Computer tools for hazard identification, modelling and analysis

P. Heino[a], A. Poucet[b] and J. Suokas[a]
[a] *VTT, Safety Engineering Laboratory, P.O. Box 656, SF-33101 Tampere (Finland)*
[b] *Joint Research Centre, Institute for Systems Engineering and Informatics, I-21020 Ispra (VA) (Italy)*

(Received April 8, 1991; accepted in revised form July 31, 1991)

Abstract

Plant safety and reliability analysis is a complex task often with a need for computer support. The phases of the analysis are information collection, qualitative hazard identification, system modelling and analysis, and consequence assessment. Computer software tools have been developed for different analysis tasks. They are very useful in the automation of the routine parts of the analysis. However, knowledge-based techniques are needed in the development of more powerful tools. This paper gives a short review of software tools developed for the documentation and calculation tasks of safety and reliability analysis. The main emphasis is on describing and evaluating the possibilities of knowledge engineering to support human reasoning in hazard identification and system modelling. An example of an advanced software environment, STARS, for carrying out multi-level knowledge-based safety and reliability analysis is presented.

1. Introduction

Plant safety and reliability analysis consists in general of the following phases:
1. Collection of plant specific information, definition of the objectives and boundaries of the analysis, and description of the activities and systems to be studied.
2. Qualitative hazard identification in which possible events and scenarios are identified and a preliminary assessment of their hazard potential is made. Methods used in this phase include different preliminary hazard analysis (PHA) methods, hazard and operability analysis (HAZOP), failure modes and effects analysis (FMEA), etc.
3. System modelling and analysis during which detailed logic models, describing the deep causes of the above-mentioned events and scenarios, are con-

Correspondence to: Dr. J. Suokas, VTT, Safety Engineering Laboratory, P.O. Box 656, SF-33101 Tampere (Finland)

structed and analysed. The methods used in this phase are fault tree analysis (FTA), event tree analysis (ETA), Markov analysis and possibly methods for human reliability assessment and dependent failure analysis.
4. Consequence assessment in which the consequences of hazardous events and scenarios are modelled and analysed in detail. In this phase, the physical models for chemical release atmospheric dispersion, fire and explosion are used.

In the overall process of plant safety analysis, a vast amount of data and information is to be elicited and manipulated, and complex calculations have to be performed. It is therefore obvious that computer support is essential.

In this paper, recent developments in computer support for plant safety assessment are discussed. The first computer applications concerned mainly routine calculations of reliability, availability and accident frequency based on fault trees, and the calculation of gas releases and dispersions based on physical models. The next stage was to get assistance for the documentation of an analysis—for example, to draw a fault tree or to document the results of a HAZOP study. The third stage in the development of computer aids for safety analysis were programs aiding the diagnostic tasks in an analysis. This attempt to automate several earlier manual tasks has been continued with improved efficiency using knowledge engineering techniques.

Some of the tools that can be used to perform the analysis in Phases 2 and 3 are described. A short review of conventional computer programs supporting the documentation tasks—writing down HAZOP results, drawing fault trees, etc.—or carrying out mathematical calculations of fault trees is given first. The main emphasis is, however, on describing and evaluating the possibilities of knowledge engineering techniques in more demanding human reasoning tasks, such as hazard identification and system modelling, particularly the construction of fault trees. Consequence assessment (Phase 4) is left out from this article due to the large volume of research and number of applications on that research area. A review on this topic has recently been made by Kakko [1].

Mostly, safety and reliability analysis tools address a particular analysis task. In order to support the whole spectrum of activities, there is a need for an environment designed for computer-aided plant safety assessment. The STARS (Software Tool for Analysis of Reliability and Safety) project discussed in this paper among other topics is one attempt to create such an environment.

2. Hazard identification and qualitative analysis

The early stages of safety analysis aim at the identification of potential hazards relating to the analysed system. Initial safety assessment may even be carried out when only general level information about the plant and the products, raw materials and main reactions is available. The methods used at that stage are mainly based on the application of key words and check lists. In general, those methods can be called PHA methods.

When the first block drawings of a process are available, systematic methods like HAZOP study become applicable. However, the knowledge on the process and its detail is still incomplete, which makes automatic computer-based analysis problematic. Instead, interactive computer programs which support the analyst by presenting relevant information and providing flexible tools for documentation are useful. The intention is to combine the capabilities of the computer program and the human analyst in an efficient way.

2.1 Documentation tools for hazard identification

A simple flexible tool for documentation can be a valuable aid in the routine tasks of HAZOP and other similar methods which produce large amounts of textural documentation. CAFOS [2] was one of the first attempts to that direction. Several programs have become available for that purpose recently with the growth of the personal computing community. These programs are based on the well known text processing and spreadsheet calculation techniques. One example of such commercial microcomputer programs is SAnDoc [3]. Another example is HAZOP-PC [4], which is a commercial personal computer software package to facilitate the performance of hazard analyses for industrial systems using the HAZOP method. Its primary function is to provide means for recording the results of a HAZOP study as the study is conducted. In addition to HAZOP specific word processing capabilities, it includes facilities for using and editing check lists and calculating risk rankings.

