Pergamon

Computers chem. Engng, Vol. 21, Suppl., pp. S923-S928, 1997

Printed in Great Britain

PII:S0098-1354(97)00167-1

0098-1354/97 $17.00+0.00

A Prototype for Integrating Automatic Fault Tree/Event Tree/HAZOP Analysis

D. H. Kuo, D. S. Hsu and C. T. Chang

Department of Chemical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, Republic of China

Abstract - The prototype of an integrated hazard analysis system IHAS has been developed in this study. Essentially any process can be analyzed with this software if the system topology is correctly supplied by user. Three widely accepted hazard assessment procedures, i.e. FTA, ETA and HAZOP, can be performed automatically. From the results obtained in practical applications, one can see that the quality of hazard analysis can be improved if IHAS is used as an aid to the human experts.

INTRODUCTION

Hazard analysis is an important step in designing or revamping any chemical plant. Fault tree analysis (FTA), event tree analysis (ETA) and hazard and operability study (HAZOP) are three of the most effective and widely recognized methods used for such a purpose in industry. Since, in general, these techniques are implemented manually by experts in brainstorming sessions, the need for manpower and time is often overwhelming. Thus, there are real incentives to automate these safety analysis procedures.

In the past two decades, the research on automatic fault tree analysis has advanced significantly. Among the various methods proposed in the literature, digraph is certainly the most popular one used for qualitatively modelling the fault propagation behaviors, e.g. Lapp and Powers (1977) and Chang and Hwang (1992). After additional studies in recent years (Hwang, 1993; Chang and Hwang, 1994; Chang *et al.*, 1994, Chang *et al.*, 1996), the digraph based approach has been not only developed into a practical tool for building fault trees but also extended to solve problems in event-tree synthesis.

On the other hand, research projects that aimed to automate HAZOP have begun to emerge (Venkatasubramanian and Vaidhyanathan, 1994). In general, the analysis performed in HAZOP format tends to be less rigorous and comprehensive than that in fault trees and event trees. Since part of the results generated with the latter two approaches can be adopted as the conclusions of HAZOP, our intention in this work is to integrate FTA, ETA and HAZOP into one software for use as an aid in real applications. A brief description of its prototype is presented in the sequel.

THE SYSTEM FRAMEWORK

In order to perform FTA, ETA and HAZOP for any given process, a generic saftware should at least be able to perform several basic tasks, i.e. (1) transformation of P&ID into machine processable input codes; (2) system digraph synthesis and loop identification; (3) fault tree and event tree analysis; (4) generation of HAZOP reports. The prototype of such an integrated hazard analysis system has been developed in our study. This system, referred to as IHAS, consists of 14 tool programs, a component digraph data base and a remedial action knowledge base. The software itself is written in Borland C++. The main window, which appears after clicking the system icon, is presented in Fig. 1. Task 1 described above can be done by chosing the option "Equipment." Task 2 is accomplished in "Digraph" and task 3 in "FTA/ETA." Finally, the task of generating a HAZOP report can be accomplished in "HAZOP."

DIGRAPH ANALYSIS

Since all algorithms adopted in IHAS are digraph based, it is necessary to first construct a component data base which contains a collection of small digraphs. Each of these digraphs is represented with a node list and a gain list. In the former list, the inputs to every output are specified. The gains between every pair of input and output are stored in the latter. If there is a need to incorporate process specific informations, the user can make use of one of the tool programs, i.e. *component data interpreter*, to modify the default digraphs or even add new ones into the data base.

