To Mum and Dad
Thanks

Rules For Modelling In Computer Aided Fault Tree Synthesis

by

Andrew Hunt

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy

of the Loughborough University of Technology

April 1992

b99(9(18

Acknowledgement

Abstract

In the design of process plants safety has assumed an increasingly high profile. One of the techniques used in hazard identification is the fault tree, which involves first the synthesis of the tree and then its analysis. The construction of a fault tree, however, requires special skills and can be a time-consuming process. It is therefore attractive to develop computer aids for the synthesis stage to match those which already exist for the analysis of the tree. A computer based system for fault tree synthesis has been developed at Loughborough University. This thesis is part of a continuing programme of work associated with this facility.

The starting point for fault tree synthesis is the piping and instrumentation diagram for the plant. The analyst converts this to a configuration diagram consisting of a set of units and the connections between them. Each unit is described by a model, which may be called down from the model library or, if not available there, configured by the user. The unit models consist of sets of mini-fault trees which are then linked together under a special top event model to form the main fault tree.

This thesis is concerned primarily with the generation of the unit models and, in particular, the formulation of rules for the modelling and the creation of various model units. The methodology uses propagation equations, initial event statements and modified decision tables to model fault propagation in, and failure modes of, individual items of process plant. These are then converted to a set of mini-fault trees.

A technique has been devised to decompose a large system so that smaller parts of it can be studied separately. The flow and pressure relationships for vessels have been defined in detail and a taxonomy of vessel connections created. This has allowed a set of aids to vessel model generation to be developed, including a set of core propagation equations, model templates and a model generation routine. A taxonomy of units has been defined and a unit model library based on this has been created. Core models have been created for other common units such as heat exchangers, including a model generation routine, and chemical reaction vessels.

Fault trees have been synthesised for example systems and other improvements have been made both to the modelling and the synthesis. These include improvements to the modelling of recurring but difficult features such as dividers and headers and control and trip systems and the development of a method for creating models which produce fault trees comparable to those used in hazard assessments.

Contents

List Of Contents

3. Outline Of Methodology

PART 2

4. Rules For Problem Decomposition And Data Input

5. Specification Of Lists And Model Libraries

6. Rules For Modelling Flow And Pressure

7. Rules For The Generation Of Vessel Models

8. Rules For The Generation Of Heat Exchanger Models

9. Rules For The Generation Of Reaction Vessel Models

10. Rules For Modelling Divider-Header Combinations

PART 3

13. Worked Examples

14. Conclusions And Recommendations

APPENDICES

PART 1

Chapter 1

Introduction

1.1 Introduction To Fault Tree Synthesis

In the design of process plants handling hazardous materials safety has assumed an increasingly high profile and the design effort which must be devoted to it has grown. Formal systems are used both for the identification and the assessment of hazards. The most widely used method of hazard identification is the hazard and operability (hazop) study. A hazard so identified may then be selected for further study. One of the techniques used is the fault tree, which involves first the synthesis of the tree and then its analysis.

A fault tree is a graphical representation showing failure logic between events. A single, undesired event, the top event, is broken down into its primary causes. These, in turn, can then be broken down into their own causes and so on until the causes terminate in base events which can be interpreted easily. Between each layer a logic gate links an event to its causal events.

Such a fault tree may then be analysed qualitatively, giving combinations of base events (minimum cutsets) which may cause the top event, or quantitatively, using failure data for the base events giving a probability for the top event. For example, an explosion, for which a probability is not readily available, has primary causes of a release of an explosive mixture and an ignition source. While it may be possible to give a probability for an ignition source being available the release is not as simple. Therefore it is necessary to break this event down further. The eventual cause of a release may be the failure of a control loop component which allows low temperature embrittlement of a pipeline resulting in a leak. The failure rates of such components may be provided by the manufacturers or derived from reliability studies thus allowing the probability of an explosion to be calculated.

The formal procedures used for fault tree analysis lend themselves to an automatic approach and there are currently many computer programs available to carry out this task. This has greatly reduced the amount of time required to carry out an analysis which may evolve thousands of minimum cutsets.

The construction of a fault tree, however, requires special skills and can be a time-consuming process. There are no formal guide-lines laid down for the construction of a fault tree and much of the skill relates to the interpretation of the analyst in the design,

operating procedures and failure modes of the plant. Indeed two independent analysts may derive different results for the same plant. It is therefore attractive to develop computer aids for the synthesis stage to match those which already exist for the analysis of the tree.

1.2 The FAULTFINDER Package

A computer based system for fault tree synthesis, FAULTFINDER, has been developed at Loughborough University over several years (1 - 5). This thesis is part of a continuing programme of work associated with this facility and represents Version 3 of the package. Kelly (4) created the first version and Mullhi (5) developed Version 2.

The starting point for fault tree synthesis is the piping and instrumentation diagram for the plant. The analyst converts this to a configuration diagram consisting of a set of units and the connections between them. Each unit is described by a failure model, which may be called down from the model library or, if not available there, configured by the user. An additional model, the top event model, is used for the undesired event forming the top of the fault tree. The unit model consists of sets of mini-fault trees (hereafter referred to as minitrees) which are then linked together, during synthesis, to form the main fault tree.

The FAULTFINDER package is a suite of computer programs written in the Fortran 77 programming language based on a DEC Microvax II computer system running VMS. It is possible, with a minimum of alterations, to install the programs on any other computer system or personal computer (PC). The basic directory structure is shown in Figure 1.1.

The programs in the package are;

MODGEN3.EXE	-	Unit model generation program
EVTGEN3.EXE	-	Event model generation program
MASTER3.EXE	-	Configuration data input program
FAULT3.EXE	-	Fault tree synthesis program
PLOT3.EXE	-	Fault tree drawing program
FTAP3.COM	-	Command procedure to run the fault tree analysis program FTAP

The user developed models are held in the '[.MDAT]' directories for reference but the unit model and event model libraries are contained in the '[.M]' directories from where the MASTER3.EXE program retrieves its information. The files of interest to the analyst are contained in the '[.DRAW]' and '[.CUTSET]' directories.

```
[.FAULT]
   |
FAULT3.EXE
   |_____
   |              |              |           |
              [.RESULTS]      [.DRAW]    [.CUTSET]
               Synthesis      Fault tree   Fault tree
              output file    plotting file  analysis files
                                 |              |
                              PLOT3.EXE     FTAP3.COM
                                 |              |
                              Fault tree     Minimum
                              text file      cutsets file
   |
[.MASTER]
   |
MASTER3.EXE
   |_____
   |        |            |
          [.DATA]    [.RESULTS]
        Configuration  Configuration
          data files    output files
   |
[.EVTGEN]
   |
EVTGEN3.EXE
   |_____
   |        |            |            |             |
          [.DATA]     [.MDAT]      [.MOD]        [.M]
        Event model  Event model  Event model  Event model
          data files  text files  text minitrees  numerical
                                               minitrees
   |
[.MODGEN]
   |
MODGEN3.EXE
   |_____
   |        |            |            |             |
          [.DATA]     [.MDAT]      [.MOD]        [.M]
        Unit model   Unit model   Unit model   Unit model
          data files  text files  text minitrees  numerical
                                               minitrees
   |
[.GENDAT]
General data files
```

Figure 1.1 - FAULTFINDER Directory Structure

In order to check the correctness of the synthesised fault trees the fault tree analysis program FTAP (6) Is used. This program offers a top-down or bottom-up method of processing or a combination of the two using prime implicants called the 'Nelson' method. The non-minimal cutsets produced initially are reduced using either a modular decomposition technique or a 'dual' algorithm to increase the code's efficiency. The program has considerable flexibility and the user is able to control the processing and output.

1.3 The Work In This Thesis

This thesis is concerned primarily with the generation of the unit models and, in particular, with the formulation of rules for the modelling and creation of various modelling units. The aim of this is to produce a more user-friendly methodology and a higher degree of automation. The methodology uses propagation equations, initial event statements and modified decision tables to model fault propagation in, and failure modes of, individual items of process plant. These are then converted to a set of minitrees which are linked together to form the fault tree.

When analysing a large process plant it may be necessary to create several new models and, as is often the case, the resultant fault tree may be correspondingly large. Should this fault tree turn out to be incorrect, it may be difficult to determine where the fault lies. However, a technique has been devised to decompose a large system so that smaller parts of it can be studied separately. The advantage of this is that the fault trees are smaller and the section being studied may only contain one new model making it easier to determine where the fault is.

The model library forms an important part of the synthesis process and an ill-equiped library will reduce the acceptance of the package. For this purpose a taxonomy of units has been defined and a unit model library based on this has been created. However, the library will not contain every possible unit so it is necessary to provide the user with suitable aids to ease the creation of new models.

Experience has shown the vessel to be the most likely unit to require a new model. Therefore the methodology for the flow and pressure relationships in vessels has been revised. This was necessary because the previous methodology was unstructured and it was unclear how to approach the development of a new model.

The new methodology for vessels incorporates a taxonomy of vessel connections allowing a set of core propagation equations to be produced. From this it has been possible to develop a set of rules and further aids for vessel model generation such as model templates and a model generation routine.

Other units often requiring new models are heat exchangers and reaction vessels. The new methodology has allowed rules to be formalised and a heat exchanger model generation routine has been produced resulting in a set of core models. The problems of chemical reaction vessels are complex but rules have been developed to aid the user in the creation of such a model. Example models have been produced as an illustration.

A common approach to this type of work is to select unconventional examples from the literature and synthesise fault trees for them. As a result improvements, shown by these studies to be needed, have been made both to the modelling and the synthesis. These include, in particular, changes to the modelling of dividers and headers and their combinations, and of control and trip loops. Further work has been carried out developing a method for creating models to produce fault trees comparable to those found in hazard assessments.

1.4 The Structure Of This Thesis

This thesis is essentially divided into three parts. The first section looks at other work in the field and deals with the existing methodology, the second deals with new methodologies, developments and improvements, and the third sums up the work with examples and gives recommendations for future work. In more detail this is:

Part 1

Chapter 2   is a survey of available literature on the subjects of fault tree synthesis, qualitative modelling and model representation.

Chapter 3   gives an introduction to the methodology used by the FAULTFINDER package.

Part 2

Chapter 4   specifies the rules pertaining to the decomposition stage and configuration data input. It also details a facility to break up the configuration into manageable sections.

Chapter 5   gives an introduction to the variables and deviations used. A new fault and event library is presented and a taxonomy of unit models, top event models and secondary failure models is given.

Chapter 6   presents a new methodology for modelling flow and pressure in vessels using a taxonomy of vessel port connections to produce a set of core propagation equations.

Chapter 7   specifies the rules for modelling vessels and presents a vessel model generation routine and a set of vessel model templates which can be manipulated by the user.

Chapter 8   specifies the rules for modelling heat exchangers and presents a heat exchanger model generation routine and a set of example models.

Chapter 9   gives an outline of the rules necessary to generate reaction vessel models and two example models are presented.

Chapter 10 gives additional rules and improvements made to modelling dividers, headers, their combinations and other associated units.

Chapter 11   is an investigation of over-determined control loops and problems with combined control and trip valve systems.

Chapter 12   is a study of fault tree stability and presentation, a facility to generate risk assessment type fault trees and dimensionality aspects.

Part 3

Chapter 13   presents some worked examples studied in the course of this project.

Chapter 14   concludes the work in the project and gives recommendations for future work.

1.5 Terminology Used In Fault Trees

In the developing field of fault tree synthesis and related work many types of symbol are used (7, 8). These include AND, OR, EXCLUSIVE-OR, PRIORITY AND, NOT and INHIBIT gates and base, conditioning, diamond, external and intermediate events. In order to prevent confusion, those used within this thesis and by the FAULTFINDER package will be defined here.

The logic gate which is used to link events and their causal conditions may take one of the following forms:

AND gate	All input conditions must exist for the output event to occur
r-OUT-OF-n gate	Special AND gate implying that 'r' input conditions must exist for the output event to occur
OR gate	Only one input condition must occur for the output event to occur

However, on a fault tree it is not necessary to separately identify the two types of OR gate and the special AND gate simply has a 'r/n' symbol inside the gate symbol. These are shown in Figure 1.2.

The other symbols used refer to the event type. In this case there are three separate representations, also shown in Figure 1.2. These are:

Transmissive/intermediate event	an event requiring further development
Diamond event	a variable deviation not developed any further because it crosses the system boundary or it is developed elsewhere.
Base event	an event which represents a primary fault.
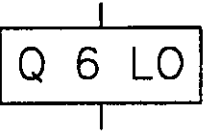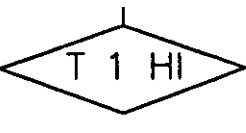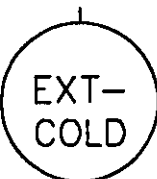
Standard Symbol	Thesis Symbol	Representation
△ (AND gate symbol)	AND	AND gate
2/3 (2-out-of-3 gate symbol)	2-OUT-OF-3	2-OUT-OF-3 gate
⌒ (OR gate symbol)	OR	OR gate
Q 6 LO	LO-PRESS	Transmissive/intermediate event
T 1 HI (diamond)	P 1 NONE	Diamond event
EXT-COLD (circle)	PART-BLK	Base event

Figure 1.2 - Fault Tree Symbols

The logic used in the FAULTFINDER package is such that the NOT gate is unnecessary and the INHIBIT and conditioning combination is superfluous because it is assumed that any variable deviation event is sufficient to cause the top event. The use of EXCLUSIVE-OR gates has been defined (9) as unnecessary from an engineering view point and results only in an added level of complication.

The gate and event symbols used by the package correspond to those approved by the British Standards Institute (8).

Minitrees represent the simplest form of a fault tree because they contain only one level of development and consequently one logic gate. An example minitree is shown in Figure 1.3.

Figure 1.3 - A Simple Minitree
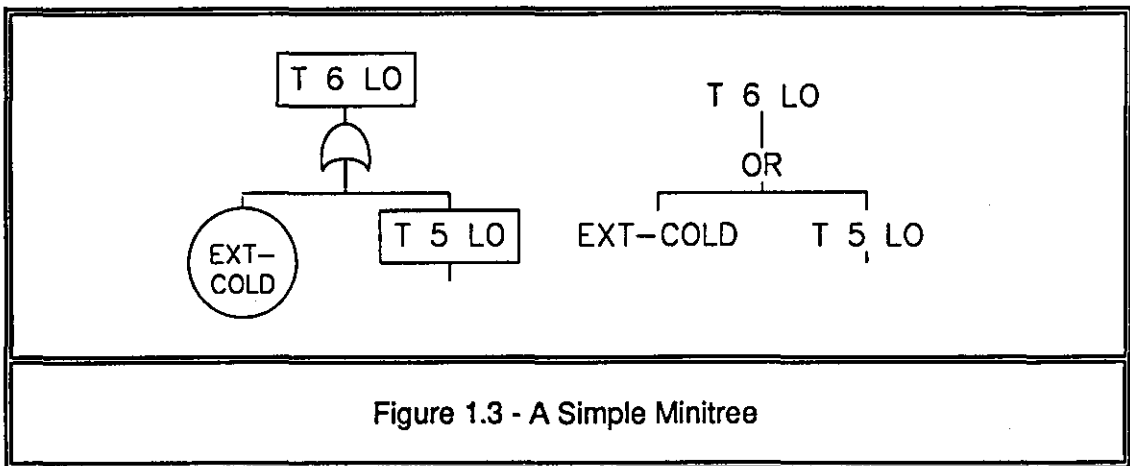
Some of the symbols used in this thesis are of non-standard type. Those likely to cause confusion are given in Figure 1.4.

Control Valve
(operating)

Trip Valve
(closed)

Centrifugal Type Pump

Positive Displacement Type Pump
eg. reciprocating, rotary

Figure 1.4 - Definition Of Some Symbols used in Thesis
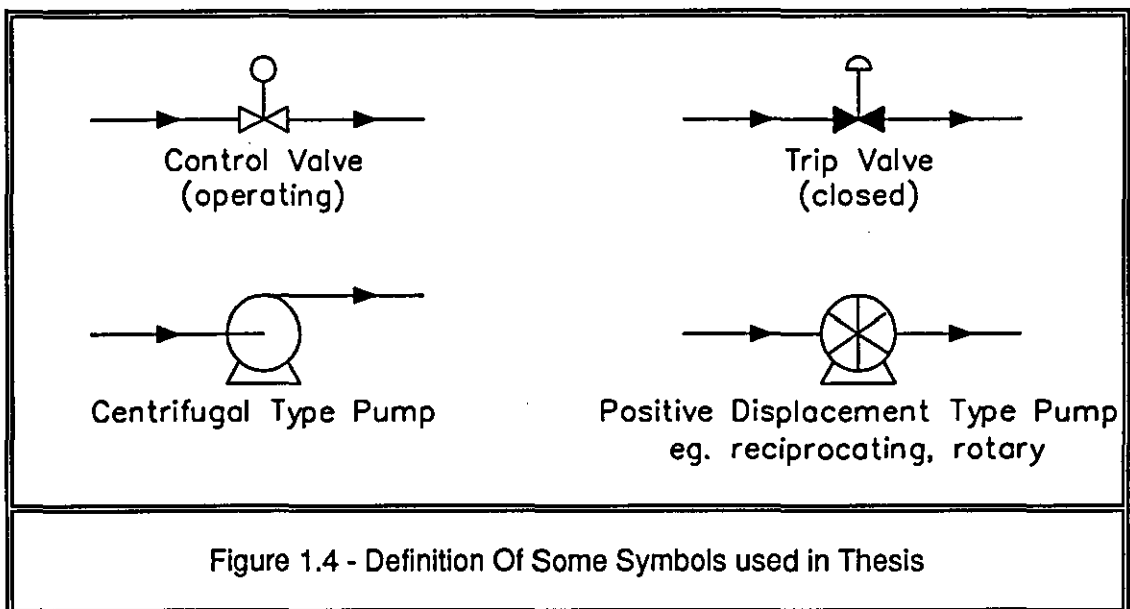
1.6 References

1. Andow P. K.; A Method For Process Alarm Analysis; PhD Thesis, Loughborough University Of Technology, 1973.

2. Martin-Solis G. A.; Fault Tree Synthesis For Real Time And Design Applications On Process Plant; PhD Thesis, Loughborough University Of Technology, 1978.

3. Shafaghi A.; Plant Modelling For Safety System Analysis; PhD Thesis, Loughborough University Of Technology, 1982.

4. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology, 1986.

5. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology, 1990.

6. Willie R. R.; Computer Aided Fault Tree Analysis; Operations Research Centre, University Of California, Berkeley, 1978.

7. Vesely W. E., Goldberg F. F., Roberts N. H., Haasl D. F.; Fault Tree Handbook; U. S. Nuclear Regulatory Commission, NUREG-0492, 1981.

8. Reliability Of Systems, Equipment And Components: Part 7. Guide To Fault Tree Analysis; British Standards Institute, BS 5760 : Part 7, 1991.

9. Lambert H. E.; Comments On The Lapp-Powers 'Computer-Aided Synthesis Of Fault Trees'; IEEE Transactions On Reliability, 28, 1979; p 6.

Chapter 2

Literature Survey

2.1 Introduction

There are currently many workers involved in the research of fault propagation in chemical process plant and in particular the subject of fault tree synthesis and analysis. As a consequence a large literature exists. Two previous workers (1, 2) in the department have carried out reviews of the subject. Therefore, only a brief introduction into the origins of this work is given. A more detailed account is given of recent publications and work related specifically to process plant. The review is restricted to the following areas:

- Fault tree work
- Qualitative modelling
- Model representation.

2.2 Fault Tree Origins

A fault tree is a sequence of events linked together by logic gates. The undesired event, or top event, is specified by the analyst and appears at the top of the tree structure. Events that may cause this event are linked to it and these in turn have their own causes linked to them. The whole structure can then be analysed to identify relationships between the top event and the factors that contribute to its occurrence.

Fault trees were first used for reliability analysis during the development of the Minuteman missile launch control system in 1961. They were confined to electronic systems in the aerospace and nuclear industry for many years due to the simple binary operation of working/not working. Later fault trees were adopted by the process industry as part of their hazard analysis work.

2.3 Analysis Of Fault Trees

The term 'fault tree analysis' is usually applied to the overall procedure of defining a system, developing a fault tree and then evaluating this to produce a set of results (3). However, it is also often used to mean just the evaluation process. In the context of this thesis fault tree analysis refers to the evaluation of the fault trees.

Fault tree analysis produces the results required to assess the reliability or hazard potential for the plant. These may be qualitative or quantitative. A qualitative result may give a series of blockages or closed valves as a cause of an emergency cooling system failure or give a particular power supply utility as an important fault location. A quantitative result gives the probability of a top event or the unreliability/unavailability of certain items of plant, thus identifying the weakest links.

The process of analysis of a fault tree, as expected, can be very time-consuming and requires a good deal of skill. However, the logic involved is well documented and from an early stage investigations were carried out into computerising the process. There exists a number of computer packages specifically designed to carry out fault tree analysis and these have been reviewed by Lee et al (3). The package listing is broken down into five sections

- qualitative evaluation for minimal cutsets
- qualitative evaluation for common-cause analysis
- qualitative evaluation for measures of importance
- quantitative evaluation for probabilities
- quantitative evaluation for measures of importance

where importance is a technical term relating to the relative contribution of a particular fault.

The intricacies of fault tree analysis are beyond the scope of the project and will not be considered any further.

2.4 Synthesis Of Fault Trees

In order to carry out fault tree analysis a suitable fault tree is required. Again, like the analysis process the construction of a fault tree is very time-consuming and requires the analyst to have a detailed understanding of the operation and failure modes of the system under study.

In the early 1970's Fussel (4) introduced a formal methodology for the construction of fault trees in electronic systems. This was made possible by the simple working/not working functionality of electrical components.

In the process industry the ability to construct a fault tree for a system is an art and takes many years of experience. This informal approach can result in different results from two independent analysts as each interprets the system in his own way.

A formal approach to fault tree construction in the process industry was not considered necessary because fault trees did not form a standard part of the design procedure for a process plant and at that time the industry had a comparatively good safety record. Other reasons put forward by Powers and Tompkins (6) can be summarised as:

- process plants were too simple and robust to warrant a formal safety technique and failure often only resulted in shutdown, followed by a restart. Compare this with the aerospace industry where failure can result in spectacular losses, and,
- hazards in process plant relate to chemical reactivity under certain conditions making it difficult to predict the behaviour and hence time-consuming to model the situation.

At that time the process industry was developing more and more complex plant as technology allowed, and for economic reasons these plants were growing larger. Then the Flixborough disaster happened and raised public concern about the process plants. This brought to light the fact that it was now necessary to carry out a hazard analysis. A widely used technique for this was fault tree synthesis and analysis.

As fault trees became an acceptable part of the design procedure for process plant ways had to be found to reduce the amount of time spent on the construction. Since then several groups of workers have been involved in the computer aided synthesis of fault trees for process plants. The recurrent authors of papers, the methodology used and the programs produced are shown in Table 2.1. The methodology is explained in later sections.

Table 2.1 - Synthesis Programs			
Authors	Date	Methodology	Program
Fussel * (4, 5)	1973	Failure Transfer Functions	DRAFT
Powers, Tompkins (6, 7)	1974	Information Flow Diagram	(no name)
Salem, Apostolakis (8 - 11)	1975	Decision Tables	CAT
Caceres, Henley (12)	1976	Block Diagrams	(no name)
Lapp, Powers (13 - 17)	1977	Digraphs	FTS
Allen (18, 19)	1980	Digraphs	(no name)
Poucet (20, 21)	1981	Modular List Processing	CAFTS
Taylor (22)	1982	Cause-Consequence Relationships	RIKKE
Kelly, Lees (23 - 27)	1986	Functional Relationships	FAULTFINDER
Kumamoto, Henley (28)	1986	Digraphs	(no name)
Napier, Palmer (29)	1988	Modularisation	(no name)
Sang Hoon Han (30)	1989	Decision Tables	AFTC
Bossche (31 - 33)	1991	Propagation Relationships and Causal Trees	FTSA

*   Although not applicable to process plant this work has been included to show its importance in the field of fault tree synthesis.

However, the efforts to develop a fault tree synthesis algorithm have encountered many problems. These include attempts to model the flow and pressure relationships and the difficulties in modelling control loops. A paper by Andow (34) explores the problem areas and reaches the following conclusions:

-   The concept of two-way flow of information is important in situations such as fluid flow.
-   The accuracy of failure models is low due to the effort spent developing algorithms.
-   The failure models do not have to be comprehensive, only a credible set of events is necessary.
-   Control loops pose particular problems to all published algorithms due to the effect of time. In these cases a different technique may be appropriate such as cause-consequence diagrams where time can be modelled.

- Fault tree methodology uses simple models but some system failures are complex which may result in incompatibilities.
- Algorithms must be carefully examined and validated.

The final conclusion has far reaching implications for all workers involved in fault tree synthesis because unless this is done the widespread use of computer codes for this work will fall into disrepute if any errors are found.