2.2 Automatic hazard identification

It would be most desirable to develop such intelligent support tools for hazard identification which in addition to documentation support, could provide diagnostic information and automatically carry out analytic tasks. However, the automatic identification of hazards would require an extensive process data base and a large knowledge base of heuristics for the hazard identification reasoning. The following is a list of typical process data needed in that kind of reasoning:
(1) process topology (process units, components, pipelines);
(2) component characteristics (type, size, material, etc.);
(3) control and protective systems:
(4) process substances and inventories;
(5) chemical reactions; and
(6) phases of batch processes.
The heuristics should include knowledge on the presence of hazards in different combinations of process characteristics and models for the propagation of process parameters through process components.

Even if all the above-listed data and knowledge is available, the results of the automatic study will contain a certain amount of erroneous and irrelevant information due to the lack of background (common sense) knowledge. Furthermore, a great deal of the results will be trivial to an experienced analyst.

A better alternative is to allow user interaction during the computer-aided analysis. When the user has the control, the paths of the analysis leading to trivial results can be easily avoided. Of course, one has to keep in mind that such paths could sometimes unexpectedly lead to non-trivial and important findings. Those findings can, in some cases, turn out to be the most important result of HAZOP studies. Therefore, a carefully designed interactive analysis procedure with flexible facilities for editing the results would seem to be the most promising approach to computer-aided knowledge-based hazard identification.

So far, a few attempts have been made to automate hazard identification to some extent. Parmar and Lees [5] describe a method for modelling fault propagation for the purpose of hazard identification in process systems. Other approaches concerning off-shore platforms and electronic circuits, respectively, are described by Reeves et al. [6] and Lehtela [7]. Some important factors which have slowed down the development in that area summarized below:

1. Limited amount of resources are usually allocated for hazard identification. Therefore, it is difficult to find time for the collection and input of required process data. On the other hand, the variety of existing computer-aided design (CAD) systems prevents the automatic use of design information in a general manner.

2. In general, hazard identification techniques can not achieve their goal by just processing data efficiently. Idea processing in hazard identification is based on heuristics and experience-based information, common sense knowledge often focusing the work. If the use of guide words (words which guide the consideration of the process, plant area, environment, etc.) and data is systematized, there is the danger that hazard identification becomes too much like the methods used in the modelling phase and less capable of identifying such hazards that can not be revealed with a routine walkthrough of the system.

3. When the results of hazard identification are documented by using computer support, a hazard identification data base is created in addition to the paper documents. Such a data base gives the possibility for flexible updating of the analysis when changes take place. However, this requires that the hazard identification software includes flexible documentation facilities. Otherwise the user will prefer to use ordinary text processing software for documentation. This sets additional requirements for the automatic features, too, since the results of automatic identification should contain a reasonable amount of relevant findings to be useful. On the other hand, the results should not contain a significant amount of irrelevant information because then the user has to spend time in cleaning up the results, which decreases the value of the tool as a documentation tool.

2.3 Knowledge-based support for hazard identification

The use of knowledge engineering techniques is one way of developing the principles of automatic hazard identification further. These techniques provide means for the use of incomplete and unstructured information and for reasoning about the relevancy of automatically generated results.

On example of the first developments in knowledge-based hazard identification is HAZOPEX [8,9] which is a research prototype expert system with 350 HAZOP rules. The aim of HAZOPEX is to support the HAZOP analysis of process systems. Relevant deviations from the normal process state are analysed, and their causes and consequences are identified and remedies planned. HAZOPEX is designed for the use of a process designer. The intention is to make HAZOP analyses an integral part of process design. This means earlier identification of defects and errors in the design, and hence better and more economic possibilities for achieving a safe and reliable process system. The effort required for a systematic HAZOP study is reduced to a lower level and the quality of the results is guaranteed.

The piping and instrumentation diagram (PID) and complementary information on the process and its components are required as input. During the analysis there is a HAZOP form on the display where the results of the interactive analysis are presented. The PID of the process is on the screen beneath the HAZOP form and can be accessed easily. The user controls the analysis through an interface based on menus. HAZOPEX goes systematically through all possible deviations in different parts of the process. It generates analysis results and makes suggestions which the user can accept, reject or complement. However, the user can choose the analysis object or the deviation of most interest if the systematic procedure seems unsuitable.

The current version of HAZOPEX was developed for research purposes. Therefore, the flexibility and effectiveness of the user interface do not fulfill the requirements of industrial use. However, the basic principles of interaction have been found appropriate. Symbolics workstation and KEE (Knowledge Engineering Environment) were used as the development environment in order to enable rapid prototyping and testing of the basic ideas. A more standard environment would be needed for industrial use.

The basic reasoning method of HAZOPEX combines the use of experience about the causes of deviations and the systematic analysis of process structure. The search starts from a deviation in a tank or in a line, and proceeds to connecting parts of the process following the process structure. On the first level, the system tells in a general form the potentially affecting disturbances and failures inside the studied part, and the deviations and disturbances at its linkages to neighbouring parts. These experience-based general level causes are described separately in a set of rules for each deviation type. The rules contain conditions which check whether the cause is possible in the process under study.

A cause is attached to a result deviation only if its condition is true. The conditions generally study whether there are such components at a proper location in the process which could cause the deviation in case of failure. A more detailed description of the inference method in HAZOPEX can be found in [8] and [9].