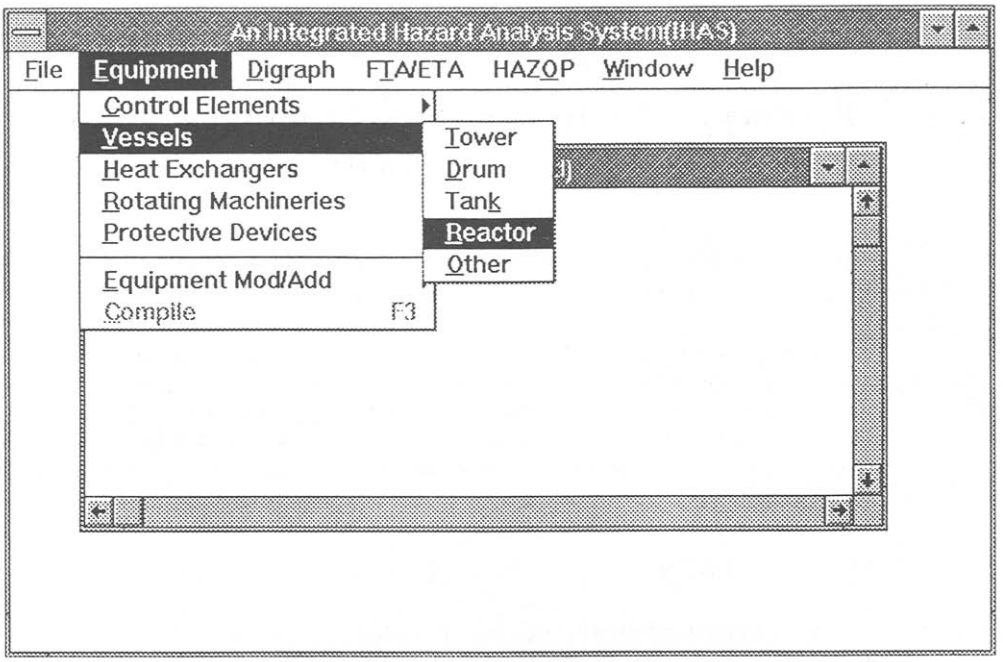

Fig. 1. The main window of IHAS and its menu concerning "Equipment."

After modifying the existing component digraphs and adding new ones to the component data base, a system digraph can be constructed according to the topology of piping and instrumentation diagram (P&ID). The equipments and their interconnections in P&ID can be specified interactively using a tool program *process reader*. Another tool program *digraph synthesizer* is then utilized to build the system model automaically. In essence, three tasks are performed in *digraph synthesizer*, i.e. (1) retrieving the component digraphs from data base, (2) relabeling them according to user specifications obtained with the *process reader*, and (3) combining the node lists and gain lists of all components and then removing repeated elements from the resulting lists. The aggregated node list and gain list obtained with the *digraph synthesizer* should be a complete representation of the system model.

As mentioned before, the system digraph is obtained by connecting component digraphs corresponding to all units in the system. Due to interaction between units, complex "loops" are often formed in these system models. From a purely structural viewpoint, two types of loops are important for implementing the digraph based safety assessment techniques, i.e. feedforward loop (FFLP - two or more paths from one node to another in a digraph) and feedback loop (FBLP - a path through the nodes in a digraph which starts and terminates at the same node). These loops can be further classified according to their functions, i.e. control loops, protection loops and process loops. As a result, the task of identifying and classifying all embedded loops becomes an indispensable step of any credible digraph based hazard analysis. An algorithm has already been developed in a previous study (Chang *et al.*, 1996) to automatically identify all loops in any digraph. The tool program *loop searcher* in IHAS was written accordingly.

FAULT TREES AND EVENT TREES

Two important steps in fault tree analysis can be carried out with IHAS, i.e. fault tree synthesis and cut set identification. The digraph based fault-tree construction algorithms were first developed by Lapp and Powers (1977) and later improved by Chang and Hwang (1992). On the other hand, the minimum cut sets can be easily identified with one of the existing standard procedures. These algorithms have been successfully implemented in IHAS.

For event tree analysis, there are also two critical tasks that can be performed automatically, i.e. event tree synthesis and accident sequence enumeration. Similar algorithms have already been developed for these purposes in our previous studies (Hwang, 1993; Kuo, 1995). They have also been incorporated in IHAS.

