the signed directed graph, and hazard identification is driven by a systematic HAZOP-style examination of the causes of all process variable deviations in the model. The potential causes are found by considering all possible fault propagation paths in the plant model being considered, using exhaustive graph search techniques.

This paper describes a trial of the HAZID software on the design information from a real medium-sized plant, which has been built and is currently in operation. The objectives were to demonstrate electronic transfer of plant design data from the client's CAD database system to HAZID, to present the results of the HAZID analysis in a structured form as a spreadsheet file, and to show that this type of search-oriented knowledge based system can perform well on a much larger test case plant than had previously been attempted.

The trial was successful in respect of each of these objectives, and also provided valuable information to guide future development of the system, which we hope will be made possible by increased interest from potential customers for the HAZID tool.

Keywords: Hazard Identification, HAZOP, Qualitative Modelling, Case Study, HAZID, Knowledge-Based System.

INTRODUCTION

The main purpose of this paper is to describe a case study in which a computer-based tool for HAZOP emulation, "HAZID", was used to identify hazards in a medium-sized continuous process plant design. The trial of the software involved extracting data electronically from a client's intelligent CAD system, analysing the plant design for hazards, and presenting the results in spreadsheet form.

Firstly, HAZOP is discussed, in order to set a background for the work done on computer emulation of this important technique. The HAZID tool is then introduced as a product of the STOPHAZ project. The tool, the models it uses and the HAZOP algorithm used to identify hazards, are outlined briefly, and some of the notable weaknesses of the tool are identified. After a description of the case study plant under examination, an account is given of the data transfer into HAZID, the production of results and the integration of the various software components. Finally, the results of the trial are briefly discussed, with an emphasis on what the future may hold for HAZID itself.

BACKGROUND

The Hazard and Operability (HAZOP) Study[1] occupies a dominant position in the broad spectrum of techniques used for the identification of process hazards. It has the advantages of being a highly methodical, systematic and effective way of examining all aspects of the detailed plant design; it is also widely known and recognised, with a distinctive reporting format which lends itself to documentation of the process. In view of the fact that submission of a safety case for many types of installation requires a fully documented hazard identification study to have been carried out, HAZOP is often indicated as the method of choice.

The principal difficulty with HAZOP is the time and expense involved – it demands the commitment of a number of technical specialists from multiple engineering disciplines, over an extended period of time, in order to examine a plant design in full. There is therefore a strong economic argument for developing computer-based tools to reduce the time and effort expended in these studies. In view of the fact that HAZOP is a highly creative process, it must be emphasised that the HAZOP team will not be made redundant by the development of improved software – the objective is merely to reduce the time spent by the team in the more routine activities related to HAZOP.

Aside from the obvious role that computers play in facilitating CAD and intelligent CAD (iCAD) systems, with the associated access to plant data which that implies, tools for the documentation and management of the HAZOP meetings have been successfully marketed for a number of years now. These provide help for the HAZOP team "secretary" and the team "leader" in their roles of recording discussions, actions and results of the meetings, and in following up the execution of those agreed actions.

A more ambitious goal is to model the behaviour of the plant under consideration and, by using simulation, predict the possible hazards which may arise in the plant. This approach seeks to emulate some of the reasoning which people do in a traditional HAZOP meeting. The types of simulation models required for this sort of application must allow non-steady states in the plant, as these are the states of interest for identification of hazardous scenarios. The models must also cover a wide variety of possible plant states in either normal or abnormal scenarios. Numerically precise values for process variables may not be a realistic or even desirable possibility in this case. For these reasons, qualitative reasoning techniques are often used.

The most common type of qualitative model used in this domain links process variables together with simple connections which specify a direct or reverse influence direction between disturbances of one variable and the next. Initiating faults in the plant are then linked to deviations in process variables and other deviations are linked to final hazards or consequences. By tracing the paths of propagation through the plant, it is possible to discover links between faults and consequences which may be remotely separated in the plant, via a sequence of deviations which mediate the propagation. The "signed directed graph" (SDG) is one representation for the qualitative linkages described here which has been used quite often in previous attempts to model chemical plants.[2,3,4] Other methods include so-called functional equations, decision tables, etc. A review of previous attempts to model plants in this way is given by Lees,[5] and in a previous paper on the HAZID tool.[6]

The most usual way of tackling plants which contain more than a couple of equipment items is to break down the plant into a number of "units", each of which is modelled by a "unit model". The units present are considered to be interconnected by some means – usually by virtue of process streams which flow between them.

THE STOPHAZ PROJECT AND HAZID SOFTWARE TOOL

The "STOPHAZ" Project (Software Tools for Operability and Hazard Studies) was funded by the European Commission (EC) as part of the ESPRIT Research Programme, between 1993 and 1996. Ten partner organisations took part in the project. These were: Aspentech, Bureau Veritas, Hyprotech, ICI Engineering, Intrasoft, Loughborough University, SfK, Snamprogetti, TXT and VTT. The objective of the project was to investigate novel software tools to improve chemical process safety. There were three main strands of this work:

- An engineering design "hyper-book" for design engineers to use in converting flowsheets to engineering line diagrams (ELDs).
- Tools to facilitate the refinement of plant operating procedures.
- Hazard identification by emulation of the HAZOP study method.