The problems of control loops were investigated by Shafaghi (35, 36) who addressed the opaqueness of the fault tree. Rather than using a unit model approach this method uses a structured format based on control loop failures and other associated protective devices. This gives an ordered fault tree which is comparatively easy to understand. The methodology described is a manual one but a combination of unit models and control loop structure is used in the work presented in this thesis.

2.5 Extent Of Automation

Many of the computer algorithms have been designed to aid the construction of fault trees. Various methodologies are used but as concluded by Kohda and Henley (37) they all require a certain amount of human interaction and skill to prevent the generation of incomplete, inconsistent and incorrect fault trees. This may occur in the modelling of the units or system, or in compiling and editing the draft fault tree.

Methodologies which use a unit based process are those of Salem (8 - 11), Poucet (20, 21), Taylor (22), Lees (23 - 27), Napier and Palmer (29) and Sang Hoon Han (30). These decompose the plant to a unit based structure and the models, if necessary, are created manually and stored in a unit library. There is a certain degree of automation in this process as the data entered by the user is converted to the required mini-fault tree format. The fault tree is then synthesised automatically by linking together the appropriate mini-fault trees using the entered configuration data.

The remaining methodologies synthesise a fault tree based on the system configuration. The digraph technique (13 - 19, 28) uses a manually drawn digraph of the system which encompasses all the control and trip loop information as derived by the analyst. Although the task is simplified by the availability of a library of digraphs for various components the user is still left to form the connections and effects of the protective devices.

The work of Bossche (31 - 33) uses the configuration topology to construct a macro-fault tree and Caceres uses path tracing through the system. Both these are automatic but the resultant tree must be edited to include the control loops.

All the methodologies require time and skill to produce a correct fault tree whether it is in creating unit models, defining the system digraph or adding in the control loop affects. However, the workers concerned are striving to make their algorithms more user friendly with more automated processes.

As can be seen, there is still much work to be carried out in the field of computer aided fault tree synthesis.

2.6 Qualitative Modelling

There are two broad categories of model which can be developed for a system, quantitative and qualitative. The former consists of a complete set of algebraic and differential equations with numerical values assigned as parameters for the model. However, it may be more convenient in certain situations, or necessary where numerical values are not known, to use qualitative modelling. This uses equations describing a qualitative relationship. For example,

$$a = f(b, -c)$$

which means 'a' is a function of 'b' and 'c', such that 'a' increases if 'b' increases or 'c' decreases, and vice-versa.

Accounts of qualitative modelling or 'naive physics' as it is sometimes known include those of De Kleer and Brown (38, 39), Kuipers (40, 41) and Forbus (42).

Qualitative models are built up from an understanding of how a system functions based on a description of its structure and its normal operation but without the need for specific numerical values. There is a need for qualitative modelling because a method based on an empirical association has serious limitations in the scope it can cover. De Kleer and Brown (39) go on to say that with a qualitative approach some loss of information cannot be avoided. This stems from the adoption of a pre-determined set of values such as -, 0 and + or low, normal and high.

When developing a qualitative model Brown and De Kleer (38) propose that the model must:

- deal with all possible faults producing unexpected behaviour and accurately model behaviour in regions not tested (this implies an adhesion to certain principles)
- portray underlying mechanisms and cause and effect relationships common to the entire class

-    be robust and learnable.

However, Kuipers (41) states that there are conflicting requirements necessary for a qualitative approach:

-    the model should express what we know about the mechanism
-    the model should not require assumptions beyond what we know
-    from the model it must be mathematically and computationally feasible to derive predictions
-    from the model it should be possible to match predictions against observations.

Despite this, much work has been carried out on the subject of qualitative modelling mainly in the field of artificial intelligence. Various theories and methodologies have been put forward including Qualitative Physics (39), Qualitative Reasoning (40, 41) and Qualitative Process Theory (42).

Forbus (42) sets out a series of reasoning tasks which must be followed when using qualitative dynamics to model a situation:

1.   Determining Activity - deduce what is happening in a situation at a particular time.
2.   Prediction - deduce what might happen in the future or futures of some situation.
3.   Postdiction - deduce how a particular state of affairs might have come about.
4.   Skeptical Analysis - determine if the description of a physical situation is consistent.
5.   Measurement Interpretation - given a particular description and observation, infer what else exists and is happening.
6.   Experiment Planning - given knowledge of observations, plan actions to yield more information.
7.   Causal Reasoning - compute a description of the behaviour.

Once a model theory has been created the model so created, according to Brown and De Kleer (38, 39), should obey five fundamental principles:

1.   Structure in Function - the structure of a device is described in terms of its components and inter-connections.
2.   No Function in Structure - the laws of the parts of the device of a particular class may not presume the functioning of the whole that are not made about the class in general.
3.   Class Wide Assumption - assumptions for a particular device must be distinguished from those generic to the entire class.
4.   Locality - a part can only act on, or be acted on by, its immediate neighbour.
5.   Causality - each event has an explicit cause.

These can be illustrated using the example shown in Figure 2.1.
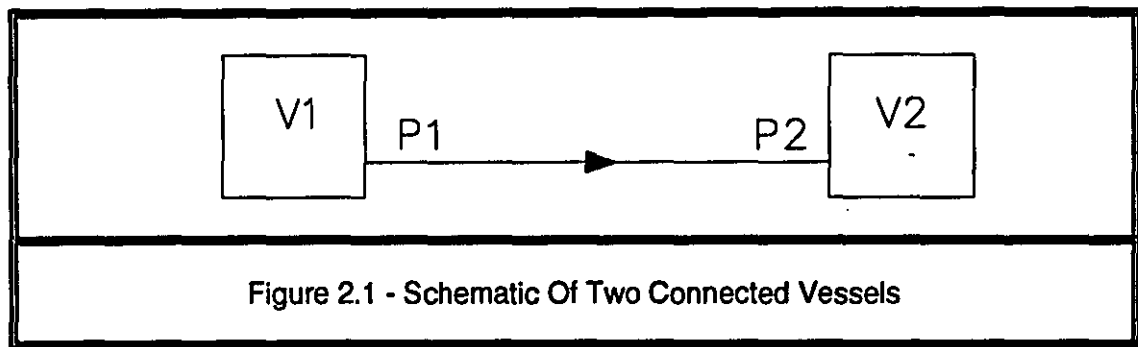
Figure 2.1 - Schematic Of Two Connected Vessels

It can be said that there will be flow from vessel V1 to vessel V2 if V2 is at a lower pressure. However, this assumes that V1 is connected to V2.

If the principles are to be enforced then the model must be localised. Therefore, there will be flow from vessel V1 if the pressure at the outlet P1 is lower than the pressure inside V1. There will be flow along the connecting pipe if P2 is lower than P1 and there will be flow into vessel V2 if the pressure at the inlet P2 is higher than the pressure inside V2.

To maintain the locality approach a connection topology is necessary to indicate communication and shared information (structure to function). The class wide assumption and no function in structure principles are applied by modelling, for example, every pipe in the same way. The causality principle provides the learnability factor as each cause/event relationship unfolds.

2.7 Automatic Qualitative Modelling

Once a methodology for determining the format of a model has been devised it is then necessary to create the models. The traditional approach (43) has been to determine what happens physically, develop a set of equations to describe this physics and then put them into a computer to solve. The problem here is that the computer is only used to solve an already built model. Catino et al (43) present a method for automatically building qualitative models in which a computer chooses the relevant physical and chemical properties from a library. The approach to modelling is based on the qualitative process theory of Forbus (42).

Starting with the plant diagram the model builder selects the desired equipment and its connectivity. A decision is made on the level of detail required and a set of assumptions about the operating conditions is applied. The computer then constructs a causal

qualitative model of the entire plant. As with all modelling programs the choice of detail and assumptions is very important and too little constraint can result in multiple models. If the right application is supplied a single model may be possible in some cases.

Only the physical and chemical properties are stored in a library and large plants are split into small sections containing two or three process units. The complete plant is made up of these sections with the output from one being the input to the next.

Based on a physical and chemical library the authors (43) give three capabilities necessary to build general models of chemical plant:

- build a basic library of phenomena that occur in chemical systems which can be used to describe the behaviour of many different types of plant
- model plug flow in pipe-type vessels
- focus on particular aspects of behaviour so that fewer solutions are produced allowing a quicker result.

These highlight the problems encountered when trying to develop a package to adequately model a chemical plant. The package must be adaptable and versatile enough to be applied to any chemical process plant. Without this the package would have only a limited use.

Modelling flow propagation in a chemical plant is a two-way process. If a pipe develops a leak then flow upstream will increase and flow downstream will decrease. A method had to be found to model this correctly but without allowing a continuous looping effect or inconsistencies as described by Martin-Solis et al (44). The solution was to use two variables, one for tracing causes upstream and one for tracing causes downstream, and a set of boundary conditions and not allowed faults. This process was adopted by Lees and co-workers (23 - 27) and a similar method using two variables by Taylor (22).

The focusing aspect depends on the use for the particular package. For a simulation process the user is interested in the behaviour of the plant under different process conditions and for a hazard analysis the user is interested in deviations from the normal operating condition and faults that may cause this.

A disadvantage of this method is that the number of different predicted behaviours increases exponentially as the number of sections increases. The growth can be limited by a focusing technique but it has to be re-applied every time the model is altered and the result is not stored in a library for use with other suitable plants. Other programs (8 - 11, 22, 23 - 27) use a component or unit base which allows the models to be stored in a library for further use.

2.8 Model Representation

This section deals with the types of model representation used by various workers in the field of fault tree synthesis of chemical process plant. A brief listing is given in Table 2.1.

2.8.1 Failure Transfer Functions

The component failure transfer function used by Fussel (4, 5) describes one mode of failure for a component. The functions are made up from the following parts:

- the output event and logic gate give the mode of failure being considered and the logic with which the failure transfer function is coupled into the fault tree

- an internal event and logic gate represent a fault requiring further logical development within the failure transfer function

- an 'input event' may be either a primary or an undeveloped fault event and represents the furthest development possible by considering the isolated component

- the 'discriminator' is a flag designating which failure transfer functions may co-exist in the final fault tree

- the 'co-ordinator' is a flag indicating which failure transfer function, in a set, is to be used for a given initial condition.

The failure transfer functions can be represented as mini-fault trees and may contain more than one level of development. They are considered independent of the system being analysed and are designed to be catalogued into a component library. The use of the flags allows the exclusion of certain failure transfer functions from the fault tree increasing its transparency. Synthesis is achieved by linking together the failure transfer functions because an input event to one is an output event of another.

This methodology marked the beginning of the development of computer programs for computer aided fault tree synthesis. The author states that although the methodology has been developed for electrical systems it can be applied to any fault tree construction work. Many existing programs still use a component based technique.

2.8.2 Information Flow Diagrams

The information flow diagram (6) for a unit is a mapping of how each component affects another. The information, for example, indicates that a particular output variable may be dependent on several input variables. A process flow sheet gives the linkage between units from which it is possible to trace the failure pathways for hazardous events.

This technique formed the basis for the Boolean equations (7) and later the digraph work of Lapp and Powers (13 - 17).

2.8.3 Decision Tables

A decision table, as used by Salem et al (8 - 11), is an extension of the binary logic of true or false in truth tables. The logic has been expanded to allow more states. The principle was developed for use in electrical systems but can easily be applied to chemical plant. The fault tree is synthesised by starting at a particular state and tracing through the appropriate decision tables for each component.

The decision table consists of an input state, an internal mode and a resultant output state. The example cited and shown in Table 2.2 is an overload fuse and the input and output states are defined as; 0= no signal, 1= normal and 2= overload. The internal modes are; 0= good, 1= failed open and 2= failed closed. The number of rows in the decision table is a product of the input states and the internal modes (in this case 3 x 3 giving 9).

Table 2.2 - Decision Table Representation			
Row	Input State	Internal Mode	Output State
1	0	0	0
2	0	1	0
3	0	2	0
4	1	0	1
5	1	1	0
6	1	2	1
7	2	0	0
8	2	1	0
9	2	2	2

The process has been further modified by using a 'don't care' logic. This involves checking the decision table and, if the output state is indifferent regardless of the internal mode or the input state, then condensing the rows as applicable. The modified version is shown in Table 2.3.

Table 2.3 - Condensed Decision Table Representation			
Row	Input State	Internal Mode	Output State
4	1	0	1
6	1	2	1
7	2	0	0
9	2	2	2
10	0	-	0
11	-	1	0

where '-' represents a 'don't care' state or mode.

This method has been illustrated in the literature using electronic systems but the authors state that it is adaptable to mechanical and hydraulic systems and, therefore, may be suitable to chemical processes. Indeed the technique has been used by Sang Hoon Han (30) for the development of another algorithm.

2.8.4 Block Diagrams

The block diagrams used by Caceres and Henley (12) consist of blocks and nodes interconnected by directed lines. The blocks represent process units and the nodes are the junctions between then. A path-finding algorithm is used to find all the routes in reverse order (ie. output node to input node) and then converting these to a set of minimal paths. Fault tree generation is then possible by using Boolean expressions for the paths.

The process is very simple and the fault tree events only represent the failure of the block or unit. It does not, for example, allow for different failures of the same unit which may have different probabilities. However, this procedure is based on fault tree generation for electrical systems which have simple success or failure of units and has formed a base for process plant fault trees.

2.8.5 Digraphs

The use of directed graphs or digraphs was initially made by Lapp and Powers (13 - 17). A digraph consists of a set of nodes, representing the process variables and faults, connected by directed edges, representing the influence of one node on another.

The degree of influence or gain, representing the possible deviation states, is given by a number in the range -10, -1, 0, 1 and 10 on the directed edge. The 0 is a normal state, the ±1 a moderate disturbance and the ±10 a large disturbance. An edge may also have a dependent relationship giving the required fault for a particular gain.

The models are represented in the form of a table giving the value of the independent node and the corresponding resultant value of the dependent node. Table 2.4 shows a link between pressure and flow rate for a pneumatic value such that as the pressure increases the flow is reduced (a gain of -1).

Table 2.4 - Digraph Representation	
Pressure	Flow Rate
-10	+10
-1	+1
0	0
+1	-1
+10	-10

The fault tree is produced by first representing the entire system as a digraph and then entering this data into the computer.

The Lapp and Powers algorithm has had a new dimension added by Shaeiwitz (17) in its ability to handle sequential operations. The only other published work on this type of situation has been by Kelly and Lees (26) for a pump change-over system.

The digraph method has proved very popular amongst workers in the safety field. Allen (18, 19) and Kumamoto and Henley (28) both favour a digraph approach rather than the component based transfer functions. They declare a better handling of complex system with loops. However, a disadvantage is the need to develop a digraph for the entire system although this itself may be made up of component digraphs. Such a system is used by

Andrews (45, 46) where each unit has its own digraph and these are linked together to form the whole system. This process requires a certain degree of skill because it is carried out manually and also requires the identification of each control loop.

The use of digraphs is not limited to fault tree construction codes and some workers have sought to by-pass the construction stage and go from system digraph to failure modes. Iri (47, 48) uses the digraphs to determine cause-effect relationships using an appropriate algorithm to produce the failure modes and Chung-Chien Chang (49) processes them to provide on-line fault diagnosis information.

Further work on the Lapp and Powers algorithm has resulted in a system for on-line fault diagnosis as described by Ulerich (50). The process starts with a system digraph encompassing all the control loop and operational procedures information. From this a causal fault tree is derived by computer which is then supplemented by real-time hazard calculation trees and fault detection trees. The cutsets of the causal tree and hazard tree are compared to determine if a hazard is imminent and the cutsets from the causal tree and detection tree are used to determine likely faults. This has provided an entrée into the work of alarm diagnosis.

2.8.6 Modular List Processing

This technique, used by Poucet (20), provides information about a component or group of components such that a macro-fault tree can be constructed. During synthesis these trees are connected together and pruned as necessary. The program makes use of the list processing provided by the PL/1 language.

The models are stored as a set of tables representing the macro-fault trees as shown in Table 2.5.

Table 2.5 - List Processing Representation							
TOP	p1		G1	p3		E1	p2
OR			AND			E	
p2			p5				
p3			p6				
p4							

The table consists of a variable and a pointer (eg. p1, p2) giving the address for the variable. The variable is made up of three parts; the first is the node name (eg. TOP - top event, G1 - gate one, E1 - event one), the second is node type (eg. OR - OR gate, AND - AND gate, E - event) and the third is a list of pointers giving the addresses of all the elements connected to that node.

2.8.7 Cause-Consequence Relationships

The models used by Taylor (22) are stored as mini-fault trees consisting of an output event or state change above an AND gate, below which there is an input event and a component condition or state. The fault tree is synthesised by connecting together these mini-fault trees.

Each of the mini-fault trees describes the effect, in terms of component state change or output event, resulting when an input or a spontaneous event occurs. An event is expressed as <variable> BECOMES <value>, eg. PRESSURE BECOMES HIGH (P → HI), and a condition as <variable> IS <value>, eg. VALVE POSITION IS LOW. The mini-fault trees are represented in table form as shown in Table 2.6.

Table 2.6 - Cause-Consequence Table Representation						
Transition Table For Output Pressure P_o						
	stem is ZERO	stem is LO	stem is DISTLO	stem is NORMAL	stem is DISTHI	stem is HI
P_i → ZERO	-	→ ZERO	→ ZERO	→ ZERO	→ ZERO	→ ZERO
P_i → LO	-	→ LO or DISTLO	→ LO or DISTLO	→ LO or DISTLO	→ LO or DISTLO	→ LO
P_i → DISTLO	-	→ DISTLO	→ DISTLO	→ DISTLO	→ DISTLO	→ DISTLO
P_i → DISTHI	-	→ DISTHI	→ DISTHI	→ DISTHI	→ DISTHI	→ DISTHI
P_i → HI	-	→ HI or DISTHI	→ HI or DISTHI	→ HI or DISTHI	→ HI or DISTHI	→ HI
P_i → VHI	-	→ VHI	→ VHI	→ VHI	→ VHI	→ VHI

Each row represents a possible input event and each column a possible condition. The elements of the table represent the component state change or output event. The example in Table 2.6 is for a control valve, the rows are for the input pressures, the columns are the valve stem positions and the elements are the output pressures. The deviations are given in an increasing order of severity as used by this methodology.

2.8.8 Functional Relationships

Functional relationships are the starting points for unit models as used by Lees and co-workers (23 - 27). The functional information contained in propagation equations, initiating event statements and modified decision tables is converted to a set of minitrees for each unit. The fault tree is synthesised by linking together the appropriate minitrees from the models.

The information takes the form

T OUT= F(T IN)	propagation equation
F EXT-COLD: T OUT LO	initiating event statement
F EXT-FIRE F NOSPRINK T T OUT HI	modified decision table.

which mean: output temperature is a function of the input temperature, low output temperature may be caused by an external cold source and high output temperature may be caused by an external fire AND no sprinkler system.

These are converted to the minitree format shown in Table 2.7.

Table 2.7 - Minitrees Derived From Functional Representation								
OR	T OUT LO	T	OR	T OUT HI	T	AND	DTROW1	T
	T IN LO	B		T IN HI	B		EXT-FIRE	F
	EXT-COLD	F		DTROW1	D		NOSPRINK	F

Further detail of the modelling can be found in Chapter 3 of this thesis.

2.8.9 Modularisation

The method proposed by Napier and Palmer (29) is a modularisation technique. The process involves decomposing the plant to a unit level and then synthesising sub-trees for these items. The final fault tree is made up by linking together the sub-trees from a storage library.

The authors place much emphasis on developing an algorithm which allows different methods of fault tree synthesis to be used in conjunction with one another. The overall plan is to be able to integrate the procedures into a design package as an aid to the design engineer.

2.8.10 Propagation Relationships And Causal Trees

The fault tree synthesis algorithm of Bossche (31 - 33) uses a two step approach to obtain the fault tree. First a causal tree is built showing the propagation paths of all the events giving rise to the top event deviation. Then all the control loops are traced in the causal tree to determine which paths may be corrected and from this the fault tree is abstracted.

The component models are made up of propagation relationships, these give the effect of one variable on another and the severity of this effect. It is similar in approach to the digraph method but uses a table format shown in Table 2.8.

Table 2.8 - Propagation Relationship Representation	
State	Relationship
Normal	$CO_2 \overset{+1}{\leftrightarrow} CO_1$
	$CO_2 \overset{+1}{\leftarrow} CO_3$
Stuck	$CO_2 \overset{0}{\leftarrow} CO_3$
Closed	$CO_2 \, (-10)$
Open	$CO_2 \, (+10)$
Reversed	$CO_2 \overset{-1}{\leftarrow} CO_3$

The representation is for a control valve. The flows in (CO_1) and out (CO_2) always have the same amplitude and so there is a direct propagation relationship between the two in both direction. The control signal (CO_3) regulates the valve throughput and can be represented by a single propagation relationship to either the input flow rate or the output flow rate. This gives the normal operation of the valve but the effects of failures can also be included. For example, a valve stuck failure represents a break in the relationship between control signal and output flow rate giving a zero relationship.

2.9 Software Packages

With the increasing availability of computer hardware and software for various stages of safety and reliability analysis it is inevitable that packages will appear on the market that will take the design engineer through the stages of the process and instrumentation diagram (P & ID) , plant layout, hazard identification and qualitative analysis.

Many authors of computer codes have striven to make their work compatible with existing programs. Fault tree construction packages make available data files which can be interpreted by many analysis programs and the analysis programs in turn can adapt to read in various outputs from construction packages. With an aim to reduce the keyboard work necessary in the input stage of construction packages, links are also being forged with the data structures used in computer aided design (CAD) packages which hold the P & ID.

There are at present some packages which encompass both the synthesis and the analysis of fault trees. One such package is the STARS (Software Tool for the Analysis of Reliability and Safety) expert system (21). This is different from the normal format of the computer packages in that rather than just automating the routine and repetitive tasks it is able to manipulate knowledge and give computer support. The aim of the system is to identify potential hazards, events and event sequences leading to hazards, construct event trees and fault trees and then analyse them.

The process proceeds in a series of phases. First the plant configuration has to be entered. This is similar to any other package except that a CAD tool is included which enables the layout to be drawn graphically on the screen. The data is taken directly from this. This is an advantage over other packages which rely on tabled data. From this topology a macro-fault tree is constructed based on the functionality of the plant. The process is interactive and the user may be prompted for additional information. The macro-fault tree is then expanded by taking into consideration component characteristics and control and instrumentation. An editor allows the users to alter the fault tree as necessary. The final phase is logical and probabilistic analysis of the tree.

2.10 Summary

This chapter has presented an outline of the work carried out in the field of fault tree construction for chemical process plant. The areas covered are fault tree analysis and synthesis, qualitative modelling and model representation.

2.11 References

1. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1986; pp 2-1-2-20.

2. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 2/1-2/27.

3. Lee W. S., Grosh D. L., Tillman F. A., Lie C. H.; Fault Trees Analysis, Methods, And Applications - A Review; IEEE Transactions On Reliability, 34, 1985; pp 194-203.

4. Fussel J. B.; A Formal Methodology For Fault Tree Construction; Nuclear Science And Engineering, 52, 1973; pp 421-432.

5. Fussel J. B.; Computer Aided Fault Tree Construction For Electrical Systems; In Barlow R. E., Fussel J. B., Singpurwalla N. D. (eds.); Reliability And Fault Tree Analysis, SIAM Conference, Philadelphia, 1975; pp 37-56.

6. Powers G. J., Tompkins F. C.; Fault Trees Synthesis For Chemical Processes; AIChE Journal, 20, 1974; pp 376-387.

7. Powers G. J., Tompkins F. C., Lapp S. A.; A Safety Simulation Language For Chemical Processes: A Procedure For Fault Trees Synthesis; In Barlow R. E., Fussel J. B., Singpurwalla N. D. (eds.); Reliability And Fault Tree Analysis, SIAM Conference, Philadelphia, 1975; pp 57-75.

8. Salem S. L., Apostolakis G. E., Okrent D.; On The Automatic Construction Of Fault Trees; Transactions Of The American Nuclear Society, 22, 1975; p 475.

9. Salem S. L., Apostolakis G. E., Okrent D.; A New Methodology For The Computer-Aided Construction Of Fault Trees; Annals Of Nuclear Energy, 4, 1977; pp 417-433.

10. Salem S. L., Wu J. S., Apostolakis G. E.; Decision Table Development And Application To The Construction Of Fault Trees; Nuclear Technology, 42, 1979; pp 51–63.

11. Salem S. L., Apostolakis G. E.; The CAT Methodology For Fault Tree Construction; In Apostolakis G. E., Garriba S., Volta G. (eds.); Synthesis And Analysis Methods For Safety And Reliability Studies, Plenum, 1980; pp 109–128.

12. Caceres S., Henley E. J.; Process Failure Analysis By Block Diagrams And Fault Trees; Industrial And Engineering Chemistry Fundamentals, 15, 1976; pp 128–134.