GENERATION OF HAZOP REPORTS

From the previous discussions, it is clear that the tool programs developed in this study are capable of generating a large portion of the informations needed in HAZOP analysis. The last function of IHAS is therefore to coordinate the execution sequence of these tools and tailor the results into the specific format of HAZOP report. Let us use a section of the olefin dimerization process presented in Fig. 2 as an example for illustrating the report generation procedure. This well known industrial process is the subject of many previous HAZOP case studies (Lawley, 1974; Venkatasubramanian and Vaidhyanathan, 1994). Consequently, the report generated by IHAS can be compared with availble results produced by a team of human experts.

Notice first that the equipments and pipelines in in Fig. 2 have already been numbered. On the basis of these numeric labels, the component models and their interconnections can be specified interactively using *process reader*. Next, the *digraph synthesizer* can be used to build the system digraph and the *loop searcher* can then be adopted to identify and classify all the embedded feedforward and feedback loops. Having executed the *loop searcher*, the following steps are taken to generate the report:

1. Select a deviation. Basically, the deviations associated with every intermediate node, i.e. a node which is neither primal nor terminal, should be considered. The allowable deviation values can be found in the gain list. For example, $p3$ (the pressure of line no. 3) is an intermediate node and its allowable deviations are $+1$ (high), $+10$ (too high), -1 (low) and -10 (too low). Thus the event "$p3(+10)$" should be selected as one of the deviations.

2. Determine the guide word. Unlike the manual HAZOP analysis, a guide word is determined *after* each deviation is selected in IHAS. This is done on the basis of a look-up table in which all proper deviations associated with every guide word are stored. For example, an increase in pressure, i.e. $p3(+10)$, is catagorized as a possible deviation of the guide word MORE OF in IHAS.

3. Generate the contents in "deviation" column. An algorithm has been developed to translate the event symbol into colloquial descriptions. The symbol "$p3(+10)$" is interpreted as "[line no. 3]: pressure is too high."

4. Construct a fault tree using the selected deviation as the top event. The tool program *fault tree builder* can be used for such a task.

5. Determine the minimum cut sets of the above fault tree. The tool program *cause finder* can be used. The minimum cut sets of the fault tree with top event "$p3(+10)$" should be: $\{p1(+10)\}$, $\{bvfcdn100(+1)\}$, $\{rpm100(+10)\}$ and $\{bfrn115(+1)\}$.

6. Generate the contents in "causes" column. The same algorithm used in step 3 can be used to translate the event symbols into colloquial descriptions. As an example, the event

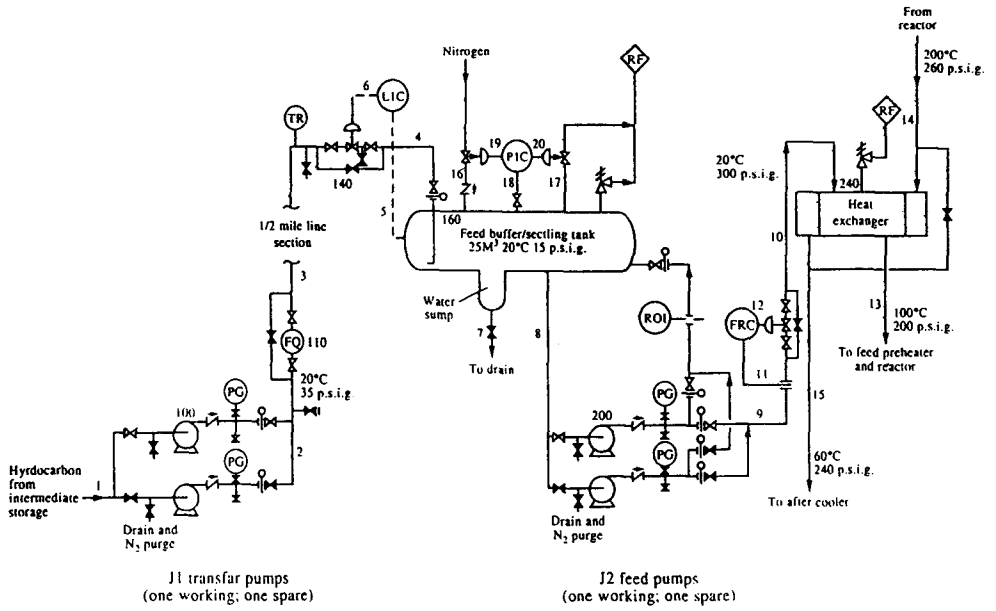

Fig. 2. The process flow diagram of the feed section of an olefine dimerization plant.