The knowledge base has been evaluated with two small case studies. The first one was an ammonia storage system analysed by a safety analysis expert with the assistance of one of the HAZOPEX developers. One important finding was that the nature of a HAZOPEX-aided analysis is quite different from typical manual analyses. The considerations go into more detail, which makes the analysis more accurate but slow. In the evaluation of the rule base for causes, it was found that HAZOPEX could identify 60% of the causes in some form and the rest had to be given by the user. This could reflect the division of the identified causes into typical ones found with routine analysis and special ones found with careful consideration of the problem's special characteristics. If a tool like HAZOPEX could carry out the routine part almost automatically, more time would be left for creative analysis work by the human analyst. A comparison to simultaneous manual analysis of the same system indicated that HAZOPEX is able to find most of the causes for deviations, the most important exception being such causes that would require deeper understanding of the behaviour of processed substances to be revealed [8].

The second case study was a reactor with multiple inputs. The results of the case study confirmed our findings about the potential role of HAZOPEX in safety analysis and plant design.

Based on the case studies and discussions, it seems possible that a tool like HAZOPEX could increase and predate the use of hazard and operability studies as a design tool. The next phase of development would be to transfer the current prototype version of HAZOPEX to an engineering workstation environment, and to create a better interface and more extensive knowledge base which would meet production use requirements. The results and experiences from the development are currently used in the international STARS project described later in this paper.

3. Systems modelling and analysis

The most popular method for performing detailed level safety and reliability analysis is fault tree analysis. It is therefore no surprise that a wide set of computer codes to support fault tree analysis have been developed. In general, the fault tree analysis involves two main tasks:

1. The *modelling task*: the construction of logic system models that describe how critical events (top events), identified in a previous hazard identification or qualitative analysis phase, are caused by combinations of elementary events like component failures, human errors or other,
2. The *logic and quantitative analysis* of the models constructed with the aim

to determine the minimal cut sets and to calculate numeric estimates and uncertainty bounds, for example, system failure frequency or accident sequence frequency.
It turns out that the bulk of the available computer tools support the second task. This is quite normal since it is this task that can be more easily described in algorithmic terms and implemented into a classic computer code.

Although many efforts have been spent already since the early 1970s in developing informatic support in the modelling stage, most of the more successful attempts were limited to provide fault tree drafting and editing capability, without giving any "intelligent" support to the process of system modelling. It is only recently, with the advent of expert system technology, that a real breakthrough has occurred in this field.

3.1 Fault/event tree logic and probabilistic analysis codes

Many codes exist for determining minimal cut sets (MCSs) and for performing various kinds of numerical calculations on these cut sets. Because there are so many FTA codes, we will refrain from giving a description of these and just give a short overview.

Basically, three main techniques exist for determining MCSs:
(1) combinatorial techniques: try out combinations of events up to a certain order;
(2) top-down or bottom-up substitution: substitution of gates by their cut sets taking into consideration the laws of Boolean algebra; and
(3) Monte Carlo simulation.

Whereas in the earliest day, FTA codes were based on combinatorial methods for MCS determination, today almost all analytical codes use top-down or bottom-up substitution combined with powerful modularization techniques. Moreover, many codes allow the use of logic or probabilistic thresholds so that only the most significant MCSs are determined. In this way, the necessary computer time and memory can both be reduced.

Monte Carlo simulation needs a lot of computer time and has the problem that one is never sure that all important MCSs are found. However, codes based on simulation can be a good solution for the analysis of very large fault trees, e.g. trees modelling an accident sequence as obtained in the fault tree-linking approach.

Most codes only deal with coherent fault trees. The analysis of non-coherent fault trees is possible by simulation methods. Some codes allow performance of an approximate analysis of non-coherent fault trees based on the assumption that the negated events in the tree have a probability close to 1. Therefore, the tree can be pruned from those events except in cases were negated events could combine with their complements and hence give rise to impossible cut sets.

The following gives a (incomplete) list of codes:

1. Monte Carlo simulation codes:
   (a) REMO [10]
   (b) MOCARE (also non-coherent trees) [11]
   (c) CRESSEX, CRESSC and CRESSCN (also non-coherent trees) [12]
2. Analytical codes:
   (a) FAUNET (top down) [13]
   (b) RISA (top-down) [14]
   (c) CM-MC (bottom-up) [15]
   (d) FTAP (top-down and bottom-up) [16]
   (e) PHAMISS (top-down and bottom-up) [17]
   (f) RELVEC [18]
   (g) NEWSALP (top-down and bottom-up) [19]
   (h) SALP-MP (bottom-up) [20]
3. Combinatorial codes:
   (a) PREQUAL [21]
   (b) PREP-KITT [22].

3.2 Fault/event tree workstations

To reduce the mechanical subtasks, such as data transcription, and to reduce the likelihood of making errors, integrated software systems for fault/event tree drafting, manipulation and analysis were developed [23–26]. Such codes allow trees to be drawn interactively on a screen and include facilities for modification of the trees, introduction of reliability parameters, passing the tree description to the analysis program, etc.

Fault/event tree workstations, as they are often called, have become quite popular. For larger projects, where more systems have to be analysed, the use of such workstations is almost a necessity. They provide a number of functions that are of great help in systems modelling:

1. They can help the analyst to choose unique event and gate labels so that mistakes in the logic structure are avoided;
2. They often provide the possibility to link with a data base containing reliability parameters for different types of components. In this way the work and the likelihood of making errors related to the data transcription process are reduced;
3. The possibilities to easily modify the structure of the tree or the data of the events make them invaluable for making sensitivity analyses;
4. The straightforward (graphic) input makes it easier to check the trees;
5. For plant level analysis, it is possible to create a library of system or subsystem trees that can be easily merged or combined in different ways in order to model different sequences. Fault/event tree workstations provide a natural engineering environment for performing system analysis. However, they fail

to support the actual modelling of the system itself and therefore do not give any "intelligent" contribution.