This last strand gave rise to the HAZID software package, a knowledge-based system using SDG models to model the behaviour of the process equipment in a plant. The aim of HAZID is to identify the hazards and their possible causes in a plant using a qualitative, SDG-based, model of the plant. It does this by emulating the procedure followed in a HAZOP study. That is, it examines the possible causes and consequences of variable deviations in the plant, finding links between remote events by fault propagation. The causes typically correspond to equipment failures and the consequences are the final hazardous events which may be caused in the scenario.

The most likely application for the current version of HAZID would be for a single process design engineer to use the tool as an aid to reviewing the plant design prior to a conventional HAZOP study. In this scenario, HAZID could provide a cost-effective way of identifying major safety concerns, for discussion in the study, or for fixing in advance of the HAZOP meetings. Because HAZID was first conceived as a HAZOP emulation tool, the level of detail required for the plant description consists of the types of equipment items which would be specified on an Engineering Line Diagram (ELD) or Piping and Instrumentation Diagram (P&ID). Therefore, the present library of models in HAZID is not well suited to evaluation of designs from earlier stages, such as conceptual design, Process Flow Diagrams (PFDs), etc. Nevertheless, further development of the model library could extend the use of essentially the same HAZID analysis program to earlier stages, facilitating its use as a tool to evaluate alternative conceptual designs, for example. We feel that once the safety analyses performed by tools such as HAZID become relatively easier (and therefore cheaper) to perform, there is scope for HAZOP-style examinations of process designs at many stages in the lifecycle.

ARCHITECTURE

As shown in Figure 1, HAZID is composed of a number of sub-systems, which each contribute to the functionality provided by the whole. Some of these are:

- A **Parser**, which reads files containing plant definitions or libraries of models used by HAZID, converting these into appropriate internal data structures.

- **Internal Plant Model** – This is the form in which the plant description is stored within HAZID. It consists of a list of equipment model instances, each with its own set of "slots" defining particular data concerning that unit.
- **Template and Equipment Models** – The equipment models used by HAZID are stored in an external library file and read in through the parser, into a list of unit models. Each unit model may make use of so-called "template" models, which define commonly re-used groups of SDG arcs in a more concise way. The template models are also read into HAZID from an external library file.
- **Fluid Model System** – During STOPHAZ, the HAZID system had access to physical properties calculations performed in external packages such as Aspentech Properties Plus, or Hyprotech HYSIM. Rules can be attached to the arcs in the SDG models of equipment to make them conditional on the properties of particular fluids in the plant. The interface to external packages is now inoperative because of the development of newer versions of the commercial software packages.
- **Link to External Database** – Access was also provided to a graphical tool for preparing outline plant description drawings during STOPHAZ. This link is no longer working, as it relies on a 16-bit Windows environment for operation of the underlying database.
- **Configuration Rules –** A number of rules for detecting poorly configured design features in a plant were also implemented in STOPHAZ. These are invoked whenever a plant model is examined using the HAZOP methodology, and add extra records to the HAZOP report which is produced.
- **HAZOP Algorithm and Report Generator –** This algorithm drives the examination of each deviation in the plant, filters the results of this work and feeds the refined results to the report generator. The result is a text file containing the deviations, causes and consequences for each unique scenario identified.

EQUIPMENT MODELS

Within HAZID, each equipment type is modelled by a unit model, which defines the arcs of an SDG representing the links between variables in that unit. Initial equipment failures are modelled as "faults" which cause specific deviations in those variables, and final hazards are modelled as "consequences" which may arise from specified deviations. In constructing a plant model, specified unit models are instantiated and connected together, forming a large, inter-connected SDG representing the structure of the whole plant.

This approach encourages the development of a smallish number of well-understood equipment models, each corresponding to a commonly used piece of equipment. In HAZID, these models are arranged into a hierarchy of equipment types, with more generic models near the root of the tree and more specialised models towards the leaves. Inheritance is conveniently used in this hierarchy to allow re-use of basic models in the definition of more specialised ones. A fuller description of the model system in HAZID, as well as the model hierarchy, is given elsewhere.[7]

Unit models are each provided with a number of slots giving information used in the definition of that equipment type. However, the main purpose is to define a set of SDG arcs giving the relations between process variables in the unit. The variables are always located at "ports" in the equipment item. There are three types of port: input ports, output ports and internal ports – each type may be associated with a number of process variables.

An additional level of flexibility is used in the models of HAZID, for groups of arcs which are commonly repeated in various unit models. The "template" feature allows such arcs to be grouped together and named, and for instances of the templates to be used in a variety of unit models.