"$p1(+10)$" can be interpreted as "[line no. 1]: pressure is too high."

7. Construct the event tree corresponding to a minimum cut set. In general, there is only one basic event in the minimum cut set that is represented by a primal node in digraph. This event is used as the initial event for building an event tree. *Event-tree builder* is the tool for this purpose.

8. Enumerate the accident sequences associated with the above event tree. A *scenario enumerator* can be used to find all possible accident sequences after the initial event occurs. The accident sequences associated with "$p1(+10)$" are:

(i) { $p1(+10)$ | $p10(+1)$},
(ii) { $p1(+10)$ $cts160$ | $p10(+1)$},
(iii) { $p1(+10)$ $ccs180$ | $p10(+1)$},
(iv) { $p1(+10)$ $crp210$ | $p10(+1)$},
(v) { $p1(+10)$ $crs210$ | $p10(+1)$},
(vi) { $p1(+10)$ $cts160$ $crp210$ | $rpbn150(+10)$ $p10(+10)$ },
(vii) { $p1(+10)$ $cts160$ $crs210$ | $rpbn150(+10)$ $p10(+10)$ },
(viii) { $p1(+10)$ $ccs180$ $crp210$ | $rpbn150(+10)$ $p10(+10)$ }, and
(ix) { $p1(+10)$ $ccs180$ $crs210$ | $rpbn150(+10)$ $p10(+10)$ }.

9. Generate the contents in "safeguards" and "consequences" column. The same algorithm used in step 3 can be used to translate the accident sequences into colloquial descriptions. For example, sequence (vi) can be interpreted as "[line no. 1]: pressure is too high AND [pressure sensor no. 160]: stucks AND [relief valve no. 210]: fails RESULT IN [settling tank no. 150]: ruptures AND [line no. 10]: pressure is too high." The "consequences" column in the HAZOP report should be filled with contents after RESULT IN in this translation. The contents in "safeguards" column, however, are related to only a subset of the events stated before RESULT IN. Specifically, the basic events in the cut set, i.e. those described in the CAUSES cloumn, must be excluded. Thus, "[pressure sensor no. 160]" and "[relief valve no. 210]" should be two possible safeguards preventing the initial event "$p1(+10)$" develop into the consequence "$rpbn150(+10)$."

10. Produce the list of remedial actions. Having obtained the causes, i.e. cut sets, and the corresponding consequences, i.e. accident sequences, of a deviation, the next task in HAZOP is to work out a proposal concerning the remedial actions required to lower the frequency of the causes or reduce the severity of the consequences. In the former case, the suggestions may be changes in design, operation, maintenence procedure or even management policy of the plant. In the latter case, improvements in the protective system and/or the emergency response program are usually considered. A simple rule based expert system is used in IHAS for generating the list of remedial actions automatically. These conclusions are obtained *directly* from a set of premisses, i.e. the causes, consequences or safeguards. In other words, the procedure involves only one-step inference and the remedial action knowledge base works simply like a look-up table. Each event in the accident sequence is treated as a possible premiss of rule. An action or actions can be found if a rule in the knowledge base is applicable. For example, the remedial measures for the consequence "$rpbn150(+10)$" are: (a) check the sizing of relief valve and (b) install a pressure indicator and alarm system.