3.3 Automatic fault tree construction codes

Automatic fault tree construction generally involves [27]:

1. A description of the system in terms of component, their behaviour and the interactions between them. Such description is mostly limited to a topological one as given by a PID diagram, but experience has shown that functional or process-related information should also be used [28].
2. An algorithm which translates the system description in relation with some top-event (system state) into a fault tree.

Experience has pointed out that there is no simple solution for fully automatic fault tree construction. The problem of fault tree construction is not an easy one to solve and requires a deep understanding of the system to be analysed. One of the problems is that it is difficult to imagine, in general, all the information and knowledge on a system needed by a computer code to construct the fault tree for a particular application. Often in manual fault tree construction, the information collection process is an iterative one, and at least partly driven by questions arising during the system modelling itself. For automatic construction all the information and knowledge must be given in advance, and moreover must be formalized in a computer readable format.

A second problem is that analysts use quite a lot of general technical knowledge and experience (heuristics) in modelling the system, e.g. to make motivated simplifications or to go into greater detail on some critical issues. It is hard to include this in a conventional computer code. Therefore, the most successful attempts to automate fault tree construction concern interactive codes and, as explained later, expert systems.

The codes for automatic fault tree construction can be subdivided in two categories according to the approach chosen to represent the system under analysis:

1. Component based: "local" component behaviour models (i.e. describing components independent of their role in the system) are used, and the system is described by its topology only. Examples are the CAT code [29], the CAFTS code [27] and the RIKKE code [30].
2. Functional, structural based: the system is represented by its functional structures (process loops, control loops, by-pass lines, stand-by lines, etc.) without necessarily describing the detailed components behind these. A graph describing these functional structures is constructed and from this graph the fault tree is derived. Examples of this approach are the DIGRAPH approach [31], the "control loop structures" approach [32] and the method proposed in [33].

In the CAT code, the component models take the form of decision tables. The tables describe how different states for one process variable at the input, lead to other states at the output in function of the component internal state. The CAT algorithm starts with an output state at a component and then goes in the upstream direction to search for the combinations of input and internal states that lead to the starting state.

A drawback of the original CAT method is that the models are single parameter models, i.e. they describe, for example, only flow or only pressure, and hence are unable to express the influence of pressure on flow. Another problem is that the search algorithm is unidirectional: from output to input. Therefore difficulties arise for states which can be caused by both upstream and downstream causes, or for situations in which the flow may be reversed, and hence inputs become outputs and vice versa.

In the RIKKE code, the component models take the form of sets of mini fault trees that describe transfer functions between inputs and outputs considering component internal states. The models consider many process parameters (flow, pressure, etc) and various levels of deviation of these parameters (e.g. low, very low, zero). Fault trees generated by RIKKE can be overly complex since they may contain branches that are irrelevant for the system and process analysed. This is due to the fact that a RIKKE component model tries to describe exhaustively the behaviour of a component isolated from its environment. However, in an application, the component behaviour is restricted by its function in the system under analysis. Consider, for example, a heat exchanger, in a particular system, the heat exchanger will have a function (to cool, to heat, etc.) and, hence, cold and hot sides accordingly. The RIKKE model will consider all possible situations exhaustively without knowing how the component is used in the particular application. Since the fault tree construction algorithm does not foresee any interaction to enter functional information or to guide the modelling, irrelevant branches may be developed.

The DIGRAPH method consists of constructing a graph in which each node represents a process variable. The edges represent the relationships between the process variables and the gain that the variables can undergo in case of disturbances (e.g. because of component failures). The construction of the fault tree is performed by starting from a process variable deviation in a node and by backtracking its causes through the graph.

The control loop approach is similar to DIGRAPH. The system is subdivided into loops. For each of these, the components involved are identified and mini fault trees for the loops are constructed. The loop then is integrated into the system logic and assumes a role similar to the node in the DIGRAPH approach.

In both approaches, the component-driven one and the functional structure-driven one, the crucial difficulty still remains the same: to describe exactly the role a component, or set of components play, in the system. In the component-based methods, this difficulty gives rise to the generation of irrelevant or impossible branches in the tree, or to the fact that the tree is incomplete. The structure-based methods merely transfer the problem to the user who first has to construct (manually) a graph in which the functionalities are expressed in a formal way.

Some more specific drawbacks are summarized below:

(1) Most codes require a very rigid formalism for describing the system and require the description to be complete in all senses before starting the fault tree construction. Hence, the formalism must be designed in such a way that it can capture the necessary information under all circumstances. This is difficult to realize.

(2) Once the fault tree construction is started, it is impossible to interact with the construction process, e.g. to make motivated simplifications or to make it possible to use the analyst's knowledge.

(3) Component models that have to be generic cannot be very detailed, so there is a trade-off between detail and broader applicability of models. In an interactive approach, models can be adapted to the particular situation during the construction process.

(4) Some codes have inherent restrictions, e.g. considering only one process parameter during modelling or tracing causes only in one direction in the system network.

A recurrent argument against complete automation is that fully automatic codes for fault tree construction reduce one of the major benefits of fault tree analysis, namely, the deep understanding of the system behaviour that the analyst gains while modelling the system [34]. Indeed, manual system modelling involves an important process of learning through analysis about potential system malfunctions and their causes. On the other hand, by using automatic methods it is possible to guarantee completeness and correctness with respect to the underlying assumptions. Automatic models also decrease the likelihood of errors in logic and increase the comparability of different analyses.