A final element of the plant model is that of the process fluids. HAZID models allow for process fluids to be specified at every port in the plant, by giving a list of the chemical substances present and their compositions. Associated with each fluid is a pressure and temperature, and a flowrate for input and output ports. Usually, fluid data relating to a number of ports in upstream locations may be specified in this way, and HAZID will go through the plant propagating fluid information wherever appropriate to points downstream.

HAZOP ALGORITHM

The algorithm used by HAZID, to examine a plant model for hazards, is based on the overall procedure used for HAZOP studies of plant design drawings.

First of all, HAZID groups equipment items into "lines", in order to sort them all into an order for examination. It does this by tracing potential flow paths through the plant model, from inlets to major units or outlets, and from major units to outlets. Within each line, the first and last units are examined first, followed by the intermediate equipment items in order.

Once sorted, each unit from the overall list is then examined in order. All appropriate deviations of variables at each of the ports belonging to that unit are generated in turn. For each deviation, HAZID searches backwards through the SDG to find causes of that deviation, then looks forwards in the SDG to find any immediate consequences (hazards). Any immediate consequences of the deviations along the paths between the causes and the deviation under consideration, are also added. The results at this stage are a record of the deviation, its possible causes and associated consequences.

After producing these results for all units in the plant, HAZID filters them to remove repeated scenarios, deviations without any consequences, etc. It also makes use of rules involving properties of the process fluids at this stage, to eliminate scenarios which are in fact not feasible for the specific plant and scenario considered (e.g. flammability is a precondition for any hazard involving fire). The operation of this rule system (the "Fluid Model System") is described in an earlier paper.[8]

Finally, the filtered results are printed out to an output file in a format resembling a traditional HAZOP report table.

EVALUATION OF HAZID IN STOPHAZ

The evaluation of the HAZID tool is described in detail in a previous paper.[9] The main method used to inform the development of the models and the software in the STOPHAZ project was to look at the results produced by the tool when applied to case study plants. Results were criticised by experts in process safety and the HAZOP study technique, to provide pointers for improvement. A specific set of case study plants (called the "learning set") was collected together to facilitate this work. Another technique which seemed useful to elicit knowledge for improving specific equipment models, was to get a small group of process experts to meet as a "mini-HAZOP" study, in order to analyse a type of equipment item, and identify all the potential failures or hazards which could be associated with it.

In addition to this development work, later versions of the HAZID software were applied to previously unseen case studies (the "test set"), to evaluate their performance. The results produced were compared to those of conventional HAZOP studies on the same plants. Performance was quite variable, but it was found that HAZID could identify between 33% and 60% of the scenarios identified by the conventional HAZOP meeting.

POINTERS FOR IMPROVEMENT OF HAZID

A number of areas of work were identified for future improvement in HAZID, during STOPHAZ and afterwards. Many of these are documented elsewhere.[10] Some of the more important topics are:

- Automated access to electronic plant description data, as stored in iCAD systems. This is a priority to make use of the tool a realistic proposition in industry, where manual re-keying of process information is not a viable option.
- Acceptability of HAZID to users – There is a need to demonstrate that the HAZID system is a desirable and reliable tool which will save industry users time and money, whilst also helping them to eliminate hazards and improve their safety performance.
- Output reporting format – The text file produced by the STOPHAZ version of HAZID is not structured enough. There is a need to put output in a more structured format, to allow further processing of results by HAZID or other programs. Additionally, alternatives to the HAZOP table format may be considered, or a more interactive type of system could be devised for hazard identification.
- Robust identification of any protections and devices present, which may prevent a scenario from occurring. This requires some further development of the model systems in HAZID, to capture what alarms, trips, interlocks, etc., actually do in a plant.
- Separate configuration rules module – It may be appropriate to identify design glitches before the HAZOP emulation tool is invoked, rather than knitting these results into the fabric of the HAZID output report itself.
- Model quality and completeness – A more substantial investment in systematic improvement of the unit models is needed, to make the tool of more interest to industry.
- Flexible models – The model system as implemented for STOPHAZ does not allow units to have arbitrary numbers of ports and therefore makes the definition of specialised vessels, not already in the unit model library, rather difficult.
- Fluid Model System – Improvements are needed for this system, as well as a wider range of fluid rules for eliminating infeasible hazards from the results.
- Consequence evaluation/classification – These issues may be tackled in an attempt to focus attention on the most important hazards detected by the system.
- Richer model system – It may be possible to capture concepts such as state-dependent behaviour, using a richer representational scheme than the SDG. If this were the case, the range of scenarios which HAZID would be able to model would be greatly increased.

CASE STUDY

There were three main technical objectives in tackling a new demonstration of the capabilities of HAZID:

- To access the plant model from a real iCAD system containing the full set of plant data, in this case AspenZyqad.
- To demonstrate that HAZID could successfully examine such a large plant, as the case study plant was several times larger than previously examined case studies.
- To produce the same HAZOP-style results table in a more structured format than plain text. The format chosen, for this trial, was as a Microsoft Excel spreadsheet file.