13. Powers G. J., Lapp S. A.; Computer Aided Fault Trees Synthesis; Chemical Engineering Principles, 1976; pp 89–93.

14. Lapp S. A., Powers G. J.; A Method For The Generation Of Fault Trees; In Gangadharan A. C., Brown S. J. (eds.); Failure Data And Failure Analysis In Power And Processing Industries; American Society Of Mechanical Engineers Conference Houston, 1977; pp 95–101.

15. Lapp S. A., Powers G. J.; Computer-Aided Synthesis Of Fault Trees; IEEE Transactions On Reliability, 26, 1977; pp 2–13.

16. Lapp S. A., Powers G. J.; Update Of Lapp-Powers Fault Trees Synthesis Algorithm; IEEE Transactions On Reliability, 28, 1979; pp 12–15.

17. Shaeiwitz J. A., Lapp S. A., Powers G. J.; Fault Tree Analysis Of Sequential Systems; Industrial And Engineering Chemistry: Process Design And Development, 16, 1977; pp 529–549.

18. Allen D. J., Rao M. S. M.; New Algorithms For The Synthesis And Analysis Of Fault Trees; Industrial And Engineering Chemistry Fundamentals, 19, 1980; pp 79–85.

19. Allen D. J.; Digraphs And Fault Trees; Industrial And Engineering Chemistry Fundamentals, 23, 1984; pp 175–180.

20. Poucet A., De Meester P.; Modular Fault Tree Synthesis - A New Method For Computer-Aided Fault Tree Construction; Reliability Engineering, 2, 1981; pp 65–76.

21. Poucet A.; STARS: Knowledge Based Tools For Safety And Reliability Analysis; Reliability Engineering And System Safety, 30, 1990, pp 379–391.

22. Taylor J. R.; An Algorithm For Fault Tree Construction; IEEE Transactions On Reliability, 31, 1982; pp 137–146.

23. Kelly B. E., Lees F. P.; The Propagation On Faults In Process Plants: 1, Modelling Of Fault Propagation; Reliability Engineering, 16, 1986; pp 3–38.

24. Kelly B. E., Lees F. P.; The Propagation On Faults In Process Plants: 2, Fault Tree Synthesis; Reliability Engineering, 16, 1986; pp 39–62.

25. Kelly B. E., Lees F. P.; The Propagation On Faults In Process Plants: 3, An Interactive Computer Based Facility; Reliability Engineering, 16, 1986; pp 63–86.

26. Kelly B. E., Lees F. P.; The Propagation On Faults In Process Plants: 4, Fault Tree Synthesis Of A Pump Change-Over Sequence; Reliability Engineering, 16, 1986; pp 87–108.

27. Mullhi J. S., Lees F. P.; The Propagation On Faults In Process Plants: 5, Fault Tree Synthesis For A Butane Vaporiser System; Reliability Engineering And System Safety, 23, 1988; pp 31–49.

28. Kumamoto H., Henley E. J.; Automated Fault Tree Synthesis By Disturbance Analysis; Industrial And Engineering Chemistry Fundamentals, 25, 1986; pp 233–239.

29. Napier D. H., Palmer P. J.; Computer Aided Fault Tree Synthesis; Pre-publication Department Of Chemical Engineering And Applied Chemistry, University Of Toronto, 1988.

30. Sang Hoon Han, Tae Woon Kim, Younge Choi; Development Of A Computer Code AFTC For Fault Tree Construction Using Decision Tables And Super Component Concept; Reliability Engineering And System Safety, 25, 1989; pp 15–31.

31. Bossche A.; Computer-Aided Fault Tree Synthesis I (System Modelling And Causal Trees); Reliability Engineering And System Safety, 32, 1991; pp 217–397.

32. Bossche A.; Computer-Aided Fault Tree Synthesis II: Fault Tree Construction; Reliability Engineering And System Safety, 33, 1991; pp 1–21.

33. Bossche A.; Computer-Aided Fault Tree Synthesis III: Real Time Fault Location; Reliability Engineering And System Safety, 33, 1991; pp 161–176.

34. Andow P. K.; Difficulties In Fault Tree Synthesis For Process Plant; IEEE Transactions On Reliability, 29, 1980; pp 2–9.

35. Shafaghi A., Andow P. K., Lees F. P.; Fault Tree Synthesis Based On Control Loop Structure; Chemical Engineering Research And Design, 62, 1984; pp 101–110.

36. Shafaghi A., Lees F. P., Andow P. K.; An Illustrative Example Of Fault Tree Synthesis Based On Control Loop Structure; Reliability Engineering, 1984; pp 193–133.

37. Kohda T., Henley E. J.; On Digraphs, Fault Trees And Cutsets; Reliability Engineering And System Safety, 20, 1988; pp 35–61.

38. Brown J. S., De Kleer J.; Towards A Theory Of Qualitative Reasoning About Mechanisms And Its Role In Trouble-shooting; In Human Detection And Diagnosis Of System Failures, NATO Symposium, 1981; pp 317–335.

39. De Kleer J., Brown J. S.; A Qualitative Physics Based On Confluences; Artificial Intelligence, 24, 1984; pp 7–83.

40. Kuipers B.; Qualitative Simulation; Artificial Intelligence, 29, 1986; pp 289–338.

41. Kuipers B.; Qualitative Reasoning: Modelling And Simulation With Incomplete Knowledge; Automatica, 25, 1989; pp 571–585.

42. Forbus K. D.; Qualitative Process Theory; Artificial Intelligence, 24, 1984; pp 7–83.

43. Catino C. A., Grantham S. D., Ungar L. H.; Automatic Generation Of Qualitative Models Of Chemical Process Units; Computers And Chemical Engineering, 15, 1991; pp 583–599.

44. Martin-Solis G. A., Andow P. K., Lees F. P.; Fault Tree Synthesis For Design And Real Time Applications; Transactions Of The IChemE, 60, 1982; pp 14–25.

45. Andrews J. D., Morgan J. M.; Application Of The Digraph Method Of Fault Tree Construction To Process plant; Reliability Engineering, 14, 1986; pp 85–106.

46. Andrews J. D., Brennan E. G.; Application Of The Digraph Method Of Fault Tree Construction To A Complex Control Configuration; Reliability Engineering And System Safety, 28, 1990; pp 357–384.

47. Iri M., Aoki K., O'Shima E., Matsuyama H.; An Algorithm For Diagnosis Of System Failures In The Chemical Process; Computers And Chemical Engineering, 3, 1979; pp 489–493.

48. Shiozaki J., Matsuyama H., O'Shima E., Iri M.; An Improved Algorithm For Diagnosis Of System Failures In The Chemical Process; Computers And Chemical Engineering, 9, 1985; pp 285–293.

49. Chung-Chien Chang, Cheng-Ching Yu; On-Line Fault Diagnosis Using The Signed Directed Graph; Industrial And Engineering Chemistry Fundamentals, 29, 1990; pp 1290–1299.

50. Ulerich N. H., Powers G. J.; On-Line Hazard Aversion And Fault Diagnosis In Chemical Processes: The Digraph And Fault Tree Method; IEEE Transactions On Reliability, 37, 1988; pp 171–177.

Chapter 3

Outline Of Methodology

3.1 Introduction

This chapter describes the basic methodology underlying the computer-based interactive facility FAULTFINDER for the representation of fault propagation in process plants and for the fault tree synthesis for such plants. The overall philosophy behind this facility has been described by Andow, Lees and Murphy (1). This description includes previous work and work carried out during this project. Accounts of detailed work carried out in earlier projects are given by Martin-Solis (2), Kelly (3, 4) and Mullhi (5).

The basic approach is:

- Start with the flow diagram of the plant and decompose this into an equivalent configuration diagram and a set of context-independent unit models.
- Each unit model is developed separately using propagation equations, initial event statements and decision tables from which a set of mini-fault trees or minitrees can be created.
- Specify the terminal event or fault condition of interest.

Using the connectivity of the units it is possible to synthesise a full fault tree for the plant from the individual minitrees. The process is systematic but templates are used for certain model combinations and consistency checks are carried out.

Where parts of the methodology are covered by later chapters only a brief description is given.

3.2 System Decomposition

The synthesis methodology employed in this project is for use in the study of plants operating in continuous conditions. However, it is possible to model batch operations to a certain degree but the methodology does not take into account effects of time on a process. The default assumption is that the process is in steady state continuous operation and the faults are deviations from this normal state.

The initial step in the modelling of fault propagation is to decompose the plant line diagram, process flow diagram, piping and instrumentation diagram or any other representation into a form which can be interpreted by the package. This diagrammatic form is called the configuration diagram and consists of units and connection lines.

3.2.1 Configuration Diagram

The configuration diagram is a block diagram which gives information on which unit models should be used and what connections exist between them. The units are the physical entities that make up the plant such as vessels and sensors and the connections are the links that exist between units such as the flow lines and signal lines. The connections have arrows on them to indicate the normal direction of information transmission.

The decomposition process is important because this determines the representation obtained on the final fault tree. The configuration diagram may consist of the basic process operations such as mixers, reactors and separators at one end of the spectrum right down to the individual components which go to make up the valves and sensors at the other. The level of decomposition is entirely at the discretion of the user.

3.2.2 Supplementary Information

The configuration diagram gives the individual units and their connectivity but many units form groups designed to carry out specific operations. Those identified are the control loops and trip systems which form an important part of the plant. Using an atomistic approach and simply 'stringing' together the minitrees involves a loss of information on the effects these units have on the plant operation and each other.

The synthesis package has a facility to handle these sub-systems but it requires supplementary information about them. The information consists of the units making up the control loop or trip system and which variables are affected in which connections. There is also some information required on the particular type of sub-system.

The systematic approach used by the synthesis package may also run into problems where information is looped. For example, a divider-header combination occurs when a process stream splits and then rejoins further downstream. This, however, results in the possibility of a loop which gives the causes of one deviation as a different deviation of the same variable at the other end of the divider-header combination. It is easy to prevent this happening in a single unit but not in a group of units. In order to prevent these inconsistencies the synthesis program treats divider-header combinations as special sub-systems. A search facility has been incorporated into the configuration data input

program MASTER which automatically identifies dividers, headers and their combinations. The user is only required to enter the flow capacities of each of the legs of the dividers and headers.

This is a brief description of the decomposition process and a more comprehensive explanation is given in Chapter 4.

3.3 The Modelling Of Fault Propagation

Generally a variable deviation in a unit will have propagated from causes in units upstream and/or downstream. For example, a high temperature in a vessel may be caused by the faulty operation of a heat exchanger further upstream. Therefore an important part of fault tree synthesis is the propagation of variable deviations or faults from where the initiating fault occurs to the point of interest.

Fault propagation is complicated by the presence of protective systems such as control loops and trip systems which may only allow the fault to pass if the system itself is faulty. This is why protective systems have to be identified in the configuration.

The propagation is further complicated by the presence of two possible directions of information transmission, particularly in process flow streams. For example a leak to the atmosphere in a high pressure pipeline may cause high flow propagation upstream and low flow propagation downstream. The pressure in the pipeline may also be affected by low pressure propagation in both directions. Other variables such as temperature and composition will only propagate upstream if accompanied by flow in the reverse direction. Two-way fault propagation is achieved by using pairs of variables, one propagates downstream in the normal direction of information transmission and the other propagates upstream against the normal direction of information transmission.

Fault propagation is achieved by using the connectivity between the units and the unit models.

3.4 The Modelling Of Units

The unit models enable faults to propagate through the unit and faults to be initiated in the unit. All unit models are made as context-independent as possible so that the models can be used in as many situations as required.

The units are broken down into ports. The ports represent entities via which information flows to or from the units. Consider the model for a simple pipe unit shown in Figure 3.1.

Figure 3.1 - Schematic Representation Of A Pipe
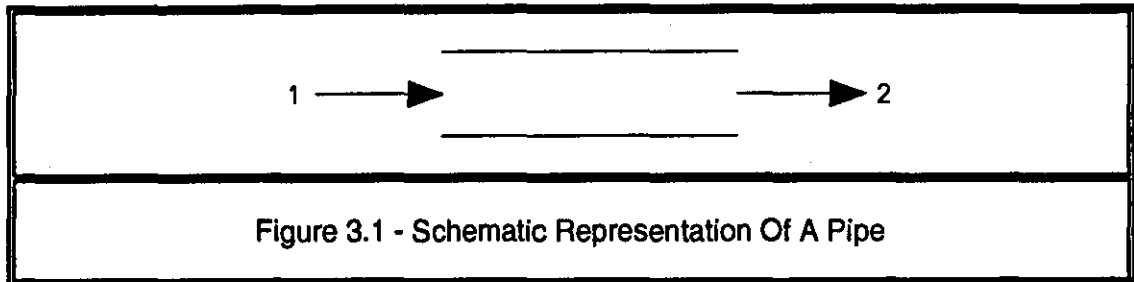
Faults propagate through the unit by entering at one port and leaving at another. For example, high temperature entering the unit at the inlet port leaves through the outlet port.

Spontaneous failures or initiating basic faults are causes of faults that propagate out of the unit. A basic fault that may cause high temperature to propagate out of the unit is an external hot source near the unit.

This information can be represented graphically in the form of a minitree as shown in Figure 3.2.

Figure 3.2 - Minitree For High Temperature Out Of A Pipe
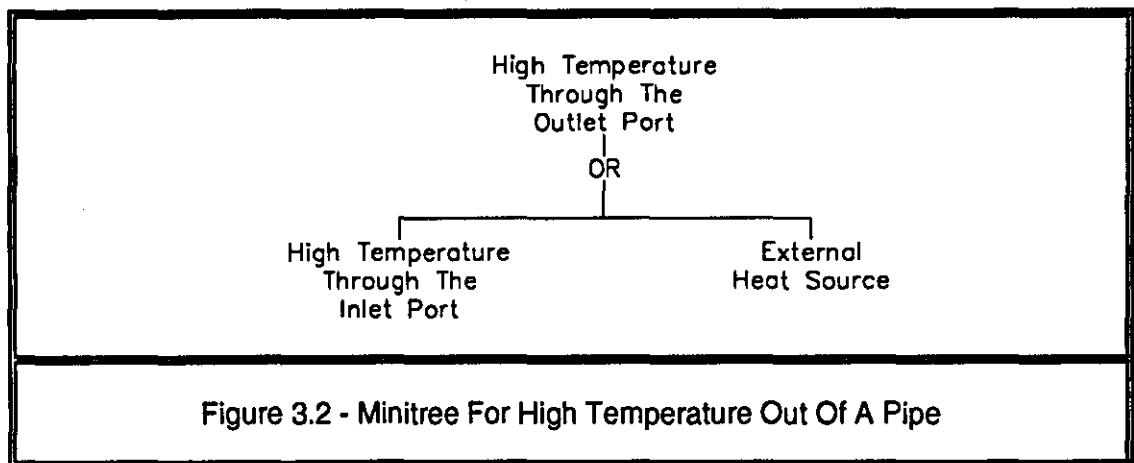
The minitree forms a simple and convenient way of representing the information. It consists of a mini-top event under which a logic gate, such as an AND or OR, is placed. The input to the logic gate consists of all the causes of the mini-top event. There is just one level of causes in any minitree.

The unit models are made up of one minitree for each variable deviation that may propagate out of the unit. A list of variables, deviations and possible combinations along with some of the initiating faults is given in Chapter 5.

3.4.1 Unit Model Format

Information about the failures and fault propagation in a unit can be defined in any of three ways. These are:

- propagation equations
- Initial event statements
- modified decision tables

The variable descriptions used have the following format

VNTYPE DEV

where

V   component letter of the variable
N   port number in the unit
TYPE   port type in the unit
DEV   abbreviation of the deviation

This is explained more fully in Chapter 5.

3.4.1.1 Propagation Equations

Propagation equations are a method of transferring variable deviations from one port to another and to pass on their effects to other variables. They are based on steady state conditions and can be derived from algebraic or differential equations or arrived at heuristically. They take the form

$a = f(b, -c...)$

This equation is interpreted as

'a' becomes high if 'b' becomes high OR 'c' becomes low, and
'a' becomes low if 'b' becomes low OR 'c' becomes high.

The propagation of most of the variable deviations can be modelled in this format.

Two-way propagation of flow is modelled using a convention involving the flow variable Q and the pressure gradient G. The advantage of this pairing over the others is that the deviations of one have a direct correspondence with the deviations of the other. This also allows a unique methodology to be employed to achieve efficient two-way fault propagation for flow. The propagation equations for flow are:

G1IN= F(Q1IN, Q2OUT)
Q2OUT= F(G1IN, G2OUT)

It should be noted that flow Q is a function of the pressure gradient G and vice-versa.

Both equations are used in the unit model to propagate flow in both directions. For example, the second equation gives the causes of high flow out of the unit (Q 2 HI) as high pressure gradient into the unit (G 1 HI) or out of the unit (G 2 HI). The causes of high pressure gradient into the unit (G 1 HI) are found in the first equation and are high flow into the unit (Q 1 HI) or out of the unit (Q 2 HI). The second occurrence of Q 2 HI is deleted. The causes of G 2 HI are found in the next downstream unit and those of Q 1 HI are found in the next upstream unit. This is illustrated in Figure 3.3.

Figure 3.3 - High Flow Propagation In A Unit
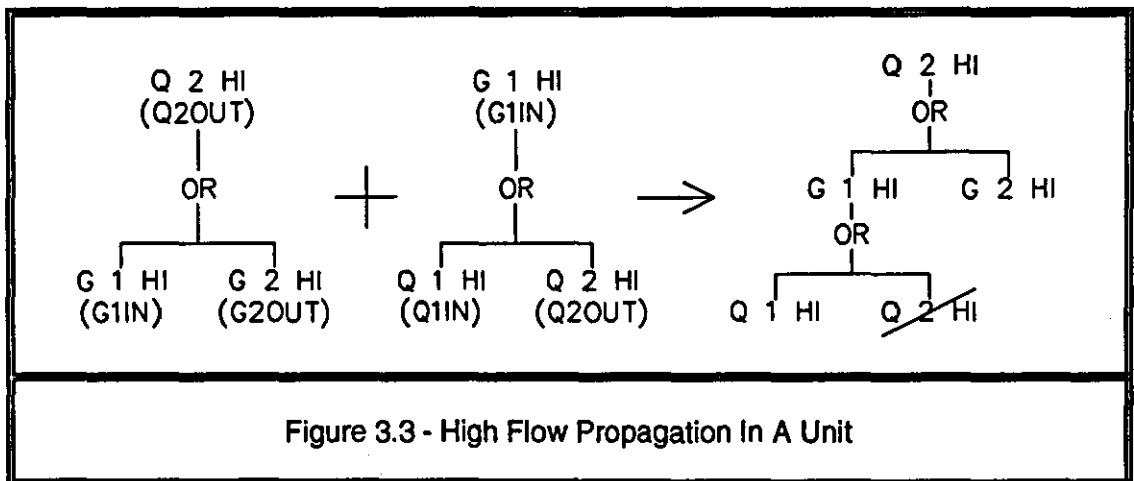
The variable names in brackets indicate which part of the propagation equation is used. It should also be noted that the port type identifier is omitted from the minitrees during processing by the model generation algorithm because it is not required.

This modelling approach is unique for flow because both G and Q are used for fault propagation in both directions. A more detailed account is given in Chapter 6.

Two-way propagation of pressure is achieved using the absolute pressure variable P and the pressure relief variable R. Only one variable is used for fault propagation in one direction because the deviations of the two variables are not matched in the same way as G and Q. Separate propagation equations are required.

    R1IN= F(R2OUT)
    P2OUT= F(P1IN)

The absolute pressure P is used to propagate pressure and relief deviations in the normal direction of flow and relief R is used to propagate pressure and relief deviations against the normal direction of flow. A more detailed account of pressure modelling is given in Chapter 6.

Two-way propagation of temperature is not as important as that for flow and pressure. It is also restricted by the need for flow against the normal direction in order for temperature deviations to propagate upstream. The variable T is used for temperature in the normal direction of flow and the variable U for temperature against the normal direction of flow. The propagation equations are:

    U1IN= F(U2OUT)
    T2OUT= F(T1IN)

Two-way propagation of composition is identical to temperature. The variable X is used for composition in the normal direction of flow and the variable Y for composition against the normal direction of flow. The propagation equations are:

    Y1IN= F(Y2OUT)
    X2OUT= F(X1IN)

The other variables such as level L and signal S do not require two-way fault propagation and are much simpler to model. They only require one propagation equation each. Some examples are:

    L3VES= F(G1IN, -Q2OUT)
    S2SIG= F(S1SIG)

There are some restrictions on the format of propagation equations. The variable on the left hand side of the equation ('a') must be one which occurs within or propagates out of the unit. The right hand side variables must be ones which propagate into the unit, except in two circumstances. The first is in flow equations which require one variable to be an outlet variable due to the way two-way fault propagation is achieved. The second is when it is permissible to have a single outlet variable provided this is the only variable on the right hand side. For example, the propagation equation for the signal from the temperature sensor unit is a function of the outlet temperature from the unit.

S3SIG= F(T2OUT)

This prevents confusion about the direction of variable deviations that propagate into the unit necessary to cause the output deviation.

3.4.1.2 Initiating Event Statements

Initiating event statements, or simply event statements, are used to model the way in which basic faults in units affect the variables that propagate out of the unit and to include the effects of variable deviations that cannot be modelled using propagation equations. They take the form

i cause : effect 1, effect 2, ...

The cause identifier 'i' gives the cause type of each cause in the event statement. These can be:

V   variable deviation
I   intermediate event
F   spontaneous failure or basic fault
O   operator action or inaction
S   state (normal or impossible)

The cause may be a fault from the basic fault and event library (Appendix B.1) or a variable deviation. The effect is always a variable deviation.

They can also be used in certain situations to include AND gates and r-OUT-OF-n (r/n) gates in models. These take the form:

i cause 1 AND i cause 2 AND ...: effect 1, effect 2, ...
i cause 1 AND r i cause 2 AND r ...: effect 1, effect 2, ...

The AND r is part of an r/n gate where 'r' is the number of causes that must exist for the event to happen. The 'n' is derived implicitly from the number of causes in the event statement.

Using the basic pipe model (Figure 3.1) which consists of a single inlet, port 1, and a single outlet, port 2, the two event statements for external temperature sources can be given as:

   F EXT-HEAT : T2OUT HI
   F EXT-COLD : T2OUT LO

which state that high temperature at the outlet may be caused by an external heat source and low outlet temperature may be caused by an external cold source.

These event statements can be combined with the propagation equation for temperature in the normal direction of flow in Section 3.4.1.1 to give the outlet temperature minitrees for the pipe model shown in Figure 3.4.

Figure 3.4 - Outlet Temperature Minitrees For A Pipe
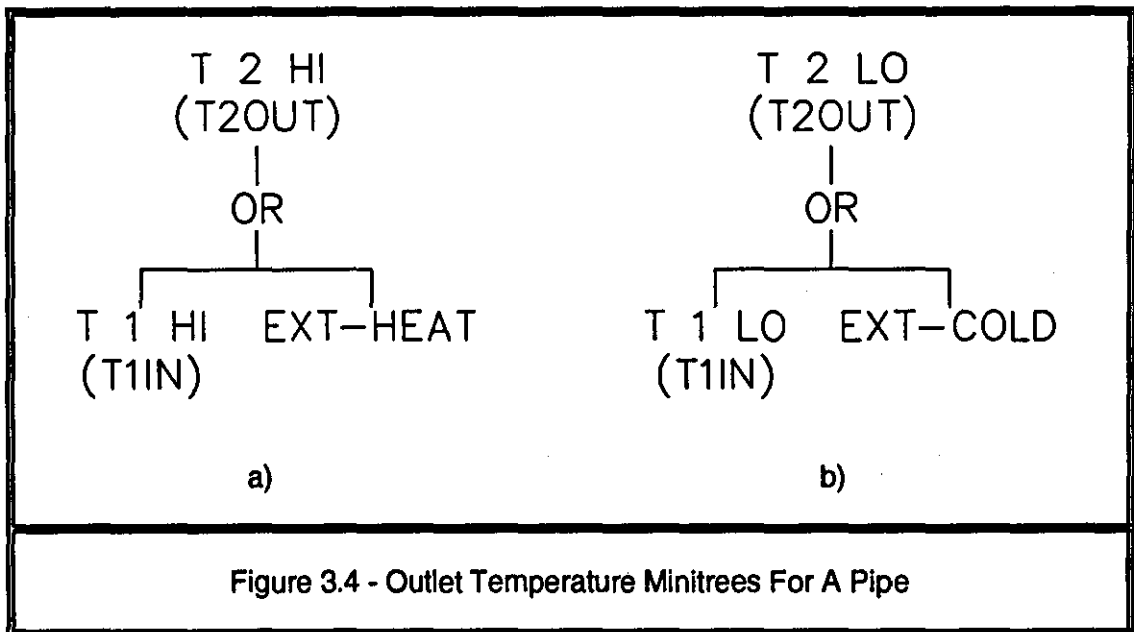
3.4.1.3 Modified Decision Tables

Modified decision tables, or simply decision tables, are used when it is necessary to introduce AND logic or to combine OR and AND logic. They take the form

   i cause 1 i cause 2 ... T effect 1, effect 2, ...

where the 'i' is the cause type identifier as used in the event statements. The 'T' is used to separate the causes and the effects. The effect of the decision table is to AND together all the elements to the left of the 'T'.