4. Integrated knowledge–based environments

4.1 Multi-level analysis of process systems

Multi-level analysis of a process system represents a suitable strategy for minimizing the need for resources in plant description and for eliciting complementary plant-specific information for the internal steering of the analysis work. The basic idea is to start when the first preliminary descriptions of the plant are available. At this level, the analysis results remain general, mostly revealing the types of hazards and their effects on the system. The results represent both a review of the design and a more detailed specification for further design.

When more detailed information is available the analysis can be continued by using both the new, more detailed, description of the plant and the analysis results obtained at the less detailed level. The aim of this approach is to min-

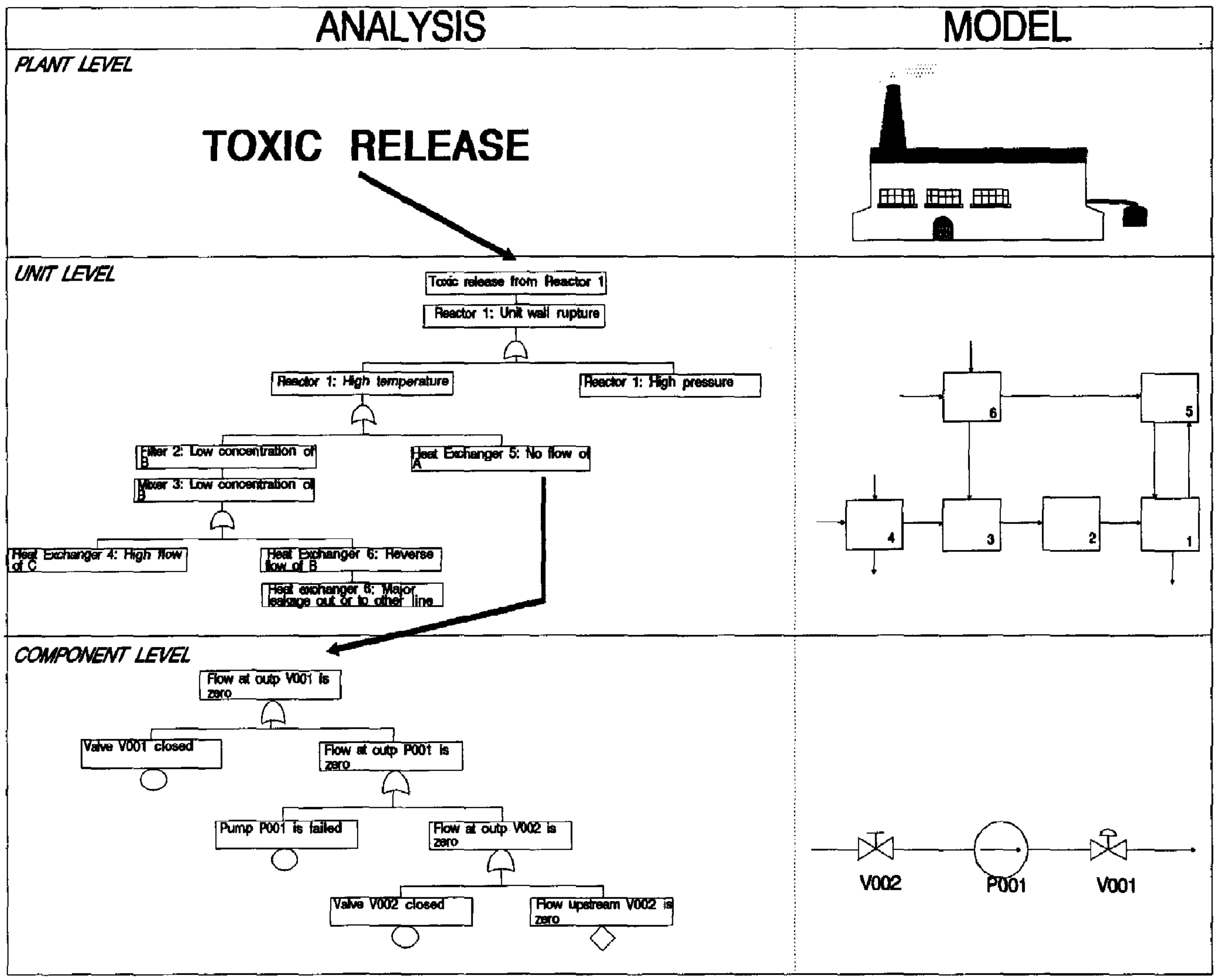
ANALYSIS
MODEL
PLANT LEVEL
TOXIC RELEASE
UNIT LEVEL
Toxic release from Reactor 1
Reactor 1: Unit wall rupture
Reactor 1: High temperature
Reactor 1: High pressure
Filter 2: Low concentration of B
Mixer 3: Low concentration of B
Heat Exchanger 5: No flow of A
Heat Exchanger 4: High flow of C
Heat Exchanger 6: Reverse flow of B
Heat exchanger 6: Major leakage out or to other line
6
5
4
3
2
1
COMPONENT LEVEL
Flow at outp V001 is zero
Valve V001 closed
Flow at outp P001 is zero
Pump P001 is failed
Flow at outp V002 is zero
Valve V002 closed
Flow upstream V002 is zero
V002
P001
V001

Fig. 1. Multi-level analysis of a plant.

imize the necessary plant description effort and to prioritize the analysis to the most significant hazards at the plant. The basic ideas of the multi-level analysis of a plant are illustrated in Fig. 1.