The effect of successfully achieving these objectives would be to demonstrate HAZID's potential to a possible customer interested in further developing the tool. In any case, the experience would also outline where the priorities for further development of HAZID should lie. Therefore, it was judged very important to note any practical difficulties encountered in achieving these three goals.

THE TEST CASE PLANT

The plant examined during this case study trial (which took place in summer 2000) was a vent gas scrubber plant designed and operated by BNFL. It contains a packed bed scrubbing column and some other, fairly conventional, vessels and other equipment, processing aqueous streams.

In size, this plant is significantly larger than other plants previously examined by HAZID. When transferred into HAZID, the plant model of the scrubber plant contains a total of 558 plant items, pipes, valves and control valves. This compares to the 127 units in one of the larger "learning set" problems examined during the STOPHAZ project (a benzene production plant). When analysed by HAZID, 121 of the units in the scrubber plant are examined in detail, as compared to 29 units in the case of the benzene plant.

The fluids used in the scrubber plant are mainly water-based, which is something of a change from the previous test cases, which have often been petrochemical in nature.

All the data relating to the scrubber plant is present in BNFL's AspenZyqad database system. Interfaces to this data are provided through the company's CAD system (based on AutoCAD), and when the scrubber plant is displayed in the form of piping and instrumentation diagrams (P&IDs), it takes up 7 sheets. In this form, many of the streams between units cross boundaries between P&ID sheets.

DATA TRANSFER INTO HAZID

The plant description required by HAZID describes plant items in terms of "instances" of equipment models connected to one another using process stream connections. This connectivity model of the plant is augmented by various types of state information (e.g. the normal status of valves, operational status of pumps, etc.), information about the pressures, temperatures and fluid components present at various points around the plant, and information on maximum and minimum design temperatures and pressures for the equipment in the plant.

Using this information, in conjunction with equipment models stored in a library of unit models, HAZID can build up a representation of the plant in terms of a signed directed graph (SDG) model of all the variables in the plant.

The approach used to transfer data from the AspenZyqad database into HAZID is illustrated by Figure 2. Since the HAZID development team at Loughborough University did not have access to AspenZyqad software, the first step was for BNFL to dump the contents of their database into a file of a neutral, agreed format. This operation is supported by the software they use, so this step did not present any technical challenges to us. The second step is to take the file and read the data in it, extracting any information of use and discarding the rest, in order to compile the information needed to produce a HAZID format plant description file. After this data has been transferred, the job of analysing the plant for hazards using HAZID can begin.

The extraction of data was achieved using a program written in the computer language PERL (the script was called "conversion") to read the database file into a large table, from which relevant data could be extracted. Using these data, definitions of units could then be produced and written out into a new HAZID input file. PERL was chosen for this task because it is well-suited to text processing and allows rapid prototyping of a solution to the problem.

Every equipment item in the plant corresponds to an object in the database file, with a number of associated data fields. However, there are many objects in the file which do not correspond to equipment items, and so must be ignored. In any case, a subset of objects can be found in the file, corresponding to the plant items of interest to HAZID.

The process connectivity of the plant is defined in the database file in terms of PIPING-SYSTEM objects, which each define a FROM object and a TO object, and a list of objects which appear in between the two points. The objects usually correspond to equipment items in the plant, but may also include NODEs, which are points at which branches in the piping may occur. For major equipment items which may have many inlets and/or outlets, such as vessels or columns, the source or destination objects used are usually NOZZLEs, which are named objects associated with the large equipment item. It is important to map the NOZZLEs onto appropriate named ports in the HAZID unit models. Using this definition of connectivity (i.e. the PIPING-SYSTEMs) the "conversion" program pieces together how units in the plant are connected together.

The PIPING-SYSTEM objects also contain information about the temperature and pressure, and the type of fluids present. This information is also extracted from the data table, converted into appropriate measurement units where necessary, and inserted into the HAZID input file.

In larger equipment items, upper and lower design temperature and pressure limits are also provided. As HAZID can make use of this information in eliminating some hazards, the values are extracted wherever possible.

It is important to know the intended state of a valve, so that the HAZID model used can be adjusted appropriately. It is also important to know whether a pump is intended to be running normally, or on stand-by. Initially, there was no way of getting this information out of the database in a straightforward way. However, later, overlay objects were introduced for all valves which are intended to be normally closed, which made it possible to infer the default states of all valves in the plant. A similar arrangement is envisaged for the pumps problem, but so far this is accomplished using a look-up table giving the default states of each pump in the plant.

One important function of the "conversion" script is to map from names of equipment types in the AspenZyqad system to equivalent models in the HAZID library.

This is done using a look-up table in an external file, which is read in by the conversion program when it starts.

Equipment items with multiple input or output ports give rise to a similar problem. The nozzle connections in the AspenZyqad data must be mapped onto appropriate named ports belonging to the HAZID models used. Again, this is achieved so-far using a look-up table.