11. Repeat step 7 to step 10 for all the minimum cut sets determined in step 6.

12. Repeat step 1 to 10 until all the deviations are exhausted.

A CASE STUDY

To assess the practical value of IHAS, the computer generated results must be compared with those produced manually by experts. As mentioned before, the olefin dimerization plant in Fig. 2 is the subject of many HAZOP studies and thus a realistic version of HAZOP report is available in the literature. Consequently, this report has been used as a reference to evaluate the performance of IHAS in this study. Due to space limitation, only a sample of the contents taken from the computer generated reports is presented in this paper.

Let us consider the results concerning "no flow" in line 4 here. Its causes were found in a traditional HAZOP meeting (Lawley, 1974) to be: (i) no hydrocarbon available at intermediate storage, (ii) centrifugal pump no. 100 fails, (iii) line blockage, isolation valve closed in error, or LCV fails shut, and (iv) line fracture. On the other hand, the IHAS generated results are:

- centrifugal pump no. 100: (a) centrifugal pump fails; (b) upstream tank empty; (c) suction line blocked; (d) isolation valve on suction line closed in error; (e) discharge

line blocked; (f) isolation valve on discharge line closed in error.

- transfer line no. 115: (a) fracture.
- level sensor no. 120: (a) sensor fails high; (b) a large positive drift in zero.
- level controller no. 130 (reverse acting): (a) set point change; (b) instrument air pressure is too low AND level sensor no. 120 stucks.
- A/O control valve no. 140: (a) valve fails shut; (b) isolation valve closed in error.
- centrifugal pump no. 200: (a) centrifugal pump fails; (b) suction line blocked; (c) isolation valve on suction line closed in error; (d) discharge line blocked; (e) isolation valve on discharge line closed in error.
- level sensor no. 230: (a) sensor fails high; (b) a large positive drift in zero.
- level controller no. 240 (reverse acting): (a) set point change; (b) instrument air pressure is too low AND level sensor no. 230 stucks.
- A/O control valve no. 250: (a) valve fails shut; (b) isolation valve closed in error.

Next, the consequences of "no flow" in line 4 were identified by human experts to be: (i) loss feed to reaction section and reduced output, (ii) polymer formed in heat exchanger no. 260 under no flow conditions, (iii) centrifugal pump no. 100 overheats and (iv) hydrocarbon discharged into area adjacent to public highway. The computer generated consequences are:

- centrifugal pump no. 100: (a) centrifugal pump overheats.
- transfer line no. 115: (a) process material discharged into operation area.
- buffer tank No. 150: (a) tank preesure is too high.
- shell-and-tube heat exchanger no. 260: (a) polymer formed in the heat exchanger.
- line 13: (a) flow rate is too low.

Finally, the remedial actions suggested in Lawley (1974) were (i) ensure good communications with intermediate storage operator, (ii) install low level alarm on buffer tank no. 150, (iii) install kickback on centrifugal pump no. 100, (iv) check design of centrifugal pump no. 100 strainers and (v) institute regular patrolling and inspection of transfer line. However, a somewhat longer list was produced with IHAS:

- centrifugal pump no. 100: (a) strengthen the current pump maintenance program; (b) intitute a regular testing schedule for spare pumps; (c) ensure good communications with upstream storage operator; (d) install low level alarm on upstream storage tank; (e) install kickback; (f) check design of pump strainers; (g) provide clear written instructions about when and how to open and close the isolation valves in operator manual.
- transfer line no. 115: (a) institute regular patrolling and inspection of transfer line.
- level sensor no. 120: (a) institute a regular calibration schedule; (b) strengthen the current maintenance program of level sensors.
- level controller (reverse acting) no. 130: (a) strengthen the current maintenance program of level controllers.
- A/O control valve no. 140: (a) strengthen the current maintenance program of control valves; (b) provide clear written instructions about when and how to open and close the isolation valves in operator manual.
- buffer tank (with drain valve) no. 150: (a) check sizing of pressure relief devices and install new ones if necessary; (b) install independent pressure indicator and alarm system; (c) install low level alarm on buffer tank; (d) install high level alarm on buffer tank.
- pressure sensor no. 160: (a) institute a regular calibration schedule; (b) strengthen the current maintenance program of pressure sensors.
- pressure controller no. 180: (a) strengthen the current maintenance program of pressure controllers.
- centrifugal pump no. 220: (a) check design specifications of pump suction line;

(b) provide the correct pump operation procedure to avoid cavitation; (c) strengthen the current pump maintenance program; (d) intitute a regular testing schedule for spare pumps.