The results of hazard identification and analysis are then employed when selecting the units and hazards for the most detailed investigation which is made with fault trees. In this case, these units are described in the form of piping and instrumentation diagrams. The earlier studies are mainly based on more general information such as block diagrams and flow sheets.

In multi-level analysis, the less detailed levels of analysis produce such plant-specific information on hazards which can be used to automatically guide the fault tree construction process to focus on factors relevant in that specific process. In the following, the approach developed in STARS for multi-level safety analysis is described.

4.2 Overview of STARS

STARS is an integrated software package containing tools for preliminary screening of hazards, more systematical hazard identification, modelling of the hazardous event chains, quantification of the hazards, and consequence modelling. The on-going STARS development is carried out by the Joint Research Centre and Tecsa (Italy), Risø National Laboratory (Denmark) and VTT (Finland). The activities covered by STARS include:

(1) Plant representation: STARS enables the analyst to specify a top-down description of the plant under analysis in terms of the functions, substances and processes involved, the PIDs or functional diagrams of the system(s), the characteristics of the components and the description of system operation, process control and protection. The plant representation is stored into four knowledge bases (KB): one concerning the plant functional units (the plant/unit KB), one regarding the substances (the substance KB), one regarding the (chemical) processes (the reactions KB) and one concerning the plant components and systems (the component KB). The plant description process is guided by the prestored knowledge present in the four knowledge bases. This prestored knowledge includes data, models, rules and heuristics that describe the behaviour of generic process units, components and that give information on characteristics of substances and various types of industrial reactions. This prestored knowledge is used, in conjunction with the more plant specific information provided by the analyst, by the knowledge-based modules in STARS that support the hazard identification and system modelling tasks.

(2) Qualitative analysis: a plant level tool is used for the identification of hazardous events or event sequences, and the ranking of such events and event sequences in terms of the severity of their consequences. This knowledge-based qualitative analysis tool emulates the reasoning and investigation processes applied during HAZOP analysis and failure modes and effects analysis (FMEA) and uses the knowledge stored in the plant/unit, substance and reaction KBs.

(3) Event sequence modelling: a mechanism is provided to structure the identified events and event sequences into logic models. This includes construction of event trees and/or master fault trees.
(4) Systems modelling: an expert system can be used to construct logic models for the system malfunctions that appear in the event sequences or event sequence models. This includes the construction of fault trees and the generation of state graphs or transition matrices. Moreover, a tool is foreseen to help the identification of dependencies between subsystems or components. Both these tools use mainly the component KB.
(5) Model analysis: codes are integrated for the logic and/or probabilistic analysis of the constructed models for systems and event sequences. This includes event tree analysis, fault tree analysis and Markov analysis. Moreover, the analysis phase considers also the quantification of dependencies if dependent events have been identified in the previous task.
(6) Consequence analysis: modules are being integrated for a preliminary screening of consequences based on severity, for the assessment of the physical effects of the identified events or scenarios, and for the evaluation of the consequences of these effects on man and installation (vulnerability analysis).

An important objective of the STARS project is to bring plant safety and reliability analysis closer to the design and process engineer, by offering a complete environment with the "feel and touch" of more classic tools such as CAD tools. Another important asset is that all modelling is performed from the systems representation as residing in the KBs. This means that all information and assumptions underlying the analysis are documented and that a certain level of standardization is achieved in the way a plant is described and modelled.

In the following, the knowledge-based qualitative analysis and fault tree construction methodologies of STARS are described in more detail.

4.3 Hazard identification in STARS

The STARS qualitative analysis tool (QUAL) helps the user to recognize potential hazards of the process plant and event chains leading to those hazards. The definition of priorities for the hazards and possible corrective measures is also supported. The main purpose of the priorities in this case is to help the user to focus the further analysis on the most important hazards or the most critical parts of the plant.

With QUAL, three methods can be utilized and combined to identify potential hazards and event chains:
(1) Free and inspired idea processing.
(2) Interactive knowledge-supported construction of event chains based on HAZOP logic.
(3) Check lists, guide word lists and data bases of previous analyses and experiences.

When QUAL is started, a display for the potential hazards (Potential Problems

Notebook window) and event chains (Qualitative Safety Analysis window) is presented to the user. In addition, there are windows for mini-sized process graph and for frame display.

The results of idea processing can be written down in the special window in which the automatically identified potential hazards also are displayed. On request, QUAL can also inspire the idea processing by suggesting guide words from the check list. QUAL has knowledge about failures, deviations and consequences that might occur in different unit types. The knowledge resides in separate substance, reaction, plant/unit and component knowledge bases which are utilized also by the other modules of STARS. This generic knowledge, combined with the plant-specific data that the user has put in, can be used by QUAL to construct an initial set of potential hazards and event chains automatically. Depending on the amount of the plant-specific data, the event chains that QUAL constructs are general and incomplete to a certain extent. Because of that, the user should always check and modify the event chains.

The key idea in the construction of event chains is that the user selects an interesting event from the list of potential problems to be examined as the top event of the tree. Another possibility is that the user selects an interesting unit and one deviation, and starts to build the tree to both directions (causes and consequences) from that initial event. The inference is carried out in a backward chaining manner which means that the possibility of an event is evaluated by checking whether the conditions specified for the occurrence of that event hold. The conditions can represent other events which are evaluated by the backward chaining process.