REPORT PRODUCTION IN SPREADSHEET FORM

In previous versions of HAZID, the main output format was a plain text report, containing results in four columns: deviation, causes, consequences and protections. It was felt that more structure was needed, so that these results could be post-processed by computer and/or displayed in a more versatile manner.

For this reason, an additional output from the HAZID analysis of a plant was introduced – to produce output as a Microsoft Excel spreadsheet file. Some modification of the core HAZID program code (in C++) was undertaken to make this happen.

Essentially the same information was presented in the new Excel report format, a sample of which is given as Figure 3.

SOFTWARE INTEGRATION

One additional benefit of this work was the development of a Visual Basic (VB) "wrapper" application, to launch the conversion and HAZID programs, and display the results produced by opening the results file in Microsoft Excel. This VB program was written quite quickly and has no impact on the performance of either the PERL script or the C++ program.

The VB wrapper provides a Graphical User Interface (GUI) to coordinate the execution of HAZID and the other components, while also allowing some of the configurable options in HAZID to be changed interactively. In avoiding the use of the text-based menu system of the central HAZID program, it presents a more "user-friendly" front-end to the system.

RESULTS OF TRIAL

The full analysis of the scrubber plant took 91 minutes and produced a 254KB spreadsheet file with 367 identified potential hazards. The time taken included: processing the intermediate data file, conducting a HAZOP-style examination of all the deviations in the plant model, filtering the results of the analysis and printing the filtered results to file. The hardware on which these results were produced was a mid-range desktop PC (Pentium P3, 600MHz, with 128MB of physical memory) running Microsoft NT4. The programs described here have also been successfully used in other 32-bit Windows environments.

HAZID is very demanding in its use of memory, because it makes use of a breadth-first search technique and has to store information about all the search paths it is considering. Therefore, virtual memory setup can be a problem in some versions of Windows.

During the trial of HAZID, a number of incremental improvements were made – to the database files containing the plant data and to the various parts of HAZID. Apart from the addition of the conversion program and the Excel output, the central HAZID program was enhanced by the addition of a new filter, which groups together similar causes of a scenario in the report, wherever those causes appear in similar equipment, so that the volume of the report can be reduced.

The input data was improved twice during the trial. Firstly, overlays were added to the valve objects in the file, which allowed the status of valves to be inferred by the conversion program. Then, some of the connectivity problems in the plant were fixed and sheet interconnectors (objects defining where a stream flows from one drawing onto another) were eliminated wherever possible from the database. Both these improvements meant that the file could be processed to produce a more accurate model of the actual plant design within HAZID.

Some of the above improvements had the effect of reducing the volume of the output files produced, by condensing repeated results or eliminating "nonsense" results. Some had the effect of connecting the plant units together more tightly and therefore increasing the time needed to search through the plant. The "headline" results given above relate only to the latest version of the data file, with all the new enhancements included.

The overall reaction of BNFL to the format of results produced by HAZID was positive. In meetings with the company, there has always been a great deal of interest in how HAZID could be further improved, to make it a genuinely useful tool for engineers at the design and safety verification stages of process design.

DISCUSSION AND FUTURE WORK

The new trial of HAZID has so far demonstrated that the electronic transfer of data, from a client database system into the format required by HAZID, is feasible. It has also shown that the search-based AI techniques deployed in HAZID are not defeated by the scale of problem posed in this, a medium-sized industrial plant design – and demonstrated this to a potential user of the tool. The results produced by the tool are now presented in a more attractive and structured format than before, which is certainly a valuable move. Perhaps most importantly, the experience of tackling this problem has identified some of the issues which need to be tackled more comprehensively in commercialising the HAZID software.

There is an important need for the databases of HAZID and the client CAD systems from which it draws its input to be harmonised. Thus, the names of models in the HAZID library should correspond in a predictable way with the names of objects in the intelligent CAD system. So far, this problem has been tackled by the use of a look-up table, which maps the type ("PARENT") of an object in the database file to a specific HAZID model name. In future, this may not be needed, as the two databases will use the same names to mean the same things.

The translation of connectivity in one database to the form that this information is expressed in HAZID input files, is an issue which needs to be addressed anew, for each client system HAZID is required to draw data from. Certainly, the standard method in AspenZyqad is unlikely to coincide with the method used in other environments.

How to deal with major units, in which there are multiple inputs or outputs, is another difficulty. At the moment, HAZID requires that all the ports belonging to an equipment model are defined (and named) in the unit model library, before that unit model can be used in a plant model. There is a need for further flexibility in the HAZID modelling language to support arbitrary numbers of ports, as well as a consistent approach to finding out how to model those ports, find out if they are carrying liquid or gas, what type of inlet or outlet fitting is used, etc.

In the HAZID trial described above, the issue of major units with many ports was tackled by defining a number of new generic models of vessels and tanks, which had large numbers of predefined inlets and outlet ports of various types. It then used a look-up table to allocate the particular nozzle seen in the intermediate data file to a named port belonging to the appropriate model. Clearly this is a poor solution in the longer term, and it imposes port names designed for a HAZID model onto the user of a CAD system in which the ports may be known by completely different names (e.g. C101-NOZZLE-5, etc.).