The main use of decision tables is for simple AND logic but more powerfully they can be used to combine OR and AND logic. Consider the case for high temperature propagating out of a valve which is normally closed. For this to happen there must be a high inlet temperature AND the valve must be open. For optimum efficiency while still maintaining the correct logic use must be made of decision tables. The following are used:

V T1IN HI F HV-F-OP T T2OUT HI
V T1IN HI O HV-D-OP T T2OUT HI

This states that there are two cause combinations for high outlet temperature. Either high inlet temperature AND the valve failing open OR high inlet temperature AND the valve being directed open. This produces three minitrees, one for each decision table and one which combines the effects of the two. When combined they give the structure in Figure 3.5.

Figure 3.5 - High Outlet Temperature Partial Fault Tree For A Closed Valve
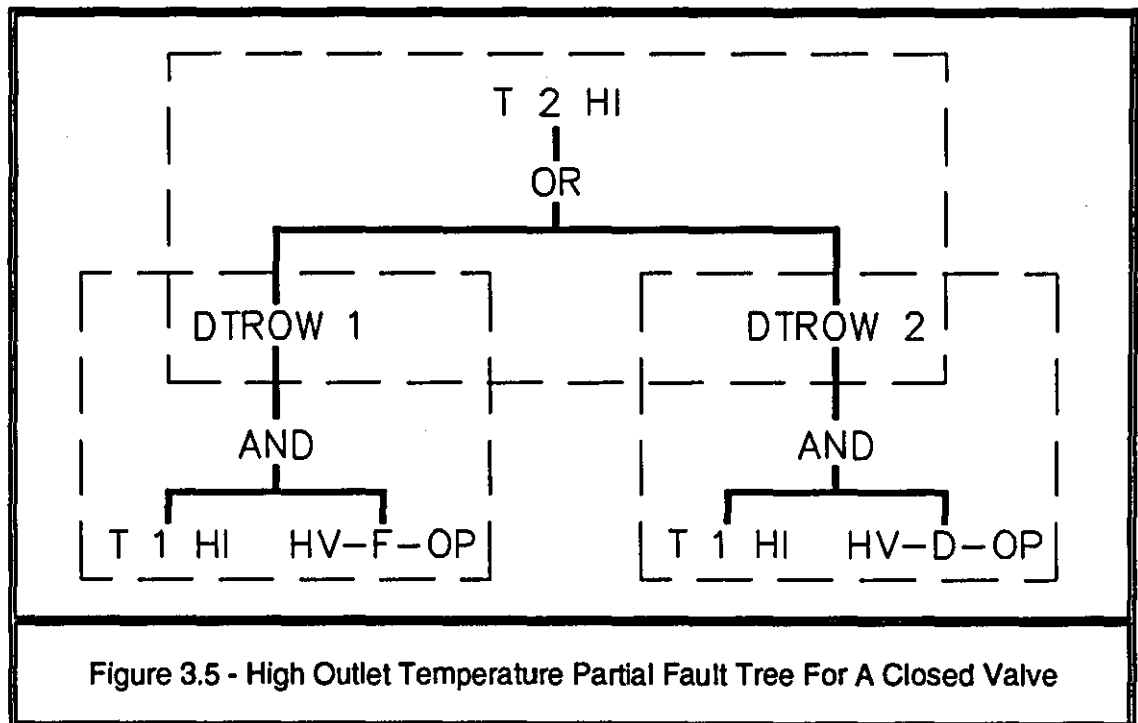
The DTROW 1 and DTROW 2 are intermediate events used to indicate from which decision table the causes are derived.

3.5 The Modelling Of Terminal Events

Terminal event, or top event, modelling deals with the termination of fault propagation. It is carried out independently of the units for three reasons

- there are many top events and to include them in every unit model is inefficient.
- it allows a top event model library to be set up.
- it allows the top event to be changed quickly and easily.

Top event models consist of a text string describing the terminal event, for example HIGHTEMP, and a set of event statements or decision tables giving the variable deviations which may cause the event. The decision tables for high temperature are:

V T2OUT HI V Q2OUT SOME T HIGHTEMP
V U1IN HI V G1IN REV T HIGHTEMP

Therefore, for the high temperature top event to occur there must be either high temperature out of the outlet port AND some flow out of the outlet port OR high temperature out of the inlet port AND flow in the reverse direction through the inlet port.

Although these decision tables are very similar to those used for high outlet temperature (section 3.4.1.3) the top events are different because they are necessary to initiate the fault trace in both directions.

A top event model is generally not specific to a particular unit but it may be specific to a class of units. In the above example, the relevant class is that of units with port 1 as an inlet port, port 2 as an outlet port and any other ports as non-flow ports.

Top event models are dealt with in Chapter 5.

3.6 Fault Tree Synthesis

Once the configuration has been specified and all the unit models and the top event model have been created the synthesis process is automatic. The procedure is to start with the top event and then synthesise the causes for each and every branch until they all terminate in a basic fault or diamond event. A diamond event is a variable deviation which crosses the system boundary and is developed no further.

The minitrees generated by the model generation algorithms are ideal for this method of fault tree synthesis. The basic principle is to link together the appropriate minitrees since the event that propagates into one unit model is simply an event which has propagated out of another unit model.

The process can be illustrated with reference to the HIGHTEMP top event in the previous section. This event has four variable deviation causes, T2OUT HI, Q2OUT SOME, U1IN HI and G1IN REV. The initial causes of these are all found in the unit where the top event occurs. Each is developed in turn. If the top event unit is a pipe type unit, then the event T2OUT HI has causes given by the minitree in Figure 3.4a. In the synthesised fault tree the port type identifier is omitted, therefore T2OUT HI becomes T 2 HI. The causal events are T 1 HI OR EXT-HEAT. The EXT-HEAT event is a basic fault and is developed no further. The event T 1 HI does not have any causes in the current model but since it is an input event it is also an output event of the next unit upstream. Therefore the event is changed to T 2 HI in the next unit. The appropriate minitree is retrieved from this unit.

This very simple approach works well but checks are required to ensure the consistency of the resultant fault tree. Several sets of consistency checks are carried out both during and after the synthesis process.

The synthesis process is also constrained by the presence of sub-systems as mentioned in the decomposition. Control loop templates are added during synthesis to model the failures of these systems and trip loop templates are added during a rationalisation stage at the end of the synthesis. The divider-header combination effects form part of the consistency checks and may also impose certain tree structures.

3.7 Consistency Checks

Consistency checks are used to remove variable deviations and basic faults from the fault tree both during and after construction. They also cover consistency in branches and between branches. There are two types of check carried out:

- series checks refer to the consistency of events within any given branch
- parallel checks refer to the consistency of events between different branches.

The sub-divisions for the events operated on are:

variable deviation - variable deviation : Boundary Conditions
variable deviation - basic fault   : Not Allowed Faults
basic fault   - basic fault   : Duplicated Events

There are four series checks and one parallel check.

3.7.1 Series Consistency Checks : Boundary Conditions

At each level of development a note is made of the plant condition. Any variable deviation event which violates this plant condition is stored in a list of boundary conditions and subsequent occurrences of the event are deleted.

For the pipe model in Figure 3.1 the fault tree developed from the two propagation equations with the addition of two event statements for high flow is given in Figure 3.6.

Figure 3.6 - Fault Tree For High Flow
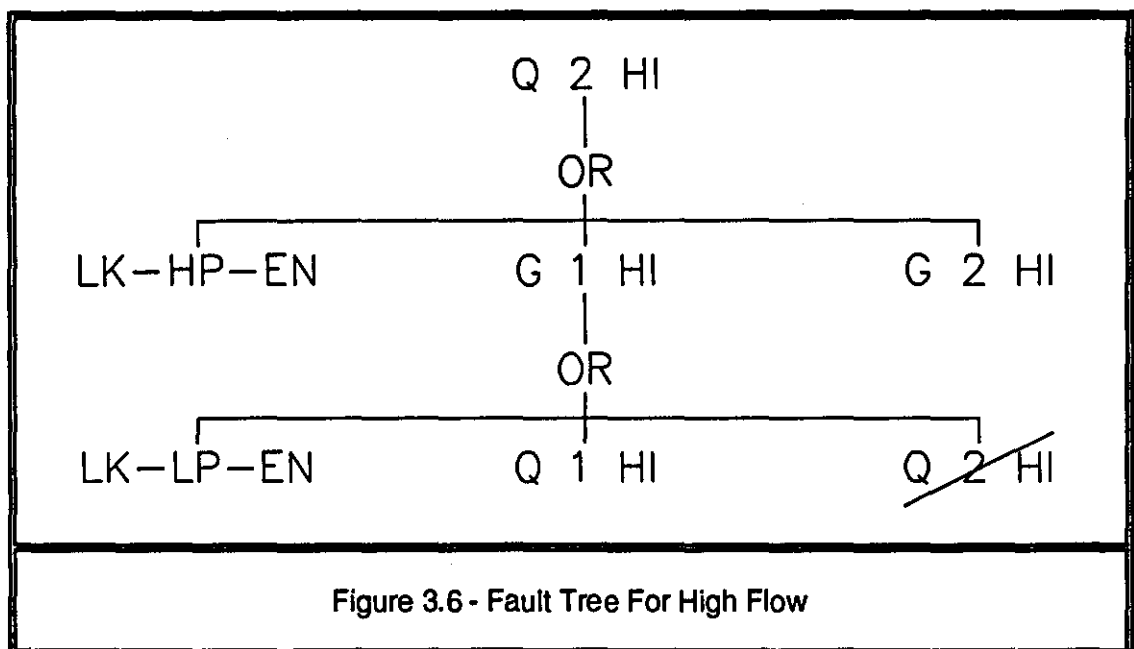
The basic principles for the boundary conditions are:

- no other deviations of the current variable can occur at the current location
- the occurrence of the same deviation for the current variable will add nothing to the fault tree.

From the boundary conditions the second occurrence of Q 2 HI in Figure 3.6 is deleted for violating the second condition.

3.7.2 Series Consistency Checks : Not Allowed Faults

Certain combinations of basic fault events are not allowed. For example, in Figure 3.7 the basic fault LK-HP-EN (leak from a high pressure environment) forms part of the minitree for Q 2 HI and the fault LK-LP-EN (leak to a low pressure environment) forms part of the minitree for G 1 HI. However, the fault LK-LP-EN is inconsistent with the fault LK-HP-EN. The consistency checks note that LK-LP-EN is added to the fault tree after LK-HP-EN and consequently LK-LP-EN is labelled a not allowed fault of Q 2 HI.
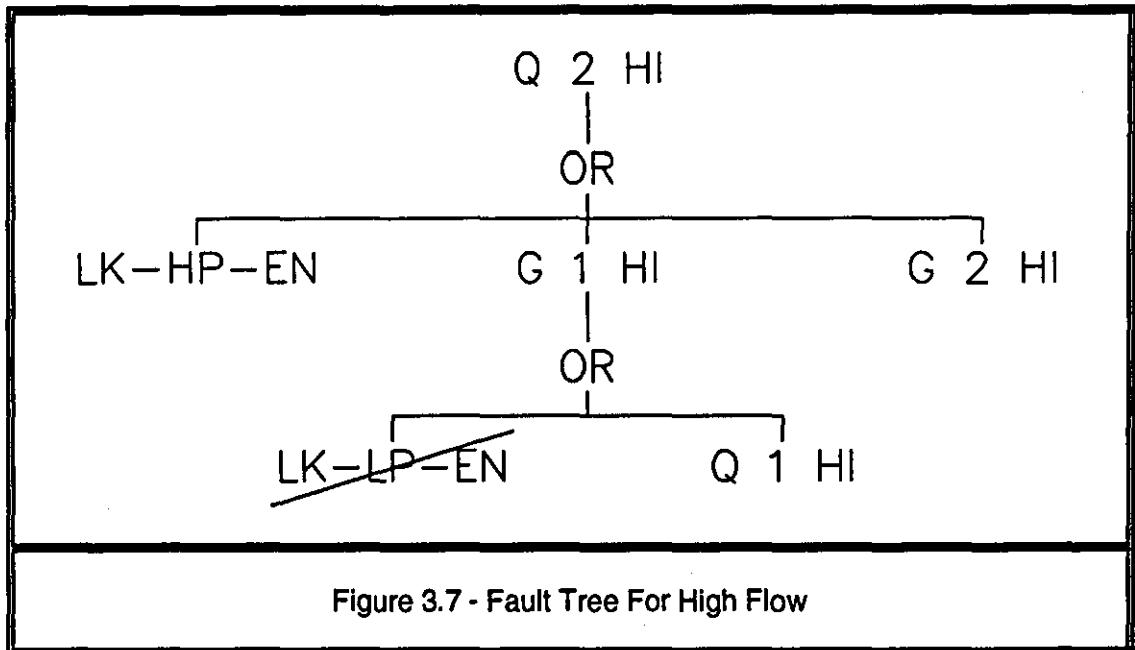
Figure 3.7 - Fault Tree For High Flow

3.7.3 Series Consistency Checks : Duplicated Events

Now consider the low flow fault tree for the pipe model shown in Figure 3.8.
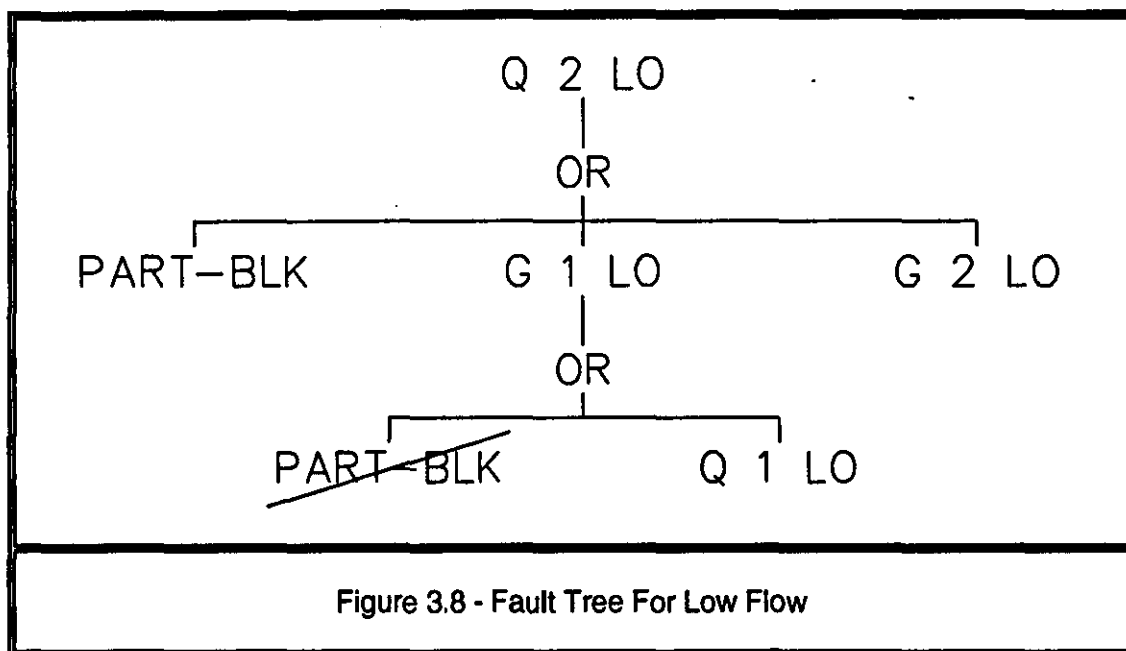
Figure 3.8 - Fault Tree For Low Flow

The second occurrence of PART-BLK is a duplicated event and adds nothing new to the fault tree so it is deleted.

3.7.4 Series Consistency Checks : Port Changes

In certain types of unit, eg. dividers and header, development of the fault tree can lead to looping around the ports and different deviations of the same variable in different branches. The port change consistency check is intended to deal with this problem. It is applied to units with more than two flow ports, for example, a divider with one inlet port and two outlet ports or a header with two inlet ports and one outlet port.

Consider a header model which has port 1 for one of the inlets and port 3 for the other. Port 2 is for the outlet.

The partial fault tree for high flow in the port 1 inlet is shown in Figure 3.9. A cause of high flow at one inlet port, G 1 HI, may be high flow at the outlet port, Q 2 HI, or low flow at the other inlet port, G 3 LO. The modelling of two-way propagation means that the causes of high outlet flow are high flow at one inlet port, G 1 HI (deleted because it occurs higher up the branch), or high flow at the other inlet port, G 3 HI. Similarly low flow at the

second inlet port may be caused by high flow at the first inlet port, Q 1 HI (deleted because it occurs higher up the branch), or low flow at the outlet port, Q 2 LO. This is clearly incorrect because there are conflicting deviations at ports 2 and 3.

```
                            G 1 HI
                              |
                             OR
        ┌─────────────────────┼─────────────────────┐
      Q 1  HI               Q 2 HI                 G 3 LO
                              |                      |
                             OR                     OR
                      ┌───────┼───────┐      ┌───────┼───────┐
                   G 1 HI  G 2 HI  G 3 HI  Q 1 HI  Q 2 LO  Q 3 LO
```

Figure 3.9 - Fault Tree For High Flow At Port 1
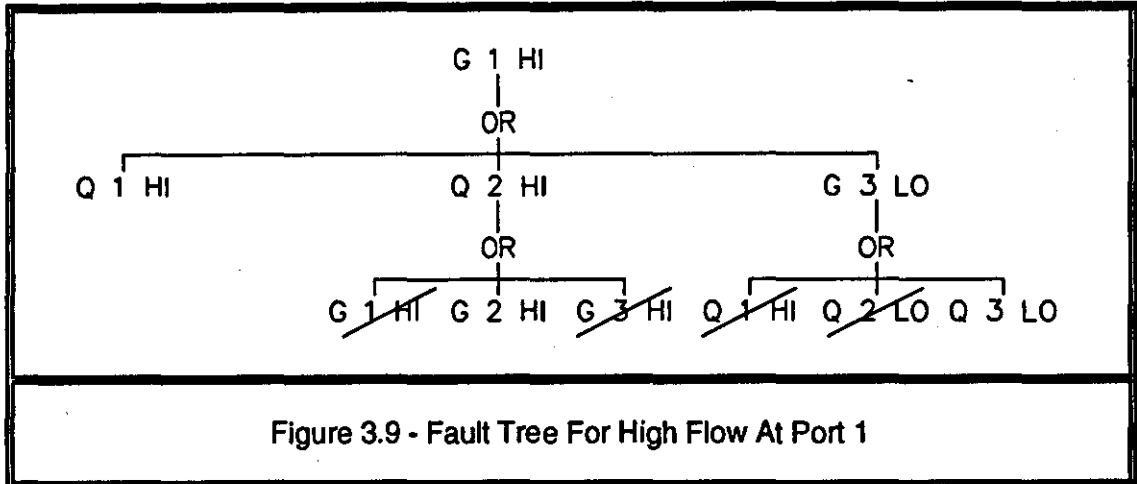
By limiting the number of port changes in any unit to just two the problem can be eliminated because the G 3 HI event and the Q 2 LO event represent the third port change and are deleted.

3.7.5 Parallel Consistency Checks

This process can only be carried out once the entire tree has been synthesised. It is used to ensure that all the branches under an AND gate are consistent with each other.

3.8 Structural Features

As mentioned previously fault tree synthesis is not just a matter of linking together the minitree from the unit models. The presence of protective devices such as control loops and trip systems resulted in problems as noted by Kelly (3). These were basically

- incorrect fault trees, due to the atomistic approach used in decomposition, when dealing with the respective failure modes
- opaque fault trees since no formal structure was applied.

By defining the protective devices as special sub-systems it is possible to model the overall failure modes as well as the individual component failures.

3.8.1 Control Loops

There are two basic types of control loop, a feedback loop and a feedforward loop. Figure 3.10 shows a feedback control loop.

Figure 3.10 - Feedback Control Loop With Separate Manipulated Stream
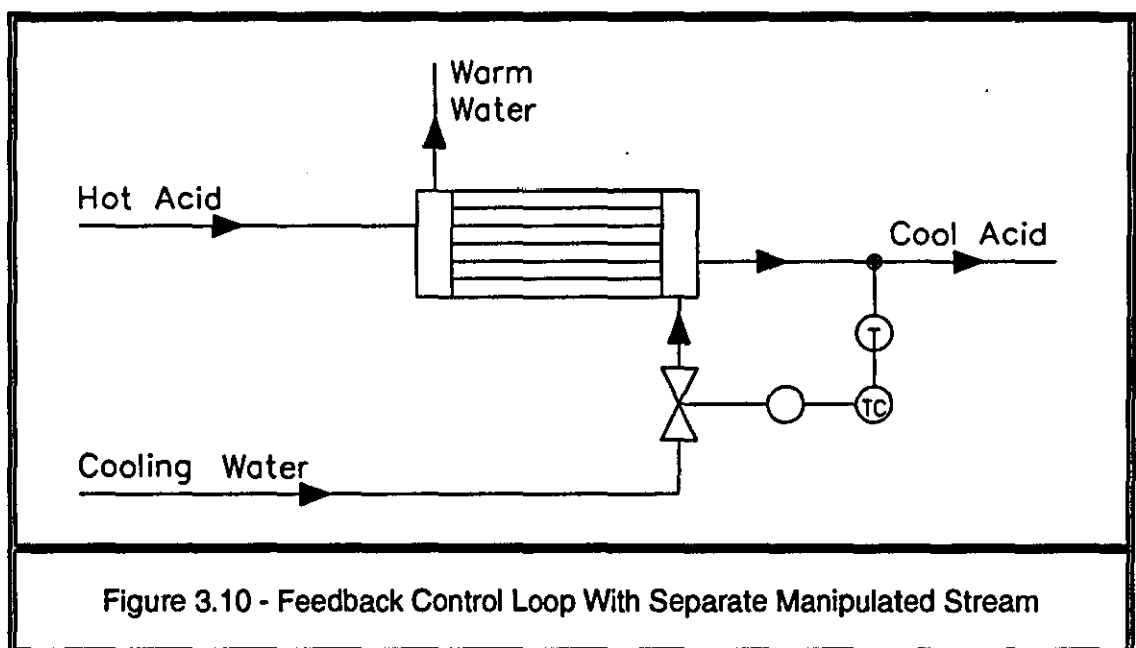
This control loop regulates the downstream temperature of the process stream by manipulating the cooling water input to the heat exchanger. The control loop is a feedback type because it is able to sense the results of its manipulative action on the coolant stream and, therefore, is able to make further adjustments based on this feedback information.

Figure 3.11 shows a feedforward control loop.
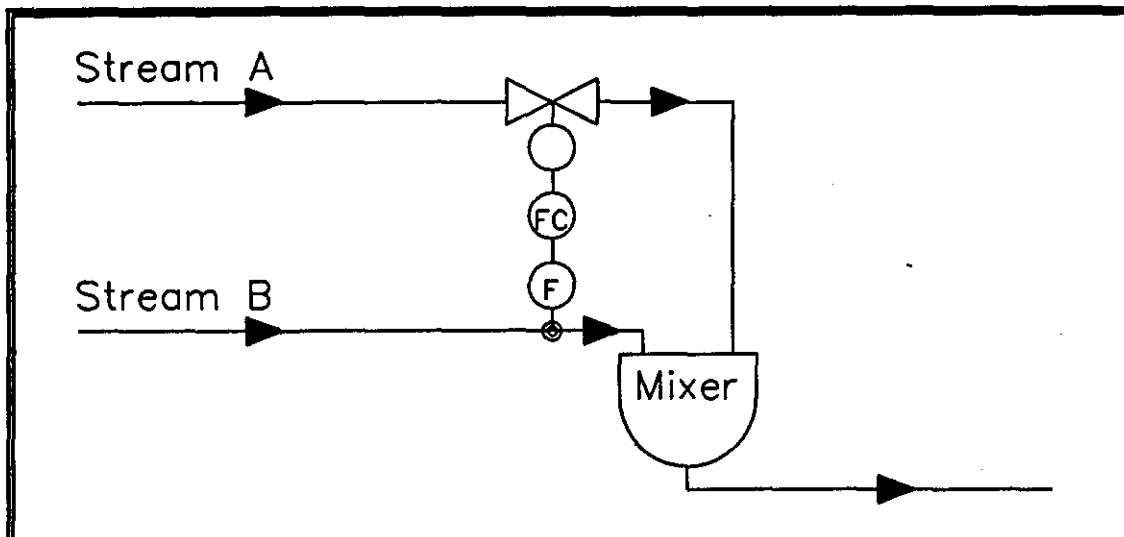
Figure 3.11 - Feedforward Control Loop With Separate Manipulated Stream

This control loop regulates the downstream composition from the mixer unit by manipulating the flow in stream A according to the value of the flow in stream B. The control loop is of the feedforward type because it is unable to sense the results of its actions.