Check lists, guide words and data bases of QUAL can be utilized to aid the modification of automatically constructed event chains. Check lists are (hierarchical) lists of questions that are used to confirm that all the essential aspects have been taken into account. In practice, the check lists are very application dependent. Guide word lists are lists of words which generally describe environmental, structural and functional factors that may affect to the safety. Guide word lists can be used with various applications. The user is allowed to modify the documentation of the analysis (e.g. potential problems list and event diagram), and also to modify the knowledge used to generate results.

The process description needed for automatic analysis at the highest level is a set of process units or a unit diagram. The end results of QUAL analysis are diagrams of event chains, equivalent HAZOP form representations and different summary listings. The event chains begin from macro component level deviations which can be analysed further with the STARS fault tree construction tool.

4.4 Fault tree construction in STARS

In the STARS environment, an expert system is used for fault tree construction [35]. The inference engine performs the construction of the fault tree logic for the given system and top event in two phases:

(1) First a macro fault tree is constructed which develops the top event in terms of states of the principal components in the system (i.e. the component typically present on the system PID or functional diagram), but without modelling how these states are on their hand caused by underlying basic causes. The macro fault tree construction is performed using generic production rules that are instanciated for the system topology. Formally, one could say that the generic first-order logic rule base is relocated using the specific system topology and function into a zero-order logic rule base from which the fault tree is derived. The fault tree is equivalent to a search tree of the zero-order logic rule base. This search tree is developed by a backward chaining process from the goal state (top event). The inference engine uses metarules that describe which set of rules model accurately the behaviour of a component given the context (e.g. function of the component in the system, system structure, etc.).
(2) Next, each principal component state (macro fault tree event) is further developed into its underlying causes, not only causes originating from behaviour of the principal component itself, but also causes originating from auxiliary and control systems, or any other systems functionally linked with the principal component. To perform this, the inference engine uses a rule base in which detailed causes of component states are given as a function of the component operating characteristics, its connections with control and protection circuits, etc.

5. Conclusions

The development of computer aids has reduced the time and resources needed in safety and risk analyses. Currently, several commercial programs are available for mathematical calculations and documentation of results. The research and development made in the field of knowledge engineering and experts systems has given interesting and promising results.

Expert system technology seems to be promising for computer-aided fault tree construction for the following reasons:
(1) During systems modelling, the analyst has to consider a large amount of information under various forms: the system topology, system functions, how the system is operated, the component states and their failure modes. Moreover, general technical knowledge is continuously used to make short cuts and simplifications and to avoid impossible ramifications. Expert system technology deals with such various forms of knowledge and provides ways to take heuristics into consideration.
(2) The reasoning process involved in fault tree construction is exactly the same as the one used in goal-driven rule-based expert systems: starting from a top event, the causes leading to this event are explored in a backward chaining way.

(3) Expert systems can easily accommodate new knowledge arising from new experience or better understanding of the system behaviour.
(4) Expert systems can explain their reasoning so that the analyst can become aware of why particular ramifications in the analysis were included and hence the learning aspect involved in manual hazard identification and system modelling can be maintained.

There are, however, several problems to be solved before knowledge-based safety analysis will be a common tool in process design. Typical problems are
(1) Describing the plant to the computer. This is an often tedious and time-consuming task. The need for resources can be reduced by multi-level analysis of the plant. In this case, the analysis can be started with a preliminary description, and more detailed information is given only on those subsystems that are the most critical. Another alternative is to extract the process description automatically from a CAD system. This, however, usually requires tailoring in different companies while the CAD systems vary much both in hardware and software.
(2) Validity of the knowledge base—how much irrelevant information is produced automatically and how much important information is missing. These are crucial questions in knowledge-based analyses. The amount of irrelevant information can be reduced in several ways: first of all, by using a good structure in the knowledge base; by employing functional process-specific information from the very beginning of the analysis; and by using a multi-level interactive approach. How to cover all the important information on process hazards is a more difficult task. This depends on the quality of the expert knowledge and experience behind the knowledge base. In this area, all new tools need to pass a thorough validation where independent manual and automatic analyses on the same systems are carried out.
(3) Change in design culture. The use of advanced computer tools in different phases of process design requires changes in the design culture. Here, two different logic processes need to be employed—a creative generation of design alternatives, and a critical analysis of the weakness of the generated alternatives. Designers should use new tools as a part of the quality management of their own work. This requires education and changes in working practices.

The following expected benefits motivate the development of new computer-aided safety analysis methods and the associated changes in the design culture:
(1) Safety of alternative designs is assessed earlier which leads to cheaper and better controlled changes,
(2) More detailed specifications to steer the system design are produced by the early evaluation of alternatives,
(3) A constant quality level of safety analyses is maintained by the division of safety analysis into a routine part integrated to design and a creative manual part to be carried out by a separate analysis team in the later stages of design,

(4) The results of computer-aided safety analysis are represented in a more comprehensive and flexible manner than the results of manual analysis,
(5) It is easy to update and extend the results when more experience is gained or changes are made to the process.

The experiences and examples presented in this paper show considerable interest and growing effort in using knowledge engineering, mainly expert systems, in the field of safety analysis and safety management. This trend will continue in increasing international co-operation for developing systems and tools corresponding to the need in several countries, as the final objective is the effective use of experience and expertise worldwide for managing the difficult problems of safety and reliability of process systems, both in design and operation.