The fluid information which it was possible to glean from the client database in this case study was rather different from the type of data which was assumed by the earlier design of HAZID. In the BNFL plant, each PIPING-SYSTEM had an attribute (PIPE-DESC) which described the fluid type present – but the level of information given was in terms of a generic code, rather than a list of chemical species. The fluid information model used in HAZID has tended to assume that there would be information about all the chemical compounds present in a stream, giving their relative proportions, as well as the flowrate, temperature and pressure of the whole stream. This was just not the case in the scrubber plant, and might well not be the case in many other industrial case studies. Therefore, a review of the issue of how best (most flexibly?) to represent fluid information in the plant model may be required for further work on HAZID – with the over-riding priority that the information given must allow inference about safety and environmental hazards.

One of the main objectives of the trial was to renew awareness of the HAZID system, with a view to getting potential customers interested in further development of the functionality it offers. The trial has demonstrated once again, to us as developers and to BNFL as potential customers, that the tool is worthwhile developing and that the practical problems in commercialising it are realistically solvable.

The intention now is to get an interested group of potential user organisations together to fund further development of the system. Each member of such a consortium would pay a flat price to the development team, and in return would gain a licence to use the software produced for a certain number of years, as well as technical advice in how to make the best use of HAZID. The members of the consortium would be able to influence the development of an important piece of new software, as well as being able to use HAZID, almost from Day 1. The development team would benefit from valuable industry feedback and guidance to inform the development of HAZID, as well as having the opportunity to commercialise the tool when an appropriate level of functionality has been achieved.

CONCLUSIONS

This paper started by giving the rationale for developing model-based systems for hazard identification (such as HAZID) – namely, to save time and money in the verification of process plant safety, and particularly in making HAZOP studies more effective. The qualitative modelling system in HAZID, as it was developed for the STOPHAZ project, was then described in outline.

The main concern of the paper has been the trial of HAZID conducted in the summer of 2000, in conjunction with BNFL. The objectives of this trial were to achieve electronic transfer of plant data from an intelligent CAD system into the HAZID program, to analyse the plant as given and produce the results of the analysis in a structured format, as a spreadsheet file.

All these objectives were achieved, and HAZID demonstrated itself well able to handle the size of the case study plant (which is significantly larger than previously analysed plant models). The format of the output produced was also found to be acceptable to the client.

The trial gave rise to several improvements in the HAZID software and also in the information kept by BNFL in their database, although the intention of the trial was never to act as a test of the database system. A number of valuable pointers for further work on HAZID have also been identified, and are outlined in the "Discussion" section above.

On the whole, this trial (which was relatively small in scale, compared to the work done during the STOPHAZ project) has been a successful experience for us as developers. It has also succeeded in reviving the development of HAZID as an on-going concern – and we look forward to working with other partner organisations in furthering the cause of process hazard identification by computer.

ACKNOWLEDGEMENTS

The authors acknowledge the support of HAZID Technologies Ltd., and would also like to thank BNFL for their cooperation in ensuring the success of the trial described here.

REFERENCES

1. Chemical Industries Association Limited, 1977, A Guide to Hazard and Operability Studies, *published by Chemical Industry Safety and Health Council of the Chemical Industries Association*.
2. Chung, P.W.H., 1993, Qualitative analysis of process plant behaviour, *6th Int. Conf. Ind. Eng. Apps. AI, published by Gordon and Breach*, 277-283.
3. Vaidhyanathan, R., Venkatasubramanian, V. and Dyke, F.T., 1996, HAZOPExpert: An expert system for automating HAZOP analysis, *Proc. Safety Prog.*, 15:80-88.
4. Iri, M., Aoki, K., O'Shima, E. and Matsuyama, H., 1979, An algorithm for diagnosis of system failures in the chemical process, *Comp. Chem. Eng.*, 3:489-493.
5. Lees, F.P., 1996, Loss Prevention in the Process Industries (2nd edition), *published by Butterworth Heinemann*.
6. McCoy, S.A., Wakeman, S.J., Larkin, F.D., Jefferson, M.L., Chung, P.W.H., Rushton, A.G., Lees, F.P. and Heino, P.M., 1999, HAZID, A Computer Aid For Hazard Identification 1. The STOPHAZ Package and the HAZID Code: An Overview, the Issues and the Structure, *Trans. IChemE., Part B, (P.S.E.P.)*, 77: 317-327.
7. McCoy, S.A., Wakeman, S.J., Larkin, F.D., Chung, P.W.H., Rushton, A.G., and Lees, F.P., 1999, HAZID, A Computer Aid For Hazard Identification 2. Unit Model System, *Trans. IChemE., Part B, (P.S.E.P.)*, 77:328-334.
8. McCoy, S.A., Wakeman, S.J., Larkin, F.D., Chung, P.W.H., Rushton, A.G., Lees, F.P. and Heino, P.M., 1999, HAZID, A Computer Aid For Hazard Identification 3. The Fluid Model and Consequence Evaluation Systems, *Trans. IChemE., Part B, (P.S.E.P.)*, 77:335-353.
9. McCoy, S.A., Wakeman, S.J., Larkin, F.D., Chung, P.W.H., Rushton, A.G., and Lees, F.P., 2000, HAZID, A Computer Aid For Hazard Identification 4. Learning Set,

Main Study System, Output Quality and Validation Trials, *Trans. IChemE., Part B, (P.S.E.P.)*, 78: 91-119.