Both the above types have separate manipulated streams. That is to say, the stream in which the control valve manipulates the flow is separate from the stream where the sensor is.

However, there are some control loops which do not have a separate manipulated stream and the distinction is important. An example is shown in Figure 3.12.
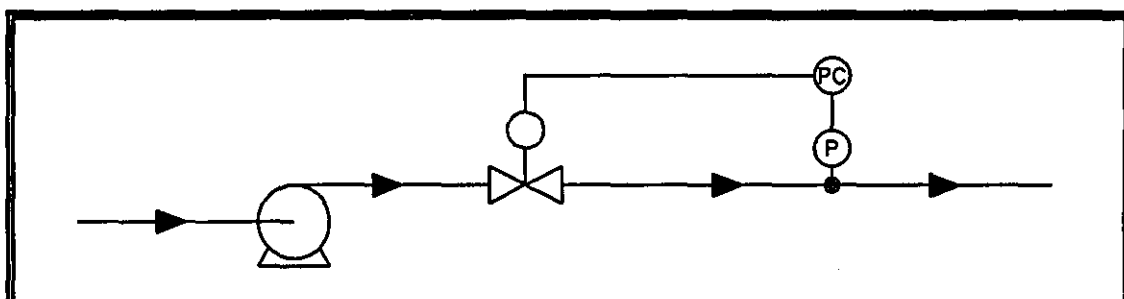
Figure 3.12 - Feedback Control Loop Without A Separate Manipulated Stream

This control loop manipulates the flow according to the sensed value of the pressure, thus regulating the pressure.

These three distinctions are important because the type of the control loop determines how the loop can fail. For this purpose the synthesis algorithm applies one of three control loop templates to structure the fault tree according to the type of control loop under consideration.

For a regulated variable deviation in a feedback or feedforward control loop the template applied is shown in Figure 3.13. The terminology is explained later.

Figure 3.13 - Control Loop Template For A Regulated Variable Deviation
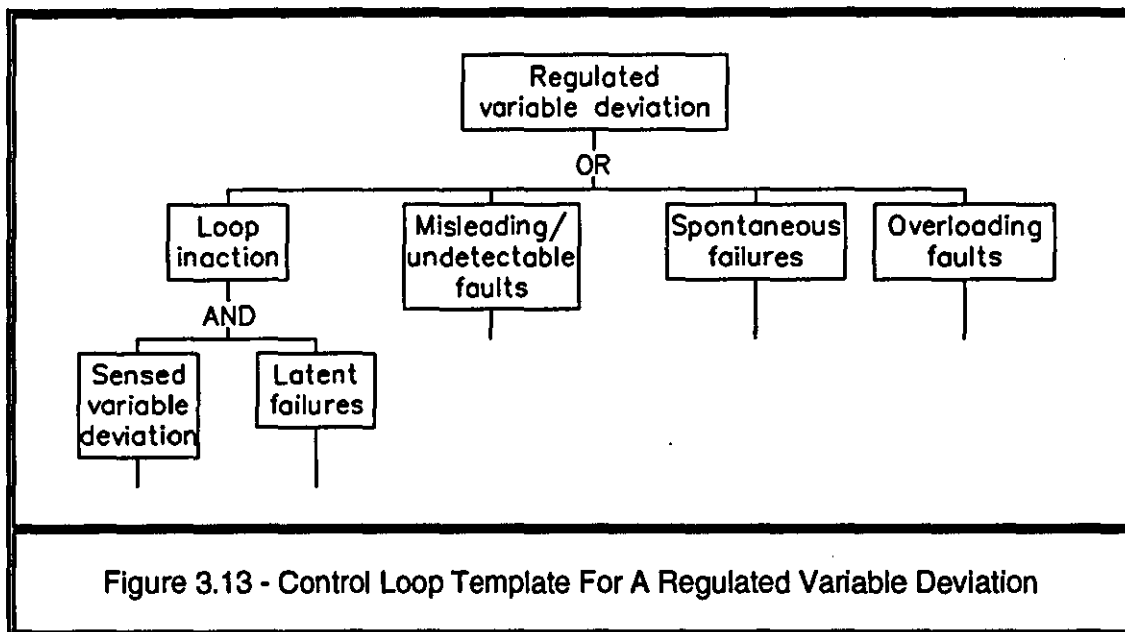
For flow deviations in the manipulated stream of a feedback control loop the template applied is shown in Figure 3.14.

Figure 3.14
Feedback Control Loop Template For A Manipulated Variable Deviation
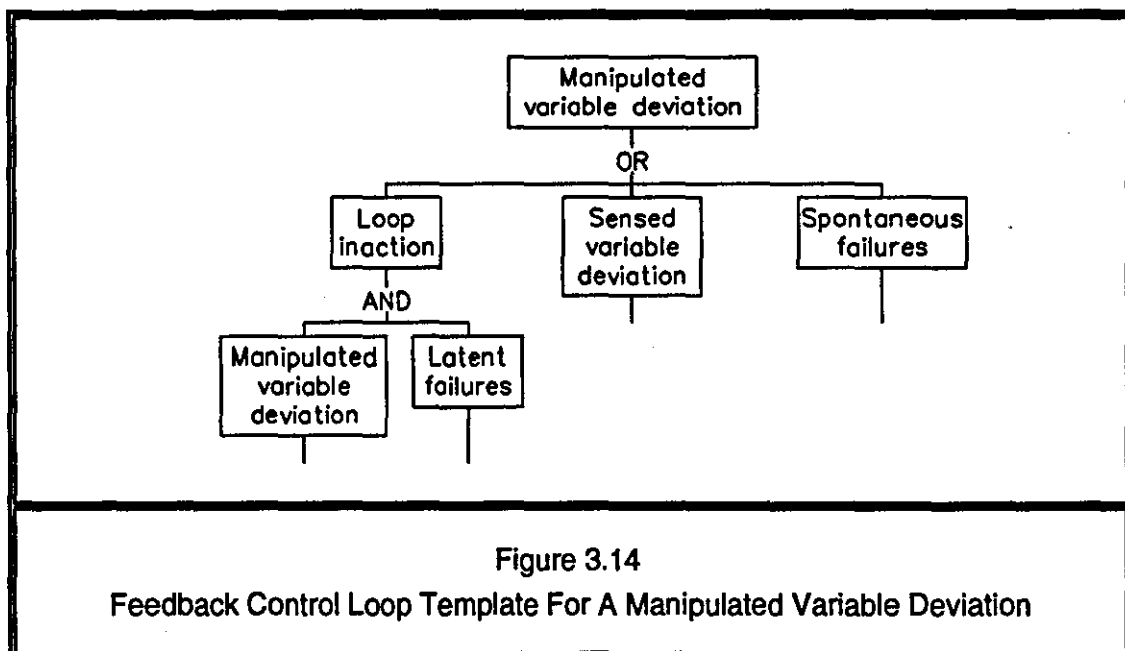
For flow deviations in the manipulated stream of a feedforward control loop the template applied is shown in Figure 3.15.


```
         ┌─────────────────────┐
         │    Manipulated      │
         │ variable  deviation │
         └─────────────────────┘
                    │
                   OR
        ┌───────────┼───────────┐
┌─────────────┐ ┌──────────┐ ┌─────────────┐
│ Manipulated │ │  Sensed  │ │ Spontaneous │
│  variable   │ │ variable │ │  failures   │
│  deviation  │ │deviation │ │             │
└─────────────┘ └──────────┘ └─────────────┘
       │             │              │
```

Figure 3.15
Feedforward Control Loop Template For A Manipulated Variable Deviation
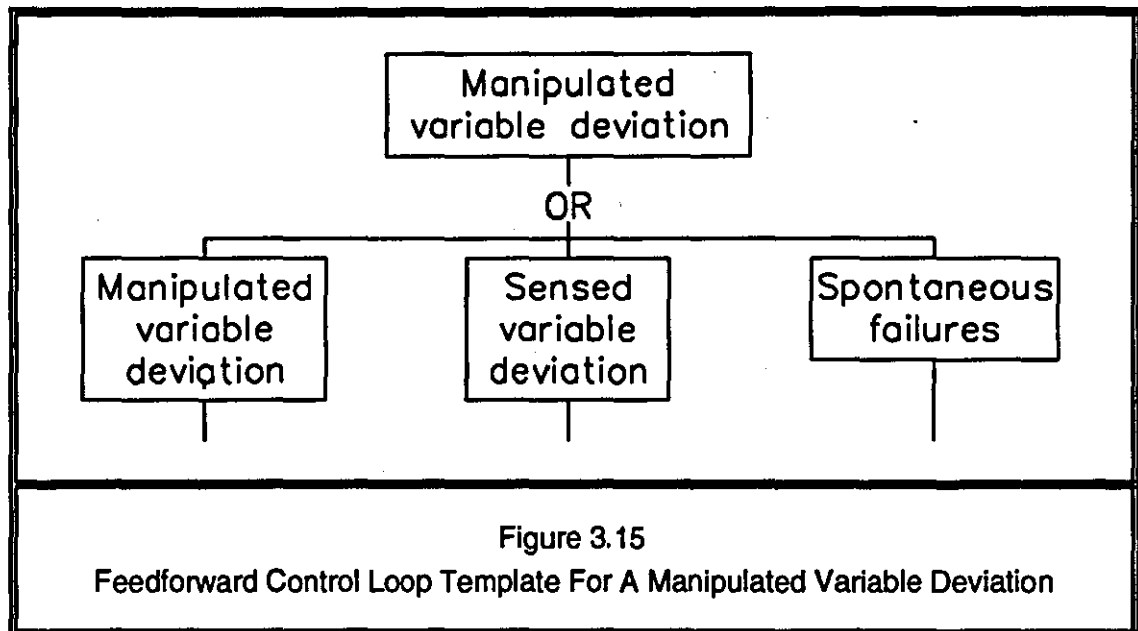
The spontaneous and latent failure branches represent the basic failure modes of the control loop due to malfunction of any of its components. The spontaneous failures are those which give rise to an erroneous increase or decrease in the control valve aperture. The two states are mutually exclusive but only one is used in any particular application of the template.

The latent failures are those which render the control loop invariant and consequently the control valve aperture remains stuck in the plant steady state position.

The overloading faults are deviations in the manipulated stream for which the control loop is unable to compensate. An example of this is complete loss of flow.

The misleading/undetectable faults represent events such as a leak in the pipe between the sensor and the control valve which causes high flow through the sensor but low flow through the control valve. The control loop will interpret this to mean there is too much flow and will inappropriately close the valve.

In feedback control the sensed variable is the same as the regulated variable. In Figure 3.13 the sensed variable deviation branch traces the causes of this deviation from the sensor. This event is correctable by the normal action of the control loop and is therefore ANDed with the latent failures branch.

The sensed variable deviation branch in Figures 3.14 and 3.15 represents the causes of the deviation in the sensed variable traced from the sensor. A deviation in this variable is correctable by the control loop and consequently is a cause of the manipulated variable deviation.

The manipulated variable deviation branch in Figures 3.14 and 3.15 represent causes of the deviation other than those due to the control loop. For a feedback control loop the manipulated stream deviation has an effect on the sensed variable and is ultimately detectable by the control loop. Therefore, it is ANDed with the latent failures of the loop. A feedforward control loop cannot detect deviations in the manipulated stream, therefore, there is no latent failures branch.

3.8.2 Trip Loops

There are two basic types of trip loop, a closed valve trip and an open valve trip. A closed valve trip has the trip valve normally closed but opens on demand and an open valve trip has the trip valve normally open but closes on demand.

An example of a closed valve trip is shown in Figure 3.16. The pump protection consists of a kick-back leg which normally has no flow but if the flow at the outlet of the pump should fall to zero the trip operates opening the valve and allowing flow.

Figure 3.16 - Closed Valve Trip Loop
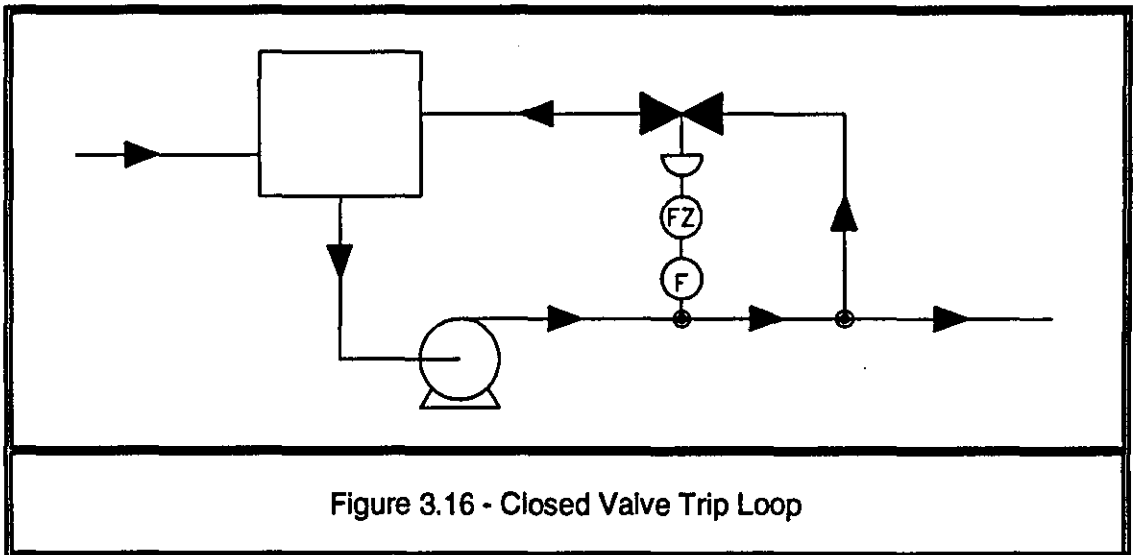
The open valve trips can be broken down into two sub-groups, those which act on the sensed variable are feedback trips and those which do not are feedforward trips.

A feedback open valve trip loop is shown in Figure 3.17. The pressure is normally controlled by a control system (not shown) but should the pressure exceed a certain value the trip activates to shut off the flow and consequently the pressure source.

Figure 3.17 - Feedback Open Valve Trip Loop
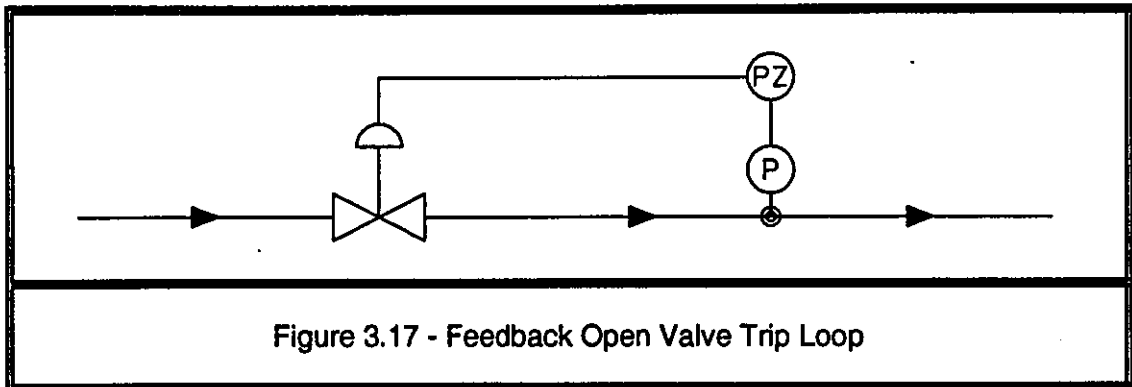
A feedforward open valve trip loop is shown in Figure 3.18. The temperature of the acid is normally controlled by a control system (not shown) but should the coolant flow fall to zero then the trip activates to shut off the flow of acid.

Figure 3.18 - Feedforward Open Valve Trip Loop
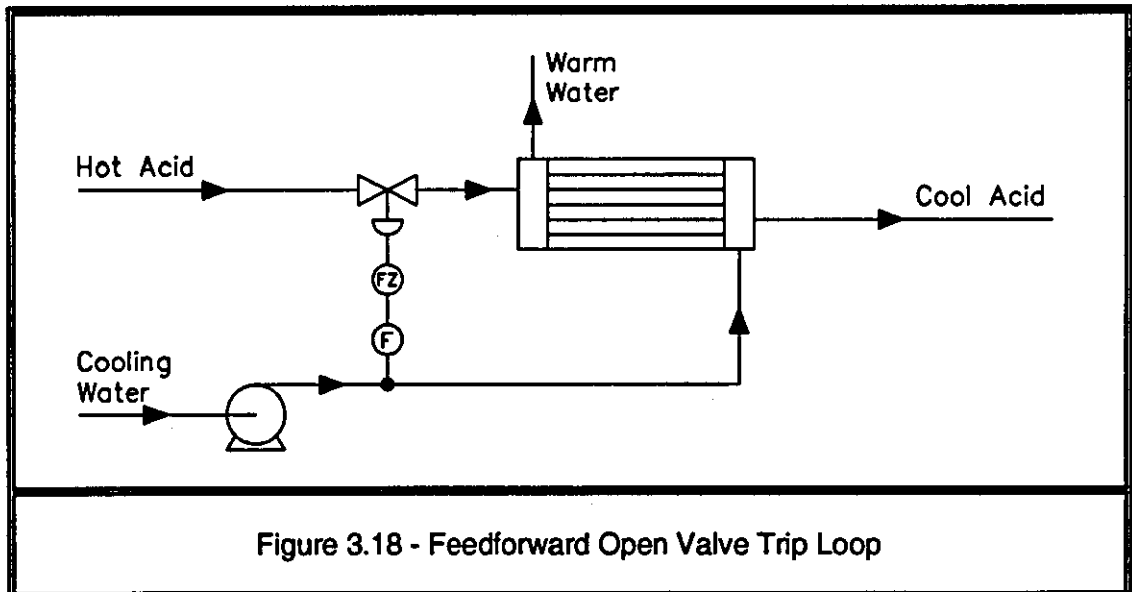
There are two types of failure mode that can occur in trip loops. Trip loop operational failure (TL-OP-F) is where the trip operates without a demand from the process. This can be adequately modelled in an atomistic approach because this is simply an alternative branch in the fault tree. The trip loop is not designed to prevent, for example, no flow in the pipeline of Figure 3.17 but its operational failure is a cause of this.

Trip loop functional failure (TL-FN-F) is where the trip fails to operate when there is a demand from the process. This requires special treatment for determining which events may be protected against and for the structure of the failure branch. The basic approach used is:

- synthesise the fault tree as normal, noting any events that propagate through the trip valve. These may be ANDed with TL-FN-F later.
- synthesise a 'should activate' or SHAC tree for each loop to define the range of conditions under which the trip system is designed to operate.
- synthesise a TL-FN-F tree for each loop to define the failures which will cause the trip to fail to operate.
- compare the SHAC tree and the events noted in the normal tree, any events the system can detect and prevent are ANDed with the TL-FN-F branch.

The difficulty in this process is in determining which events the trip can protect against. For example, in Figure 3.17 the trip can protect against high pressure propagating into the trip valve from upstream but cannot protect against a leak from a high pressure environment in the pipe between the trip valve and the sensor. Some rules have been developed by Mullhi (5) for identifying the events to be ANDed with TL-FN-F.

For a closed valve trip (Figure 3.16) there are no restrictions. The causes of deviations through the sensor are compared with the SHAC tree to see if they should be ANDed with the TL-FN-F branch. For example, no flow through the sensor is ANDed with TL-FN-F.

For a feedback open valve trip (Figure 3.17) the event is ANDed with TL-FN-F if it propagates through the trip valve and forms part of the SHAC tree. In this case high pressure into the trip valve is the appropriate event.

For a feedforward open valve trip (Figure 3.18) there are no restrictions. Any event which can cause the top event and forms part of the SHAC tree is ANDed with TL-FN-F. This is not restricted to events which propagate through the trip valve. For example, high outlet temperature from the heat exchanger may be caused by no flow of coolant into the heat exchanger. The no flow event is ANDed with TL-FN-F.

Once the trip failures have been synthesised and ANDed into the main tree a rationalisation process is carried out. This is necessary because the simple ANDing together of the protected events leads to a large and opaque tree. The rationalisation process moves the positions of the AND gates allowing excess occurrences to be removed and a more comprehensible tree to be built without changing the logic of the tree.

3.8.3 Divider-Header Combinations

The detection of divider-header combinations is automatic and the user is only required to enter the flow capacities of each leg. More detail on this process is given in Chapter 4. There are three types of divider-header combination

- by-pass with flow (eg. excess flow around a heat exchanger)
- by-pass without flow (eg. by-pass around a control valve)
- parallel flow systems (eg. multiple pump banks)

The structure imposed on the fault tree with respect to the divider-header combinations is minimal. There is just one divider and one header model used by the package to model all the types. These models have either two outlets and one inlet or two inlets and one outlet. However, distinctions are made during synthesis.

For a non-flow leg the only meaningful deviations are SOME and REV, all other deviations in this leg are deleted or replaced by SOME. A check is also made to ensure only one direction of propagation is traced.

In systems with more than two legs of parallel flow, groups of dividers and headers are used to produce the required number of legs. The synthesis package is able to identify these from the capacities entered for each leg of the dividers and headers.

Consider a two leg system with 100% capacity legs, for low flow to propagate out of the header there must be low flow in both legs (AND gate). This also applies to a three leg system. However, now consider a two leg system with 50% capacity legs. For low flow to propagate out of the header, low flow in either leg may be a cause (OR gate). However, for a three leg system low flow must occur in two of the three inlet legs (r-OUT-OF-n gate). The structure in Figure 3.19 is applied.
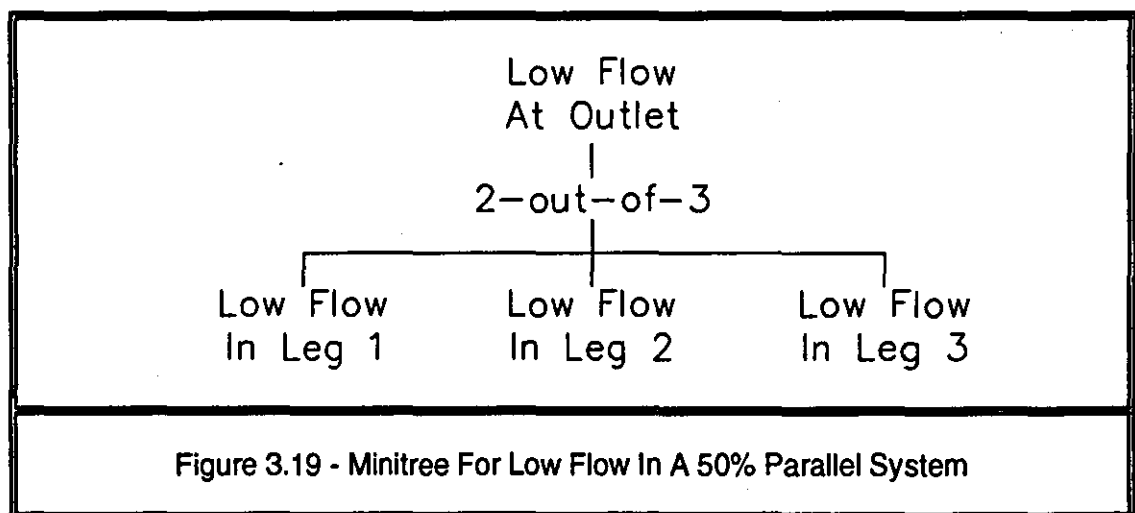
Figure 3.19 - Minitree For Low Flow In A 50% Parallel System

It is also necessary to distinguish between blockage type faults and leakage type faults. This is because they have different influences: a blockage will simply reduce the flow due to an increase in resistance but a leakage represents a loss of fluid for which the system may not be able to compensate. Consequently blockage faults for each leg are grouped under an r-OUT-OF-n gate and leakage faults under an OR gate.