- flow sensor no. 230: (a) institute a regular calibration schedule; (b) strengthen the current maintenance program of flow sensors.
- flow controller (reverse acting) no. 240: (a) strengthen the current maintenance program of flow controllers; (b) check design of pump strainers; (c) provide clear written instructions about when and how to open and close the isolation valves in operator manual.
- A/O control valve no. 250: (a) strengthen the current maintenance program of control valves; (b) provide clear written instructions about when and how to open and close the isolation valves in operator manual.

From these results, one can see that the automatic hazard analysis is not only correct but also more comprehensive in the sense that many of the possible accidents and also remedial actions not discussed in the conventional brainstorming meetings may be identified by IHAS.

CONCLUSIONS

The prototype of an integrated hazard analysis system IHAS has been developed in this study. Three widely accepted hazard assessment techniques, i.e. FTA, ETA and HAZOP, can be performed automatically with this software. From the results we obtained in practical applications, one can conclude that the quality of hazard analysis can be improved significantly if IHAS is used as an aid to the human experts. This is due to the facts that the analyses performed by the computer system are more consistent, rigorous and comprehensive. Also, since the time spent on identifying routine accident scenarios is saved, the human experts can concentrate on the rare events that may have serious implications. Finally, it should be noted that, as a result of building such a system for a particular plant, the process specific operation experiences and knowledges are transformed into standard forms and become easily assessible. Consequently, IHAS can also be used as an effective tool for training new engineers.

NOMENCLATURE

$bvfcdn$ = pump discharge line blocked or isolation valve closed in error.
rpm = rotation speed of centrifugal pump.
$bfrn$ = local fire near transfer line.
cts = pressure transducer stucks.
ccs = pressure controller stucks.
crp = relief valve fails.
crs = vent line of relief valve choked.
$rpbn$ = tank ruptures.

References

[1] Chang, C. T. and Hwang, H. C., New development of the digraph-based techniques for fault-tree synthesis, *Ind. Eng. Chem. Res.*, **31**, (1992) 1490.

[2] Chang, C. T., Hwang, H. C., Hwang, K. S. and Hsu, D. S., The loop identification and classification algorithms for digraph-based safety analysis, *Comput. & Chem. Engng.*, (1996) in press.

[3] Chang, C. T. and Hwang, K. S., Studies on the digraph-based approach for fault-tree synthesis 1. the ratio-control systems, *Ind. Eng. Chem. Res.*, **33**, (1994) 1520.

[4] Chang, C. T., Hsu, D. S. and Hwang D. M., Studies on the digraph-based approach for fault-tree synthesis 2. the trip systems, *Ind. Eng. Chem. Res.*, **33**, (1994) 1700.

[5] Hwang, H. S., *Automation Studies of Fault Tree Analysis And Event Tree Analysis*, (1993) MS Thesis, Naitonal Cheng Kung University, Tainan, Taiwan, R.O.C.

[6] Kuo, D. H., *A Digraph-Based System for Automatic HAZOP Assessment*, (1995) MS Thesis, Naitonal Cheng Kung University, Tainan, Taiwan, R.O.C.

[7] Lapp, S. A. and Powers G. J., Computer-aided synthesis of fault trees, *IEEE Trans. Reliability*, **R-26**, (1977) 2.

[8] Lawley, H. G., Operability studies and hazard analysis, *Chem. Eng. Prog.*, **70**, (Apr. 1974) 105.

[9] Venkatasubramanian, V. and Vaidhyanathan, R., A knowledge-based framework for automating HAZOP analysis, *AIChE Journal*, **40**, (1994) 496.