10. McCoy, S.A., Wakeman, S.J., Larkin, F.D., Chung, P.W.H., Rushton, A.G., and Lees, F.P., 2000, HAZID, A Computer Aid For Hazard Identification 5. Future Development Topics and Conclusions, *Trans. IChemE., Part B, (P.S.E.P.)*, 78: 120-142.

FIGURES

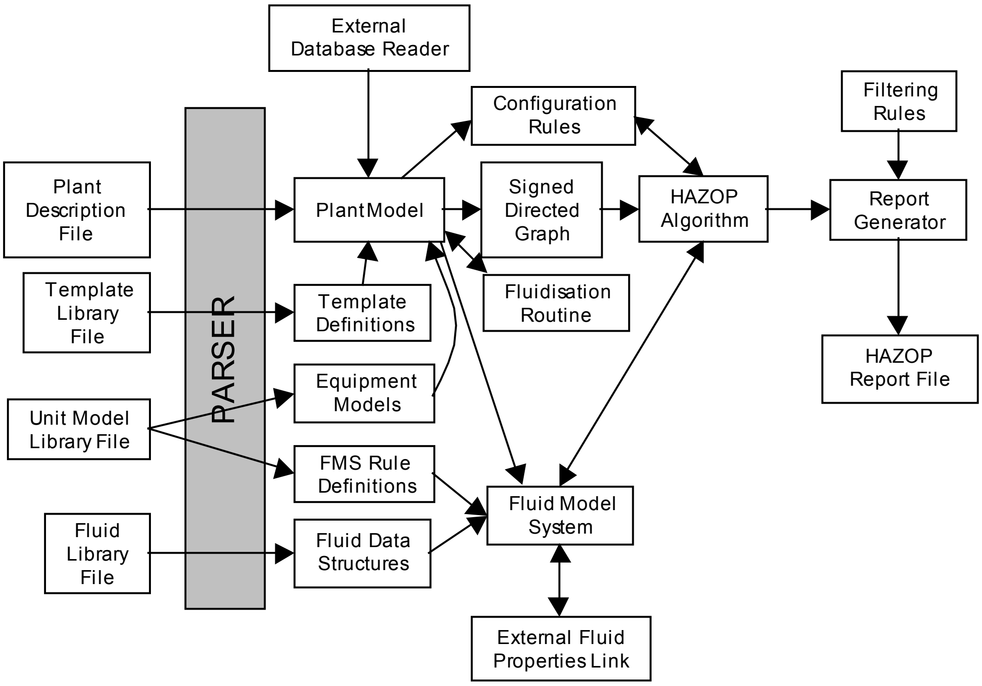


Figure 1: System Architecture of the HAZID Tool

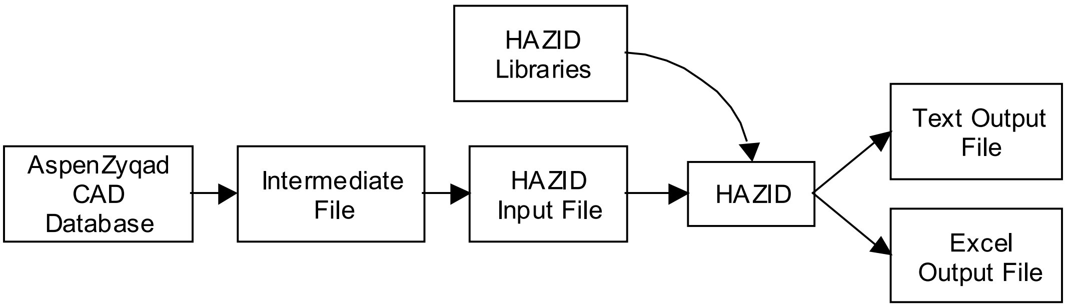
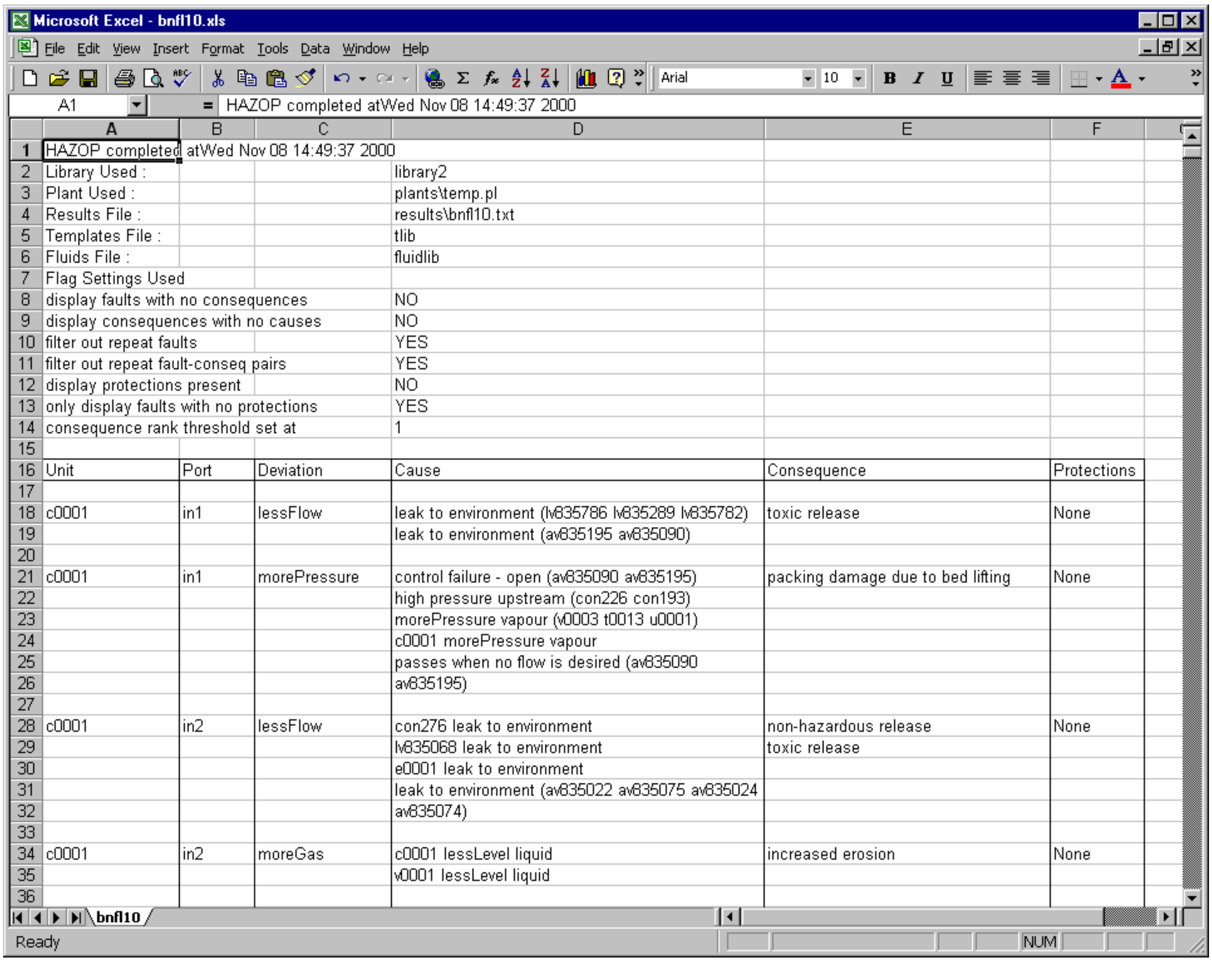


Figure 2: Data Transfer in the HAZID Case Study



A1 = HAZOP completed atWed Nov 08 14:49:37 2000

	A	B	C	D	E	F
1	HAZOP completed atWed Nov 08 14:49:37 2000					
2	Library Used :			library2		
3	Plant Used :			plants\temp.pl		
4	Results File :			results\bnfl10.txt		
5	Templates File :			tlib		
6	Fluids File :			fluidlib		
7	Flag Settings Used					
8	display faults with no consequences			NO		
9	display consequences with no causes			NO		
10	filter out repeat faults			YES		
11	filter out repeat fault-conseq pairs			YES		
12	display protections present			NO		
13	only display faults with no protections			YES		
14	consequence rank threshold set at			1		
15						
16	Unit	Port	Deviation	Cause	Consequence	Protections
17						
18	c0001	in1	lessFlow	leak to environment (lv835786 lv835289 lv835782)	toxic release	None
19				leak to environment (av835195 av835090)		
20						
21	c0001	in1	morePressure	control failure - open (av835090 av835195)	packing damage due to bed lifting	None
22				high pressure upstream (con226 con193)		
23				morePressure vapour (v0003 t0013 u0001)		
24				c0001 morePressure vapour		
25				passes when no flow is desired (av835090		
26				av835195)		
27						
28	c0001	in2	lessFlow	con276 leak to environment	non-hazardous release	None
29				lv835068 leak to environment	toxic release	
30				e0001 leak to environment		
31				leak to environment (av835022 av835075 av835024		
32				av835074)		
33						
34	c0001	in2	moreGas	c0001 lessLevel liquid	increased erosion	None
35				v0001 lessLevel liquid		
36						

Figure 3: Example Spreadsheet Output from HAZID