The faults that occur within the divider-header combination system are also kept separate from those which occur outside of it. The imposed structure for low flow in a three leg parallel divider-header combination (DHC) system is shown in Figure 3.20.
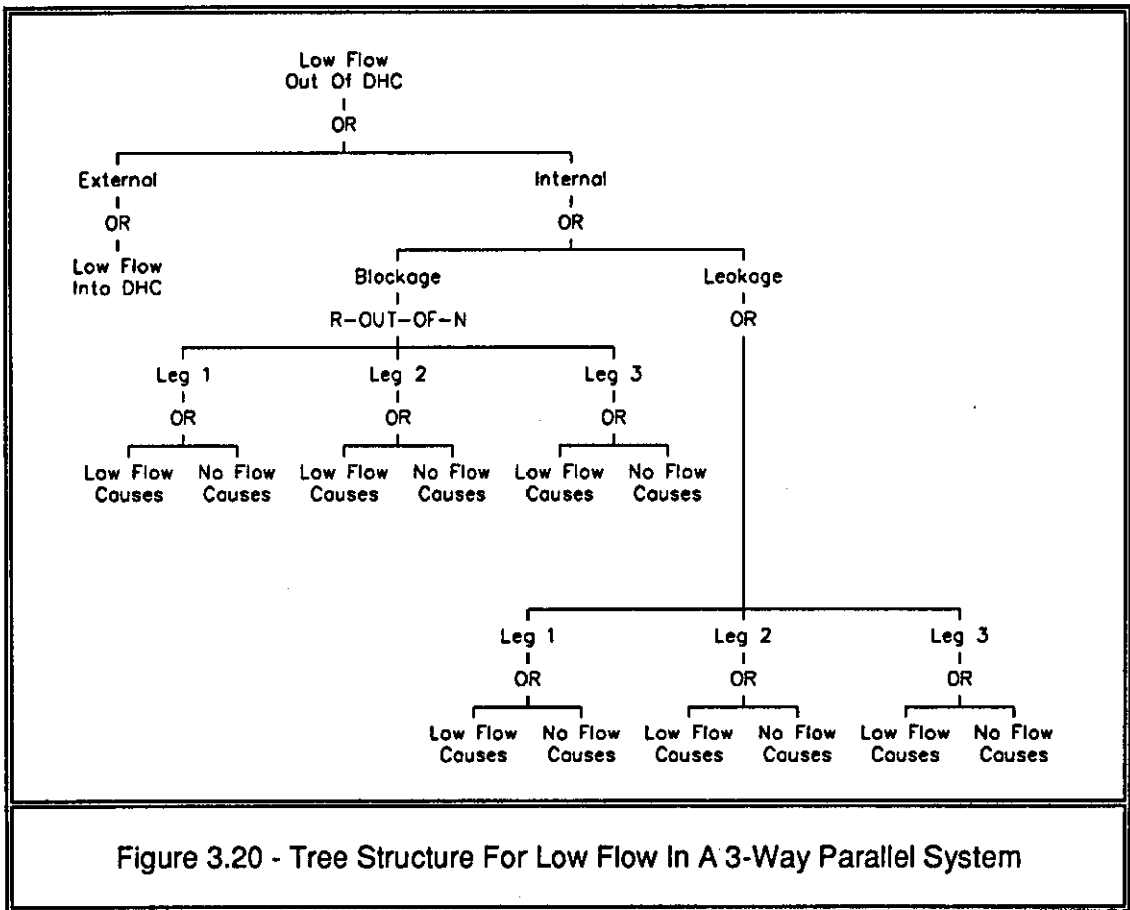
Figure 3.20 - Tree Structure For Low Flow In A 3-Way Parallel System

3.9 Advanced Features

As well as synthesising fault trees for basic steady state plant operation it is also possible to model the failure modes when the plant undergoes a series of sequential operations. There is also a facility to model secondary failures which allow additional events to be added to an existing model without altering the model.

3.9.1 Sequential Operations

This facility allows the failure states of a plant to be modelled as it undergoes a change in operation. For example, consider a pair of pumps in parallel, one operating and one stopped and isolated. A sequence of operations must be carried out to affect a change over of pumps

initial state: pump 1 operating, pump 2 stopped, pump 1 valves open, pump 2 valves closed
step 1   open pump 2 valves
step 2   turn on pump 2
step 3   turn off pump 1
step 4   close pump 1 valves
final state: pump 1 stopped, pump 2 operating, pump 1 valves closed, pump 2 valves open.

After each step has been carried out the failure mode of the plant has changed. Therefore, it is necessary to change the plant configuration. This is achieved by identifying which units change and giving the new unit model library numbers for them. It is also possible for the user to specify a different top event at each step.

For each step in the sequence the algorithm synthesises a separate sub-tree. The process has remained unchanged since its application by Kelly (3).

3.9.1 Secondary Failures

Secondary failures were introduced in order to model specific faults only applicable in certain situations. For example, a low temperature in a pipeline may result in a blockage due to freezing of the fluid. While it would be very easy to model low temperature deviations causing blockages in the pipe model this is undesirable because not all pipes are liable to freezing.

The solution was to set up a library of secondary failures which could be applied to particular sections of plant in specific applications. The failures are incorporated by specifying the units or streams which are susceptible and the particular effects of the failure. This allows different effects to be modelled for different situations. The causes of the secondary failure are contained in the model itself.

The process has remained unchanged since its introduction by Kelly (3).

3.10 Summary

This chapter has given a brief introduction to the detailed methodology used for fault propagation modelling and fault tree synthesis. The following chapters look in more detail at certain aspects and deal with some of the problems encountered.

3.11 References

1. Andow P. K., Lees F. P., Murphy C. P.; The Propagation Of Faults In Process Plants: A State Of The Art Review; Chemical Process Hazards With Special Reference To Plant Design, 7th International Symposium, UMIST, England; Institute Of Chemical Engineers Symposium Series Number 58, Rugby; 1980; pp 225–243.

2. Martin-Solis G. A.; Fault Tree Synthesis For Real Time And Design Applications; PhD Thesis, Loughborough University Of Technology; 1978.

3. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1986.

4. Kelly B. E., Lees F. P.; The Propagations Of Faults In Process Plant; Reliability Engineering, Vol 16, No 1; 1986.

5. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990.

PART 2

Chapter 4

<u>Rules For Problem Decomposition And Data Input</u>

4.1 Introduction

The first stage in any fault tree synthesis study is to turn the section of plant or whole plant into a suitable form. In the methodology this is called a 'configuration diagram'. This diagram contains information on which unit models are to be used and what connections exist between individual units.

Additional information is required on certain systems to enable their effects on the plant to be modelled correctly. These systems are

- dividers (where a stream splits), headers (where two streams join) and their combinations (eg. a by-pass around a control valve)
- control loops
- trip loops.

It is also possible to define particular types of failure that are to be included in the plant model.

The configuration diagram can be derived from any representation of the plant. The usual forms are a process flow diagram or piping and instrumentation diagram. The procedure for generating the configuration diagram is called decomposition.

4.2 Decomposition

The degree of decomposition depends on how much detail is required on the fault tree synthesis output. At one end it may be sufficient to identify complete process systems such as a reactor or separation system. These units are linked together without the interconnecting items such as valves and pipes. This is minimal decomposition. At the other end individual units such as valves can be broken down into their constituent parts like the housing, gate, stem and handle. This is full decomposition.

4.2.1 Minimal Decomposition

The purpose of minimal decomposition is to carry out a simple modelling approach without losing the functionality of the plant. If a particular process plant transfers a fluid from a storage vessel to another plant but increases its pressure and temperature on the way then the configuration diagram in Figure 4.1 will be sufficient for minimal decomposition.

Figure 4.1 - Minimal Configuration Diagram
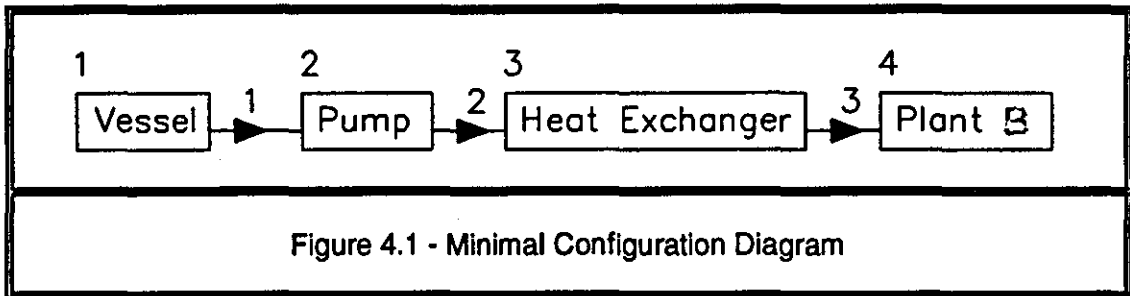
This shows the storage vessel, pump, heat exchanger and plant B. Any integral safety systems, redundant systems or the process supplies for the pump power or heating fluid are not shown. However, the functionality of a transfer from storage through a pressure raising device and heating system to the other plant is maintained.

Clearly this representation is very limited in its use. It may be more applicable to modelling an entire plant system where an overall approach is required. The advantages of minimal decomposition is that modelling is much easier and requires less time.

4.2.2 Full Decomposition

The purpose of a full decomposition is to carry out a detailed study of the individual components that make up the equipment found in a process plant. A flow sensor based on an orifice plate is made up of several components. Each side of the orifice plate has an outlet to a pipe. This is split in two with one end connected to an open hand valve and the other to a closed hand valve. The two ends passing through the open valves are connected to opposite ends of a differential pressure transducer which sends a signal to a controller or trip switch. The single closed valve forms a by-pass around the transducer. The full decomposition configuration diagram for the flow sensor is shown in Figure 4.2.

This type of decomposition gives fault trees with failures of the individual components that make up process equipment. However, this much detail requires a much greater understanding of the modelling methodology and model development can be very time consuming. There is no advantage in using this type of decomposition for large sections

of process plant because the number of models is greatly increased and the amount of time involved in modelling far outweighs the benefits. The full decomposition approach should be used to study the failures of individual units or small sections of plant.
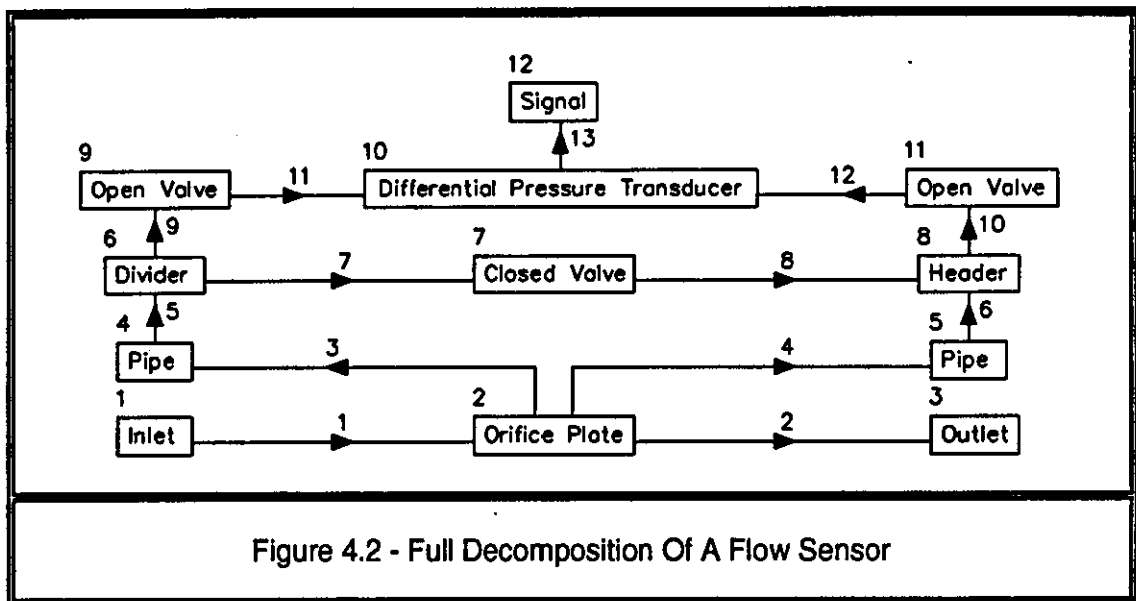
Figure 4.2 - Full Decomposition Of A Flow Sensor

4.2.3 Established Level Of Decomposition

The current level of decomposition used for modelling process plant lies between the two extremes. Using the piping and instrumentation diagram it is possible to get an almost one-to-one correspondence between the items of equipment.

For example, shell and tube heat exchangers are shown as a single unit with an inlet and outlet for the coolant and an inlet and outlet for the process fluid. There may also be relief or drain ports. The decomposition does not model individual tubes but does model the control system maintaining a constant outlet temperature for the process fluid. The failures of the heat exchanger are modelled as an internal leak between streams, external leak to the environment, fouling and partial blockage plus variable deviations caused by other deviations entering the unit.

Flow sensors, in contrast, are modelled as single units. Pipe mounted sensors are placed next to a pipe unit in what can be considered as the process flow stream. The failures are simply kept to the sensor giving a false high or low signal output.

Controllers and trip switches are modelled as two units. There is the actual logic unit, the controller or trip switch and there is the set point for the unit. In process plant some failures can be attributed to an incorrectly assigned set point. Therefore a separate unit is used for this.

This level of decomposition can be illustrated using the traditional nitric acid cooler system first used by Lapp and Powers [1]. The piping and instrumentation diagram is shown in Figure 4.3. Hot nitric acid is cooled in a heat exchanger using water. A temperature sensor on the acid outlet sends a signal to a controller which manipulates a control valve on the water inlet stream. A trip valve on the acid inlet stream operates should the flow of cooling water fall to zero. The configuration diagram is given in Figure 4.4.
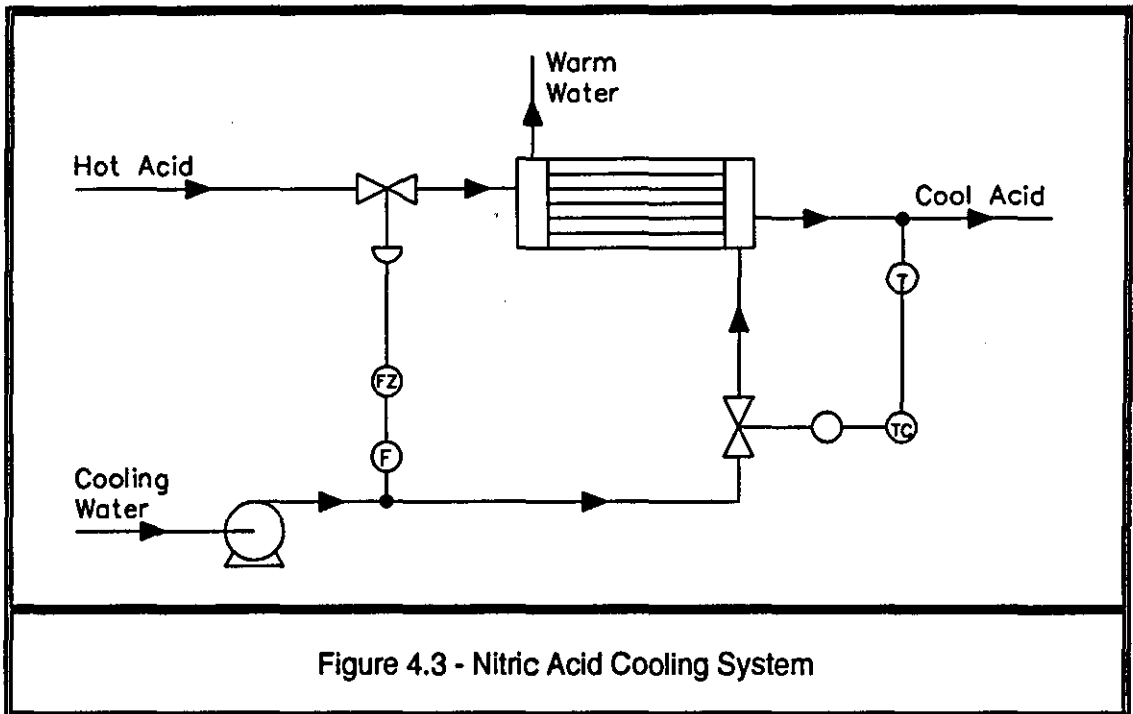
Figure 4.3 - Nitric Acid Cooling System
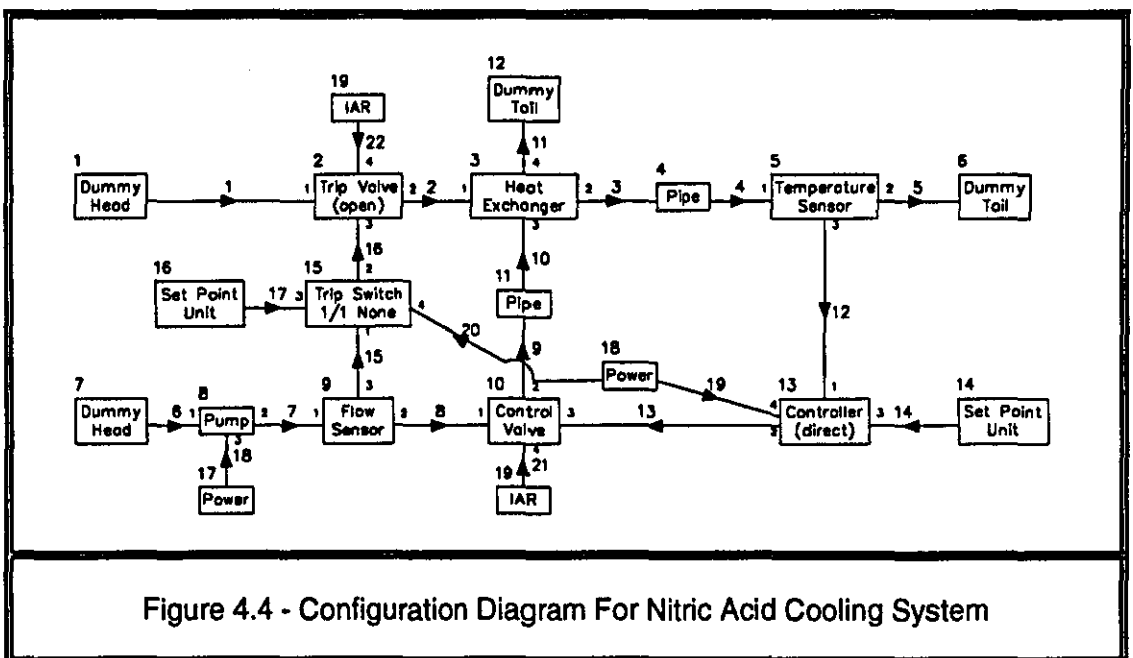
Figure 4.4 - Configuration Diagram For Nitric Acid Cooling System

4.3 Rules For Generating The Configuration Diagram

When considering generating a configuration diagram for a section of plant a few rules have been developed. These will be illustrated using Figure 4.4 as an example.

The configuration diagram is made up of two types of item. Units are the physical entities that make up the plant. These are the individual blocks which the user has identified as components such as reactors and valves.

Connections are the logical links that exist between units. These are not necessarily pipes but may be any type of link between units. It should be noted that process pipes, signal lines and electric cable can be units and are then included in the configuration diagram as such.

It is often the case that process lines cross the boundary of the plant section to be studied. The process is not concerned with causes from outside this boundary so a marker must be used to identify the section boundary. To do this dummy heads and dummy tails are used. A dummy head represents an input to the plant section. It is often the source of the process streams and has no inputs only outputs. A dummy tail represents an exit from the plant section. It is often the sink for the process streams and has no outputs only inputs. These units have no failure modes but allow the variable deviations to exist as diamond events in a fault tree, ie. the causes are not developed any further because they occur outside the section boundary.

The revised model library contains supply units such as a process nitrogen supply and sink type units such as a flare or drain. These can be used to replace the dummy head or dummy tail units to give a specific source or destination to the stream. They also contain some basic faults.

The majority of a process plant is made up of piping between units. It is very straight-forward to place a pipe unit between every unit where a pipe connection exists. However, this can generate a very large number of units which takes time to enter into the computer and to run. The same pipe model is used each time and this creates multiple occurrences of the same fault which makes the fault tree generated large and opaque. The number of pipe models used can be reduced by only using the unit at strategic places where blockages or leaks can have a profound effect. In Figure 4.4 the pipe is placed at the heat exchanger inlet for the water and the outlet for the acid.

Every unit in the configuration diagram must have a corresponding unit model in the unit model library. This is often the same model, for example, the pipe model and the set point model.

Each unit has a number of interfaces called 'ports'. These allow connections to other units. The models for each unit contain the links between ports. For example, a pipe unit has two ports, one for the inlet flow and one for the outlet flow. A heat exchanger has four ports, an inlet and outlet for the cold stream and an inlet and outlet for the hot stream. The flows of the two streams are independent but the temperature is cross-linked in the model.

The ports for each model are numbered sequentially and should be marked in the configuration diagram if confusion is likely. This enables the correct units and ports to be linked together. It is important that the correct ports are used otherwise the wrong minitree will be used from the unit model.

Each port on a unit must be connected to another unit. This is necessary because for a variable deviation to appear in the fault tree it must be associated with a connection number. Problems do not arise with inlet, outlet and signal ports because these are always connected to units as a matter of routine. The problem occurs with vessel ports. A user may have a vessel type unit which does not have any measurement device attached to it and quite rightly will not connect a unit to its vessel port. However, when the synthesis package finds, for example, high level causes high flow out of the vessel it will not have a connection number for the level deviation. This causes the package to label the level deviation as impossible. Dummy heads and dummy tails are convenient for connecting to this type of port to prevent this happening. An exception to this rule is the utility port. Many units have utility ports built into them which are often not required. If the user does not connect a dummy unit to this port the synthesis package ignores it and the deviations associated with it.

The units and connections are numbered sequentially and independent of each other. The unit numbers appear at the top left hand corner of the unit block and the connection number appears either above the arrow giving the normal direction of flow or to the right of the arrow on a vertical line. Using this system prevents confusion. The numbering should generally go from left to right in the same way as a piping and instrumentation diagram. It should start with the process flow streams, then the control and trip systems and finally the utilities.

From Figure 4.4 it can be seen that the power utility, unit 18, is connected to both the controller and the trip switch. This allows for common mode failures and each utility unit can be connected to up to nine different units. The same would be expected of the instrument air utility, IAR. However joining the control valve and trip valve to the same unit will over-crowd the configuration diagram so in this case the IAR unit has been given

twice but with the same unit number. This only affects the diagram because the data input file will have the same format for the two utilities. It should be noted that different port numbers must be used for the two valve connections to the utility.

Common mode susceptibility due to maintenance errors can also be modelled. This, for example, may take the form of a single set point unit for all, or a group of, controller and trip units.

4.4 Rules For Defining The Configuration

The configuration diagram is entered using the configuration data input program MASTER. Once the data has been entered it is not necessary to repeat the process unless the configuration is changed dramatically because the data is stored in a file specified by the user. This file can be edited to change things such as model library numbers or correct errors during the input.

A more detailed account of the rules for defining the control and trip loop data and the various types of loop is given by Mullhi (3, 4).

The following sub-sections are divided according to the prompts given by the program for the user to respond to. The separate sections deal with the basic dimensions of the configuration such as the number of units and the number of control loops and then a more detailed analysis of the individual items such as the model library numbers of each unit and the make-up of each control loop.

4.4.1 Basic Data

The first section requires the total number of individual units along with the total number of connections between these units.

The process of fault tree synthesis is very complex and it is necessary to identify certain combinations of units which act together as protective systems. These are the control loops and trip systems.

The basic constituents of a control loop are a sensor, controller and a control valve. Any one of these may be shared with others but each combination must be identified as a separate control loop. The total number of loops must be given.

The basic constituents of a trip system are a sensor, trip switch and a trip valve. Any one of these may be shared with others but each combination must be identified as a separate trip system. Trip systems are sub-divided into open valve trips and closed valve trips. The open valve trip has an open trip valve which slams shut on a trip signal. Closed valve trips have a closed trip valve which opens allowing flow on a trip signal. The total number of each type of system must be given.

The final data required in this section is for the secondary failures. These allow additional causes of specified variable deviations or faults to be used. A Type I secondary failure deals with variable deviations and a Type II failure deals with basic faults. The total number of Type I failures and the total number of Type II failures must be entered. More detail on the secondary failures is given in section 5.8.

4.4.2 Unit Model Library Numbers

This section requires the units given in the configuration diagram to be given the appropriate unit model library reference number. The program gives each unit number and the user supplies the library model number. The library listing is given in Appendix B.2.

4.4.3 Connection Topology

In this section the connections are required. For each connection number, the start or upstream unit and port number and the end or downstream unit and port number are entered.

The first three connection specifications for the hot nitric acid cooler in Figure 4.4 are given below.

Upstream		Downstream	
Unit	Port	Unit	Port
1	1	2	1
2	2	3	1
3	2	4	1

4.4.4 Control loops

The control loops are identified as a unique combination of sensor, controller and control valve. The number of these combinations has been identified and the input order should be determined by considering the importance of each loop. The primary control loops should be given first. In the case of a cascade control loop the master loop should be entered before the slave loop.

The synthesis algorithm requires the sensor unit number, the sensed variable and the control valve unit number. In order to determine which unit failures will result in loop failure the other units in the control system must be identified. This includes the controller, set point and any utilities.

There are two types of control variable that can be used. Manipulated control involves flow only. This always occurs because the control valve manipulates the flow passing through it. All the streams that the flow can be manipulated in must be specified. The general principle is to give all the stream connections upstream and downstream of the control valve up to a flow changing unit. A flow changing unit is defined as one where there can be an accumulation of fluid such as a vessel, a splitting or joining of streams such as in a divider or header or where the flow is manipulated by another control valve. It is up to the user to use discretion when a long list of connections is encountered and only give a reasonable number. The maximum permitted by the program is ten connections.