References

1 R. Kakko, Computer aided consequence analysis and some future needs, J. Hazardous Mater., 26 (1991) 105.

2 D.A. Lihou, Computer-aided operability studies for loss control, In: Proc. 3rd Int. Symp. on Loss Prevention and Safety Promotion in the Process Industries, Basle, Switzerland, 1980, Swiss Society of Chemical Engineers, 1980, pp. 448–454.

3 Anon., SAnDoc—Turvallisuusanalyysien lomakkeentaytto- ja tietokantaohjelma (SAnDoc—A computer program for documentation of safety analyses), Technical Research Centre of Finland, Occupational Safety Engineering Laboratory, Tampere, 1989.

4 Anon., User guide for HAZOP-PC Version 2, Primatech Inc., Columbus, OH, 1990.

5 J.C. Parmar and F.P. Lees, The propagation of faults in process plants: hazard identification, Reliability Eng., 17 (1987) 277–302.

6 A.B. Reeves, D.A. Linkens and G.L. Wells, HAZCHECK: a computer based approach for using checklists for identifying the causes and consequences of the release of process materials, In: Proc. Design 88, Aston, 1988.

7 M. Lehtela, Computer-aided failure mode and effect analysis of electronic circuits, Microelectron. Reliab., 30, (4) (1990).

8 I. Karvonen, P. Heino and J. Suokas, Knowledge based approach to support Hazop-studies, Technical Research Centre of Finland, Research Reports 704, Espoo, 1990.

9 J. Suokas, P. Heino and I. Karvonen, Expert systems in safety management, J. Occup. Accidents, 12 (1990).

10 R. Ricchena, Reliability analysis by Monte Carlo simulation, JRC Internal Note, Joint Research Centre, Ispra, 1977.

11 Risø Report M-2109.

12 W. Guldner, H. Polke, H. Spindler and G. Zipf, Computer code Package RALLY, GRS-57, pub details 1984.

13 O. Platz and J.V. Olsen, FAUNET – User's guide, Risø Report no. 348, 1978.

14 L. Camarinopoulos and J. Yllera, Nucl. Eng. Des., 91 (1986).

15 CEA Internal report.

16 R. Willie, Fault Tree Analysis Program, Operations Research Centre, University of California at Berkeley, CA, Report No. ORC 78-14, 1978.

17 K. Terpstra, N.H. Dekker and G. van Driel, PHAMISS users manual ECN report, 1986.

18 Anon., RELVEC Manual Volume 1: Modelling, Technical Research Centre of Finland, Espoo, 1986.

19 Ansaldo SpA, Internal report.

20 M. Astolfi, C.A. Clarotti, S. Contini and R. Picchia, JRC Eurocopy Report No. 12, Joint Research Centre, Ispra, 1980.

21 KEMA, Arnhem, personal communication.

22 W. Vesely, A time dependent methodology for fault tree evaluation, Nucl. Eng. Des., 13 (1970) 337-360.

23 R. Kwik, et al., CADRIGS: Computer aided design reliability interactive graphics system, In: Proc. ANS/ENS Int. Top. Meet. on Probabilistic Risk Assessment, Port Chester, Am. Nucl. Soc., 1981.

24 J. Lynch and R. Enzinna, Computer automated fault tree construction and analysis system, In: Proc. ANS/ENS Int. Top. Meet. on Probabilistic Risk Assessment, Port Chester, Am. Nucl. Soc., 1981.

25 G. Ericsson, M. Knochenhauer and RN. Mills, Efficient fault tree handling, the ASEA-ATOM approach, In: Proc. ANS/ENS Int. Top. Meet. on Probabilistic Safety Methods and Applications, San Francisco, Am. Nucl. Soc., 1985.

26 J. Grynblat, Advanced fault tree construction, pitfalls and practical solutions, In: Proc. 8th SRE Symposium, Helsingor, Soc. Reliab. Eng. (Scandinavian chap.), 1987.

27 A. Poucet, Betrouwbaarheidsanalyse voor complexe systemen, Doct. Diss., Kathol. Univ. Leuven, 1983.

28 A. Poucet and K.E. Petersen, Software tool advanced reliability and safety analysis. In: Proc. 8th SRE Symposium, Helsingor, Soc. Reliab. Eng. (Scandinavian chap.), 1987.

29 S. Salem and G. Apostolakis, Computer oriented approach to fault tree construction, University of California at Los Angeles, CA, Eng. Report ENG-7635, Los Angeles, CA, 1976.

30 J.R. Taylor, An algorithm for fault-tree construction, IEEE Trans. Reliability, R-31 (2) (1982) 137-145.

31 S. Lapp and G. Powers, Computer aided synthesis of fault trees, IEEE Trans. Reliability, R-26 (1) (1977) 2-13.

32 A. Shafagi, P. Andow and F.P. Lees, Fault tree synthesis based on control loop structure, Trans. IChemE, 62 (1984) 101.

33 R.C. de Vries, An automated methodology for generating a fault tree, IEEE Trans. Reliability, 39 (1990).

34 R. Evans, Automatic fault tree generation, IEEE Trans. Reliability, R-27 (4) (1978) 2.

35 A. Poucet, Knowledge based tools for safety and reliability analysis. Reliability Eng. Syst. Saf., 30 (1990) 379-397.