The regulated control variable is usually the sensed variable because this is ultimately what the controller is aiming to control. Usually the manipulated flow stream affects the value of the sensed variable. All the connections in which the regulated variable can be influenced by the controller must be specified. This will include connections upstream and downstream to units where the variable can be changed. For example, temperature can be changed in a heat exchanger and composition in a mixer. If the regulated variable is sensed at a vessel mounted sensor, it may be such that the variable does not occur outside the vessel, eg level. In this case there are no connections where the variable is regulated. Therefore the connection refers to the connection between the vessel port and the vessel mounted sensor.

In some cases two variables are very closely related in a unit. An example which has been referred to by Mullhi (3) is the temperature and composition in a distillation column. The approach used by Mullhi was to model both the variables as regulated variables in the same control loop. The loop sensed the temperature in the column and adjusted the flow of heating medium to the reboiler. This not only affected the temperature at the bottom of the distillation column but the composition as well, so both variables are

regulated. However, this requires the user to have additional knowledge on the properties of the system. A more suitable method is to model the effects of temperature on the composition in the distillation column itself. Once the column has been modelled in this way the user will only have to consider the sensed variable as the regulated variable.

The final information about the control loop is whether it is a feedforward or feedback loop. A feedback loop is one where the effects of the control manipulation is sensed by the sensor, hence, information is 'fed back' to the controller about its actions. This is the normal format because the loop is less susceptible to outside influences. A feedforward loop cannot sense the results of its actions. The sensor and control valve are in separate streams with no upstream interaction.

Using Figure 4.4 as an example the temperature control loop has the following data input.

CONTROL LOOP

SENSED VARIABLE: T
VARIABLE SENSED IN UNIT: 5
CONTROL VALVE UNIT NUMBER: 10
OTHER UNITS IN CONTROL LOOP: 13 14 18 19
VARIABLE T REGULATED IN CONNECTIONS: 3 4 5
FLOW MANIPULATED IN CONNECTIONS: 6 7 8 9 10 11
LOOP IS NOT OF FEEDFORWARD TYPE

4.4.5 Open Valve Trip Systems

The open valve trip systems are identified as a unique combination of sensor, trip switch and open trip valve. The number of these combinations has been identified and the input order should be determined by considering the order in which they would act. The final protective system should be given first because this will then appear near the top of the fault tree with the others appearing lower down.

The synthesis algorithm requires the trip valve unit number to be given separately from the list of other units in the system. The other units will include the sensor, trip switch, set point and utilities.

The other piece of information deals with feedback or feedforward systems. A feedforward trip has the sensor in a different stream to the trip valve.

Using Figure 4.4 as an example the open valve trip system has the following data input.

```
OPEN VALVE TRIP SYSTEM

TRIP VALVE UNIT NUMBER:  2
OTHER UNITS IN TRIP SYSTEM:  9  15  16  18  19
TRIP IS OF THE FEEDFORWARD TYPE
```

4.4.6 Closed Valve Trip Systems

The closed valve trip systems are identified and numbered in the same way as the open valve trip system.

Again the trip valve unit number is given separately from the list of other units in the system. The other units will include the sensor, trip switch, set point and utilities.

The feedback or feedforward difference is not required but the synthesis algorithm requires the connections which will have flow when the valve opens to be identified. This includes all connections both upstream and downstream of the valve.

There is not a trip system of this type but an example is given to show the layout.

```
CLOSED VALVE TRIP SYSTEM

TRIP VALVE UNIT NUMBER:  25
OTHER UNITS IN TRIP SYSTEM:  20  23  24  30  34
CONNECTIONS HAVING FLOW WHEN VALVE OPENS:  27  28  29
```

4.4.7 Dividers And Headers

Dividers are units which split one process stream into two. Headers are units which join two process streams into one. Divider-header combinations are mixtures of dividers and headers which form possible flow loops. They may be simple, such as a by-pass around a control valve, or more complex, such as an excess flow relief system for a pump bank. The identification of dividers, headers and their combinations has been automated, a detailed account is given by Mullhi (2). The information required for these units is the relative flow capacities of each leg. The capacities can be given as 0%, 50% and 100%.

For a divider the flow capacity is a percentage of the inlet flow. If both outlet legs have a capacity for allowing all of the inlet flow through either of them, then a 100% capacity is

4-11

used. If, for example, on a pump bank the two pumps can only provide 50% of the flow each, then the legs should be given as 50% capacity. For legs connected to closed valves resulting in no flow, then the capacity is 0%.

For a header the flow capacity is a percentage of the outlet flow. For full capacity the value is 100%, for split capacity the value is 50% and for non-flow legs the capacity is 0%. It should be noted that dividers and headers connected in a divider-header combination must have the same flow capacities.

These percentages are used by the synthesis package to determine which type of logic gate should be used when combining the flow and pressure deviations of the two legs.

This process of using numbers may be confusing to a user in some cases. For example, if the dividers and headers form part of a divider-header combination for a three way pump bank where each pump delivers 33% of the capacity the user may be inclined to give the capacity as 33%. However the package will not accept this.

A solution to this is to remove any reference to the numbers and replace it with text. The format adapted is to define the capacities as 'none', 'shared' and 'full'. The program converts these to the numerical form of 0, 50 and 100 respectively.

4.4.8 Secondary Failures

Each secondary failure has to be entered separately. After the name of the failure is given the connections or units in which the failure can occur are specified. The effects of this secondary failure are simply the combination of a variable letter and a deviation or a basic fault. The port number and port type are not required. Multiple effects can be used, for example, T HI and U HI for high temperature in the normal direction of flow and the reverse direction. More detail on secondary failure modelling is given in Section 5.8.

For the nitric acid cooling system given in Figure 4.4 two secondary failures can be used. If the heat exchanger gets an internal leak then water may enter the acid causing an exothermic reaction and rise in temperature. The Type I secondary failure used is DILUTION which is caused by high concentration of water. The connections affected by this are 1 to 5 and the variable deviations are T HI and U HI.

The other failure deals with the acid entering the cooling water stream causing leaks in the pipe work. The Type II secondary failure used is CORROSN which is caused by high concentration of acid. The units affected by this are 3 and 7 to 12 and the basic fault is a leak to a low pressure environment LK-LP-EN.

The secondary failures are entered in the following form

```
TYPE I EFFECTS
NAME: DILUTION
EFFECTS: T HI, U HI
CONNECTIONS:  1  2  3  4  5

TYPE II EFFECTS
NAME: CORROSN
SUSCEPTIBLE FAULTS: LK-LP-EN
UNITS:  3  7  8  9  10  11  12
```

4.5 Automatic Identification

This chapter has dealt with the development and data input of the configuration diagram. It can be seen that the amount of work required from the user is considerable. Therefore, studies have been carried out to reduce this.

4.5.1 Divider-Header Combinations

The identification of dividers, headers and their combinations has been developed by Mullhi (2). This process searches the connection topology array (created in section 4.4.3) for units which either have one inlet and two outlet connections or two inlets and one outlet connection. The former is identified as a divider unit and the latter as a header unit. Next the models of the units are checked and if the port types of the two outlets or two inlets do not match the unit is discarded because it cannot be a divider or header and is more likely to be a valve or sensor. The system has had to be modified because the process not only picked out dividers and headers for process flow streams but also signal lines.

The next step is to identify the divider-header combination loops. The connection topology identified as part of a divider or header is transferred to a new array such that this array only contains connections at these units. This removes all the flow-through units and intervening connections. Divider-header combination loops are identified by tracing from the outlet of each divider around the connections until it returns to the other outlet of the same divider. If it encounters a connection already crossed, the search terminates and starts on a new one. The loops identified are sorted and the duplicate ones removed.

Loops are required because they form routes where flow and pressure deviations can travel. They are used to alert the synthesis program to possible inconsistencies in deviations of the same variable. This is normally dealt with by the unit models but in these cases more than one unit is involved. An example is low flow out of a divider-header

combination by-pass around a control valve. The low flow deviation may be caused by reverse flow through the by-pass leg of the header unit ANDed with the normally closed valve failing open. A cause of reverse flow in the by-pass is reverse flow into the divider-header combination. However, reverse flow into the divider-header combination cannot be a cause of low flow out of it. The ability to identify loops alerts the program to the possibility of this inconsistency and the trace of reverse flow is stopped at the divider.

4.5.2 Control And Trip Loops

The principle for tracing along connection pathways for divider-header combinations has been used to test the feasibility for automating part of the control and trip loop specification. The study was carried out on control loops by Mullhi (3) but can equally be applied to trip loops. The work involved identifying the sensors and control valves. Then for each sensor a signal path is traced to a control valve. Each different sensor to valve route produces a new control loop. This tracing sequence will also identify the other units in the control configuration.

The sensed variable can be identified by examining the sensor model to determine which variable the signal is a function of. The connections where this variable is regulated can be determined by tracing upstream and downstream from the sensor. However, complications arise when dividers, headers, mixers, reactors and other vessel type units occur due to their influences on certain variables. The manipulated connections can be determined in a similar way but the restrictions placed on the units only apply to those influencing flow.

4.5.3 Input Of Configuration

Studies have also been carried out on using the output from a computer-aided drawing package. A similar configuration storage system is used for the units and the connectivity between them. The major difference is the representation of the unit type. The drawing package, for example, may use a single symbol for a shell and tube heat exchanger but for fault tree synthesis distinctions must be made for the different types.

Work is currently proceeding on the automated facilities.

4.6 Reduction Of Data Input Repetition

Currently the procedure adopted for analysis is to study a small section of plant and then enlarge this to cover all the plant. This procedure is used because it is easier to test new models in small configurations than it is in larger configurations. This results in the user having to enter the configuration diagram more than once because each individual plant section requires a new configuration data file. Most of the terminal work occurs at this stage of the synthesis.

Ideally it would be more acceptable if the entire plant configuration could be entered all at once in one data file and then certain sections be isolated for study. This would eliminate the need for multiple configurations for the different sections.

A break point system has been developed which allows the user to create imaginary boundaries around specific sections of plant and therefore isolate it for study. Only models of units within this section have to be contained in the model library which enables one user to develop a fault tree for one section of the plant while another is developing models for another section.

4.6.1 Break Points

The current format of the programs uses a system of dummy head and dummy tail units to mark the boundaries of a configuration beyond which the user is not interested. It seems sensible, therefore, to use dummy heads and tails to mark the new boundaries of the section of configuration of interest.

These two units are treated as special cases when encountered during synthesis of the fault tree because they are not items of equipment and, therefore, do not contain any basic faults. When a variable deviation is being traced along a flow stream and a dummy head or dummy tail unit is encountered the package labels the deviation relating to the current stream as a diamond event. When this type of event is drawn on a fault tree it informs the user that the cause of that variable deviation appears outside the bounds of the configuration.

4.6.2 Choosing A Break Point Unit

The dummy head and dummy tail units are also special types of model and cannot be used to replace just any type of unit. A dummy head has only one port leaving it and a dummy tail has only one port entering it. Units analogous to dummy units are the utilities and the termination units.

The next degree of complexity of a unit is one with two ports, one for the inlet and one for the outlet. Such units include pipes, hand valves and sample points. These units usually appear in abundance in configuration diagrams and, therefore, it seems ideal to use these units as break point units.

If the choice of break point unit is expanded to cover units with more than two ports then the programming required to alter the data files becomes complicated. Since the two port unit is common in a configuration diagram it is reasonable to limit break point units to this type. Therefore, break point units have been defined as those units which have one inlet port and one outlet port and no other type of port.

A point to note is that connections must not cross the boundary line. This means that control and trip loops must be entirely contained within the boundary or entirely excluded from it. It is permissible, however, for regulated, manipulated or flow streams to be truncated by the boundary line with a break point unit.

4.6.3 Data Alteration

Once the break point units have been identified some of the data on the configuration has to be altered to afford the same special treatment to these units as the dummy head and dummy tail units. The break point units are defined as either a new dummy head or a new dummy tail.

For the new dummy heads the unit model library reference number has to be changed to that of the dummy head unit. This assigns the minitrees for the dummy head unit model to the new dummy head unit. Next the array containing the list of connection numbers coming from a dummy head unit is modified to contain the outlet connection number from the new dummy head unit. This array alerts the synthesis package that special treatment is required for this connection number. The new dummy head unit is still connected to a unit upstream so to release this link the connection number at the inlet to the new dummy head unit has its downstream unit number (ie. the new dummy head unit number) set to zero in the connection topology array. This prevents the automatic divider-header combination search tracing outside the new section boundary. The new dummy head unit is now isolated from events outside the boundary but its port numbers are wrong. The dummy head unit has a single outlet port numbered port 1 but the outlet port on the new dummy head is port 2. In the connection topology array the connection number for the outlet from the new dummy head unit has the upstream unit (ie. the new dummy head unit) port number changed to 1. To complete the change the connection for the inlet to the new dummy head unit has its downstream unit (ie. the new dummy head unit) port number changed to 2.

This completes the installation of the new dummy head unit.

A similar process is carried out for the new dummy tail unit. The unit model library reference number is changed to that of the dummy tail unit model. The array containing a list of connection numbers going to a dummy tail unit is modified to contain the inlet to the new dummy tail unit. The connection number at the outlet of the new dummy tail unit has the upstream unit (ie the new dummy tail unit) set to zero in the connection topology array. The port numbers do not need to be changed.

This completes the installation of the new dummy tail unit.

Having defined the new boundary points it is necessary to prevent the configuration data input program MASTER reading in the unit models of those units which no longer take part in the configuration. This is achieved by specifying which units are to be excluded and then setting the library reference number of the units to zero. The program sets up an array of markers for each library number used and it only retrieves models of those units. The lowest model number is one so although there will be a marker for model zero the program will ignore it.

4.6.4 Subroutine Flowsheet

Figure 4.5 is a schematic representation of the subroutine flowsheet used to carry out the break point application. The code used for the subroutine is given in Appendix A.1.

4.6.5 Alterations To The Code

In order to implement the break point system into the fault tree synthesis code the following alterations have had to be carried out.

The call for the new break point subroutine has to be inserted immediately after the configuration data has been entered into the program and before any operations are carried out on the data. This is because some operations involve dummy head and dummy tail units so the definition of the new dummy units have to occur before this.

Since the library reference number has been set to zero for the redundant units it is necessary to change the dimensions of the array which marks whether the unit is used from 1:500 to 0:500. This allows the program to mark the model numbered zero as being used but the rest of the program only uses marks between 1 and 500.
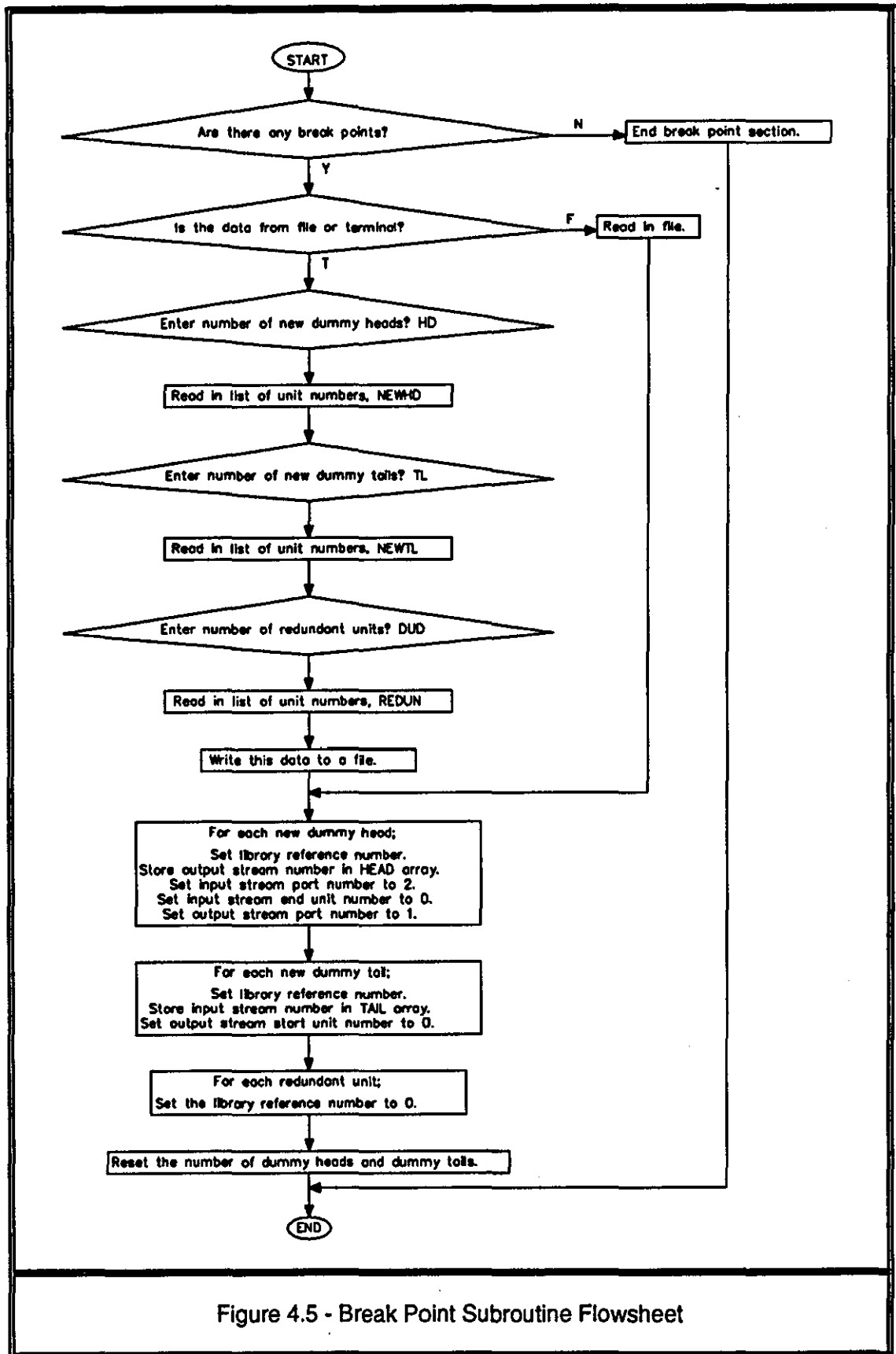
Figure 4.5 - Break Point Subroutine Flowsheet

The divider-header combination search identifies dividers and headers by searching the connections array for streams with the same unit numbers. A problem occurred here initially because there was nothing in the array to indicate where the new boundaries were and the searches went outside of them. This was remedied by setting the unit number of the redundant stream connected to the new dummy units to zero. A check was then placed in the divider unit and header unit search routines that abandons the search if a unit number of zero is encountered.

4.6.6 Example Of Break Point Use

Consider the section of plant shown in Figure 4.6.
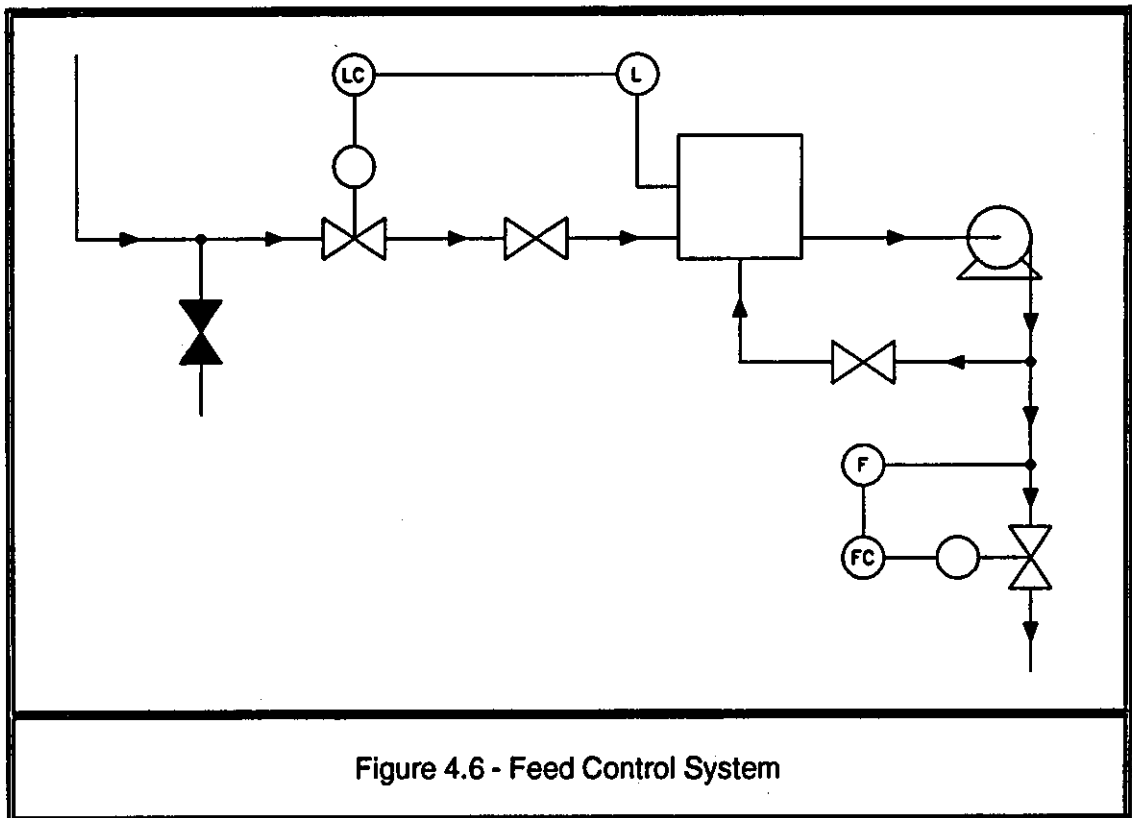
Figure 4.6 - Feed Control System

It consists of a long length of pipeline to a buffer tank whose level is controlled by means of a control valve at the inlet to the vessel. The outlet is pumped through a flow control system but there is a kick-back leg to the vessel to divert excess flow and protect the pump from over-pressure. All the lines have a capacity for 100% of the flow. The configuration diagram is shown in Figure 4.7.
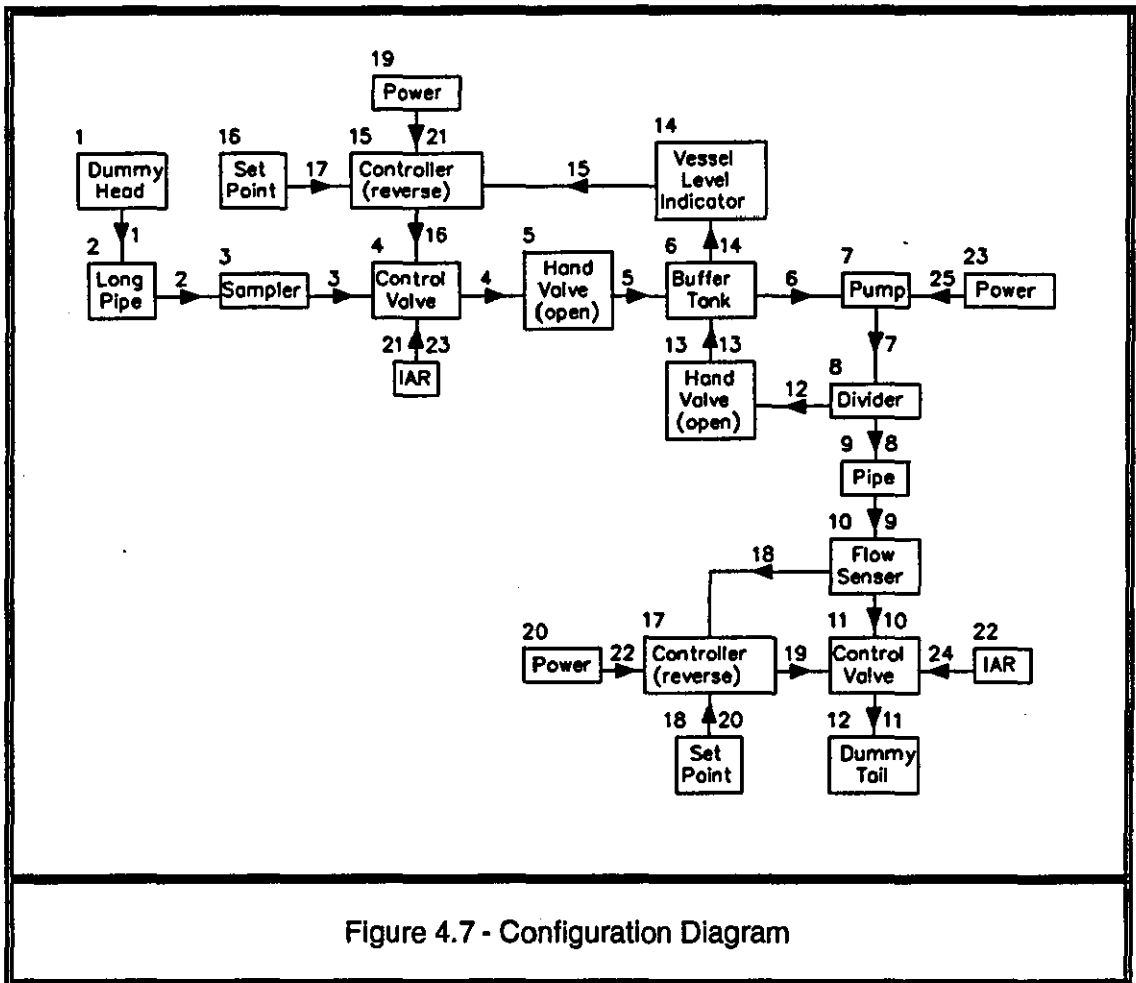
Figure 4.7 - Configuration Diagram

The top event chosen is high level in the vessel.

Figure 4.8 shows part of the fault tree synthesised for the entire configuration. The fault tree shows the level control loop (loop 1) as the primary protection system. If this loop sticks then other causes of high level are searched for. These are high flow into the vessel or low or no flow out of the vessel. Most of these can be attributed to failures in the flow control loop (loop 2).

Only the start of the control loop failure branches is given (for example branches B, C and E). Branches with a continuation line appear elsewhere in the fault tree (for example branch G). Branches ending in a variable deviation with no continuation line (for example branches A, D and F) represent variable deviations entering the plant section at the section boundary. These are diamond events. Branches ending in an event and unit number represent basic faults. Other intermediate events and variable deviations have been removed for clarity.
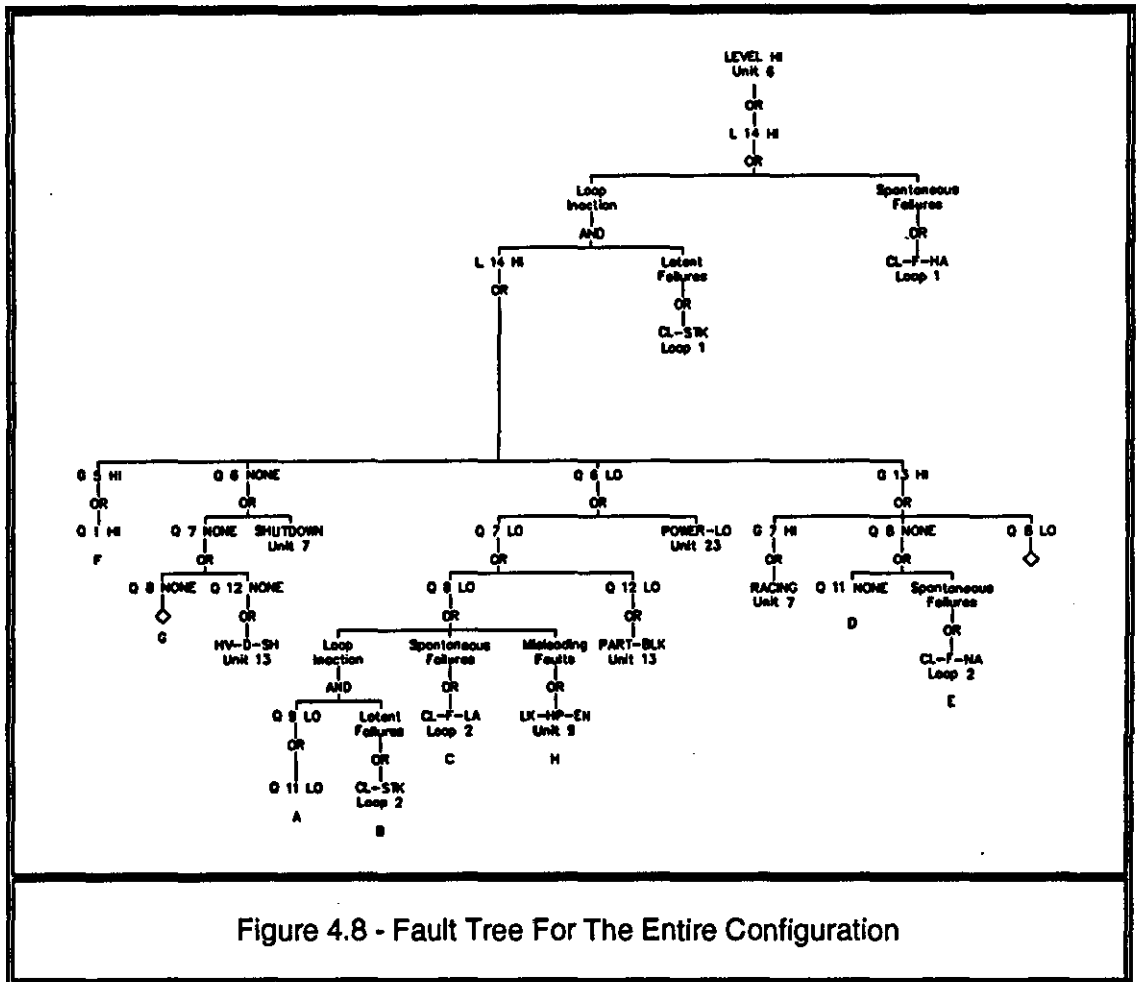
Figure 4.8 - Fault Tree For The Entire Configuration

The section of plant being studied can be reduced using the break point system. In this example units 3 and 9 have been chosen as break point units. Unit 3 is the sampler and removing this will eliminate the faults of the long length of pipe. Unit 9 is the pipe section and removing this will exclude the flow control loop from the configuration. The user is allowed to remove this control loop because it occurs entirely outside the new boundary even though connection 8 is defined as being regulated in the control loop specification.

The fault tree for the reduced configuration is shown in Figure 4.9. The branches removed are labelled as A, B, C, D and E in Figure 4.8. This eliminates all the flow control loop failures except for the 'misleading fault' in branch H.

The reason this control loop failure occurs in the fault tree is that only the units of the loop are removed, its specification and influences still remain in the configuration. The event does not cause any incorrect deletions in the fault tree even though it is a loop which has been removed from the configuration. Rather than trying to remove this occurrence from the fault tree it has been decided to leave it in. The advantage of this is that it marks in the fault tree where the protection from the flow control loop would occur.

4-21
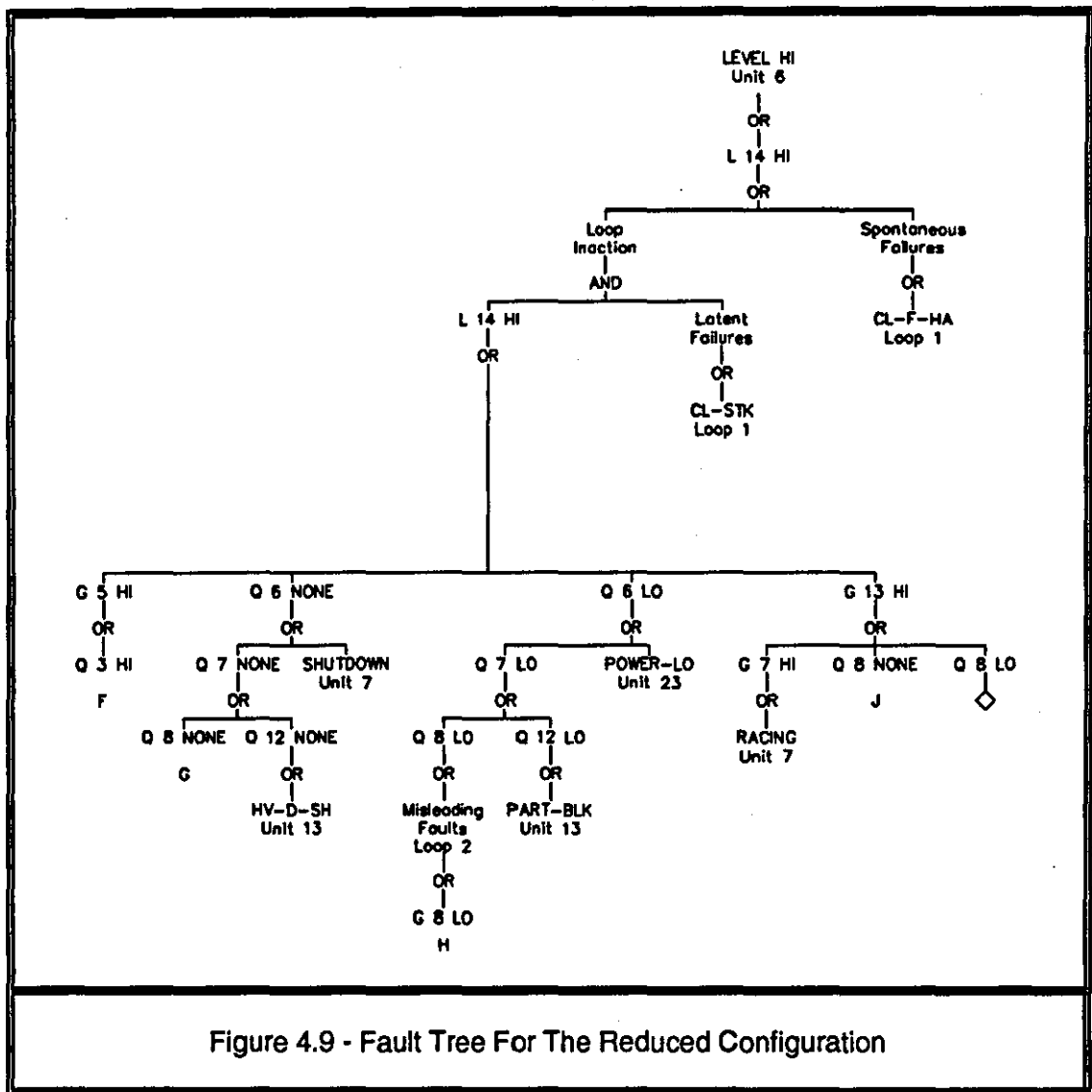
Figure 4.9 - Fault Tree For The Reduced Configuration

Since connection 3 and connection 8 are the boundary points of the configuration the propagation of flow deviations terminate at these connections. For this reason branches F, G and J on Figure 4.9 become diamond events.

The break points introduced into this example have reduced the section covered by the configuration. One of the control loops has successfully been removed from the fault tree apart from a single reference which indicates where the loop protection would occur.

4.6.7 Rules For Break Points

This is a list of rules that must be followed when using the break point system to reduce the size of the configuration being considered.

    i.   A break point unit must have only two ports, an inlet port and an outlet port.

    ii.   A break point unit that forms an entry to the configuration must be specified as a new dummy head unit.

    iii.   A break point unit that forms an exit from the configuration must be specified as a new dummy tail unit.

    iv.   All units outside of the new configuration must be specified as redundant units.

    v.   Control and trip loop signal lines must not cross configuration boundaries.

    vi.   Each new dummy head unit has its model library reference number changed to that for a dummy head unit, the outlet stream from the unit is labelled as coming from a dummy head unit, the inlet and outlet port numbers of the unit in the configuration are reversed and the inlet streams end unit number is changed to 0.

    vii.   Each new dummy tail unit has its model library reference number changed to that for a dummy tail unit, the inlet stream from the unit is labelled as going to a dummy tail unit and the outlet streams start unit number is changed to 0.

    viii.   Each redundant unit has its model library reference number changed to 0.

4.7 Summary

This chapter has given a guide for decomposing the piping and instrumentation diagram into the configuration diagram from which the data required for fault tree synthesis can be obtained. Rules are given for defining this configuration at the data input stage. A new process has been introduced to reduce the number of times a configuration has to be entered.

4.8 References

1. Lapp S. A., Powers G. L.; Computer-Aided Synthesis Of Fault Trees; IEEE Transactions On Reliability; R-26; 1977; pp 2–13.

2. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 5/1–5/36.

3. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 6/1–6/27.

4. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 7/1–7/22.

Chapter 5

Specification Of Lists And Model Libraries

5.1 Introduction

This chapter looks in more depth at the different libraries used for computer aided fault tree synthesis. These libraries contain all the variables, deviations, faults and models that can be used. Information is also held on which combinations of variables and deviations are valid. The methodology used to describe the elements of the lists is based on the work of Kelly (1) and Mullhi (2).

5.2 Variables

The variables used by the package are identified by a single upper case letter. They are:

Q   volumetric or mass flow
G   pressure gradient causing flow, used in conjunction with Q for two way propagation
T   temperature
U   temperature under reverse flow conditions
X   composition
Y   composition under reverse flow conditions
P   absolute pressure source above or below atmospheric or accumulation in gas systems
R   relief sink, used in conjunction with P for two way propagation
L   level of liquid in a vessel
S   instrument signal transmitted between units
W   set point on a logic unit

In addition to these, two new variables have been added.

M   mass
E   electric current

Variables P, T and X are used to trace deviations to a cause upstream.

Variables R, U and Y are used to trace deviations to a cause downstream. For the last two the deviation must be accompanied by reverse flow.

5.2.1 New Variables

The two new variables were introduced after problems were encountered with some of the examples.

The mass M is used to model the accumulation of material in a vessel and is associated with the vessel port. It is similar to the level variable since they are both dependent on the amount of material in the vessel but it can be used for gases. This variable was created for a problem where the flow rate of material out of a suspended buffer tank was governed by the mass of fluid in the vessel. A control valve was used to control the inlet to this tank to maintain a constant mass of fluid via a mass controller. Previously the level variable was used but was felt to be unsatisfactory.

The electric current E is used to model the variation in energy supplied to a unit. The situation which required this variable was a batch distillation process where the reboiler boil-up rate was controlled by adjusting the power supplied to the electric heater using a variable resistor. The utility units cannot be manipulated in this way and using the instrument signal variable leads to confusion.

No other variable names are used.

5.2.2 Variable Subscripts

There are three subscripts associated with the variable description. The first one is optional and applies to multiple component situations. The second gives the port number and the third defines the port type in the model where the variable occurs.

5.2.2.1 Component Subscript

This subscript is used with the composition variables X and Y to distinguish components in a multiple component system. It is a single alphabetic character between the letter 'A' and the letter 'T' and acts as an extension to the single character variable name. The subscripts should be used in order, starting with 'A', with the same subscript used for the same component for both X and Y.

If these subscripts are used the model must be defined as a multi-component model to prevent problems converting it into numerical minitrees. Multi-component minitrees can be used with single component minitrees because in a single component model all the individual components behave in the same way and the same minitree can be used.

Component subscripts are only necessary if it is impossible to model the components using only one identifier. Examples of this are streams with multiple components which undergo a selective mass transfer between phases (eg. distillation columns, partial reboilers or condensers) or the mixing of different component streams (eg. reactors).

Component subscripts are not necessary if the different component streams do not normally mix. If a fault occurs which allows mixing then the single component variable can be used to model the effects of the 'impurity' in either stream.

5.2.2.2 Port Number Subscript

This subscript refers to the port number on the unit in question where the variable occurs. The possibilities are any of the digits 1 to 9. When a unit is modelled, its ports should be numbered sequentially starting at 1.

5.2.2.3 Port Type Subscript

The final subscript refers to the port type under normal conditions, where the variable occurs. The possibilities are

IN   where flow into the unit exists
OUT   where flow out of the unit exists
SIG   where a signal enters or leaves a unit
VES   where information about the internal variables of a vessel type unit are
        modelled
UTL   where a utility, such as electrical power, enters or leaves the unit

These subscripts are used to determine which variable names and deviations can be associated with each port. The variable names which can be combined with the port types are given in Table 5.1. Those in bold indicate where the variable is defined as an output from the unit. The others show where the variable is defined as an input to the unit. Those with a '-' are not valid. Only those in bold should be used in event statements or decision tables.

The flow Q when connected to a vessel port is used to model the flow of material from one phase to another in a vessel type unit. It is of particular interest in condensers and reboilers (see Chapter 8).

The mass M is only defined at the vessel port because it is similar to the level variable and has no meaning in a pipe.

Table 5.1 - Port Type/Variable Name Combinations													
	Q	G	T	U	X	Y	P	R	L	S	W	M	E
IN	y	Y	y	Y	y	Y	y	Y	-	-	Y	-	Y
OUT	Y	y	Y	y	Y	y	Y	y	-	-	Y	-	Y
SIG	-	-	-	-	-	-	-	-	-	Y	-	-	-
VES	Y	-	Y	-	Y	-	Y	-	Y	-	-	Y	-
UTL	-	-	-	-	-	-	-	-	-	Y	-	-	-

The electric current E is used to model the variation of electrical power to a unit. This 'flow' of electric current is modelled in the same way as fluids. Therefore, electric current is defined at the same ports as the flow variable, ie. the inlet and outlet ports. This allows the variable to be used to replace, for example, a flow of steam to a reboiler and control loops to be used to model its control system.

Typical variable descriptions are R1IN, Q2OUT, Y3IN, XA2OUT, L4VES and S5UTL.

5.3 Deviations

A deviation is applied to a variable and represents a change from the normal of that variable. The normal is not necessarily constant, for example the set point of the slave control loop in a cascade loop system changes all the time, but its value represents the normal for that instant. The deviations used are

HI   the value of the variable is higher than normal
LO   the value of the variable is lower than normal
NONE   the variable has a negligible value
SOME   the variable has a finite positive value
REV   the variable has a negative or reversed value
NCHA   the signal has not changed, when a change is expected
SHAC   the trip should activate
NOP   there is no pressure source downstream
NOR   there is no sink for pressure relief upstream

For the HI and LO deviations no attempt is made to distinguish different degrees of severity, eg. high and very high. The deviation is considered sufficient to cause the top event to happen.

When combined with variables, some deviations do not have obvious meanings.

For signal S, NONE and SOME can be interpreted in two ways, either as the unit functioning correctly as required, or by the unit being faulty or operating without need.

For pressure P, REV means a large pressure loss. It implies the unit is connected via a large aperture to a lower pressure system, usually to atmosphere, and thus has a more drastic effect than LO.

For relief R, REV means a pressure gain, or that the unit is connected via a large aperture to a higher pressure system and thus has a more drastic effect than HI.

The deviation NCHA is used only with signal S. It is associated with control loops which have a functional failure attributed to the sensor, controller or valve.

The deviation SHAC is used only with a signal on a trip system to indicate its functional failure.

Table 5.2 summarises which deviation can be used with which variable. The 'Pv' variable name refers to a pressure with the port type vessel.

Table 5.2 - Deviation/Variable Name Combinations													
	Q	G	T	U	X	Y	P	R	L	Pv	S	W	M	E
HI	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
LO	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
NONE	Y	Y	-	-	-	-	Y	Y	Y	Y	Y	-	Y	Y
SOME	Y	Y	-	-	-	-	Y	Y	-	-	Y	-	-	Y
REV	Y	Y	-	-	-	-	Y	Y	-	-	-	-	-	-
NCHA	-	-	-	-	-	-	-	-	-	-	Y	-	-	-
SHAC	-	-	-	-	-	-	-	-	-	-	Y	-	-	-
NOP	-	-	-	-	-	-	-	Y	-	-	-	-	-	-
NOR	-	-	-	-	-	-	Y	-	-	Y	-	-	-	-

5.4 Fault And Event Library List

The fault list is a compendium of all the initiating and intermediate events that can be used in the unit models or during synthesis of the fault trees. It is not an exhaustive list and new ones can be added at any time.

Initiating events are basic faults that occur in previously healthy units which result in the propagation of a fault deviation out of the unit. Examples of these are a partial blockage and a control valve failing stuck.

Intermediate events are events which link up parts of the fault tree but are not variable deviations. Examples are control loop failing high and the dummy events.

The events are limited to eight characters in length and their position in the fault array gives the index number for the fault. The length limitation means that the fault name has to be abbreviated. Kelly (3) developed a list of standard abbreviations and this list has been updated to form Table 5.3. Using these abbreviations the user should be able to generate most of the faults associated with common equipment.

Table 5.3A - Equipment, etc.			
HV-	hand valve	BD-	bursting disk
CV-	control valve	BLK-	blanked flange
RCV-	remote control valve	CNT-	controller
TV-	trip valve	TSW-	trip switch
CTV-	control/trip valve	ACT-	activator
SV-	solenoid valve	OP-	operator
ESV-	emergency shutdown valve	COM-	computer
RV-	relief valve	SEL-	selector
VV-	vent valve	SW-	switch
3WV-	three way valve	REL-	relay
NRV-	non-return valve	SEN-	sensor
SMP-	sampler valve	ALM-	alarm
PRV-	pressure reducing valve	CL-	control loop
GOV-	governor valve	TL-	trip loop

Table 5.3B - Utilities, etc.			
SIG-	instrument signal line	ING-	inert gas supply
SET-P-	set point	NIT-	nitrogen supply
UTIL-	utility supply	CWT-	cooling water supply
POW-	power supply	PWT-	process water supply
STM-	steam supply	VAC-	vacuum pressure
IAR-	instrument air supply	FWT-	fire water supply
PAR-	process air supply	CO2-	carbon dioxide supply

Table 5.3C - Operations, etc.			
-D-	directed	-OR-	operational
-F-	fails	-FN-	functional
-FT-	fails to	-AT-	at

Table 5.3D - Conditions, etc.			
-SH	shut	-ON	on
-OP	open	-OF	off
-STK	stuck	-MAN	on manual
-HA	with high aperture	-DIS	disabled
-LA	with low aperture	-PB	partial blockage
-NA	with no aperture	-CB	complete blockage
-HI	high	-LOSS	loss
-LO	low	-F	fails
-BROKN	broken	-OK	working as normal

Table 5.4 is a brief list of the more common faults and the intermediate event names used. The intermediate events are those assigned by the programs during the synthesis of the fault tree.

Table 5.4 - Common Faults And Intermediate Events			
LK-LP-EN	leak to a low pressure environment	DUMMY n a (DUMMY)	intermediate dummy events, numbers and letters
LK-HP-EN	leak from a high pressure environment	INTERNAL	causes are internal to a DHC (divider-header combination)
INT-LK	internal leak	EXTERNAL	causes are external to a DHC
PART-BLK	partial blockage	LEAKAGE	leakage faults in a DHC
COMP-BLK	complete blockage	BLOCKAGE	blockage faults in a DHC
EXT-HEAT	external heat source	NO-FLOW	no flow causes in a DHC
EXT-COLD	external cold source	LO-FLOW	low flow causes in a DHC
NORMAL	normal event	HI-FLOW	high flow causes in a DHC
IMPOSS	impossible event	-PRESS	pressure causes in a DHC
SEQ-F-AT	sequence fails at	COMPLOOP	compensating loop events
SEQ-F-AF	sequence fails after	OTH-CAUS	other causes

The complete list of basic faults and intermediate events is given in Appendix B.1. This is a revised list with the elements grouped together.

5.5 Basic Fault Library List For Automatic Generation

A facility has been developed by Mullhi which allows some of the basic faults to be generated automatically. The algorithm is incorporated into the unit model generation program MODGEN as part of the unit model input feature. Initially this was set up for the six basic faults associated with pipe type units. These are

    F LK-LP-EN
    F LK-HP-EN
    F PART-BLK
    F COMP-BLK
    F EXT-HEAT
    F EXT-COLD

The full names are given in Table 5.4.

The process allows a user to generate the event statements for the fault by simply specifying the ports affected. A file contains a list of faults which can be generated automatically and for each fault another contains the variable name, port type and list of deviations that can be coupled together to form the effects of the fault. A condensed form of one of the files showing only the variables with deviations is

```
F LK-LP-EN
Q    OUT    LO     NONE   REV    ----
G    IN     HI     SOME   ----   ----
P    OUT    LO     NONE   REV    ----
R    IN     HI     SOME   NOP    ----
L    VES    LO     NONE   ----   ----
P    VES    LO     ----   ----   ----
M    VES    LO     NONE   ----   ----
```

This file has been modified to incorporate the new variable for mass. The LK-LP-EN fault is the only fault from the initial list to affect the variable. The electric current variable is not affected by any of the faults.

The method by which the program generates the event statements can be illustrated by using an example. If the event statement for a leak to a low pressure environment is to be generated for inlet port 1 the program carries out the following tasks. It scans the propagation equations to determine which variable descriptions are defined in the model. This is essentially the left hand side of the equations and should normally yield 'G1IN', 'U1IN', 'Y1IN' and 'R1IN'. Considering each row in turn it tries to match the variable description with the first two columns. If a match is found the remaining columns give any deviation which is caused by the fault. The event statement generated is

F LK-LP-EN: G1IN HI, G1IN SOME, R1IN HI, R1IN SOME, R1IN NOP

5.5.1 Inclusion Of States And A New Fault

Four new files have been added to the existing ones.

The introduction of a vessel model generation routine (Chapter 7) has resulted in the need for the NORMAL and IMPOSS state event statements to be generated automatically. Although not a fault the normal state is required to deal with some of the variable deviations started by the templates applied by the synthesis package. This event statement prevents variable deviations which would normally occur from being labelled as impossible. The

impossible state is required to prevent certain deviations from propagating through the units. Examples are reverse flow and no relief into a vessel. A condensed form of the files showing only the variables with deviations is

```
S NORMAL
Q    OUT    SOME    ----    ----    ----
G    IN     SOME    ----    ----    ----
P    OUT    SOME    REV     ----    ----
R    IN     SOME    NOP     ----    ----


S IMPOSS
P    OUT    NOR     ----    ----    ----
R    IN     NONE    REV     ----    ----
```

The other two files are used by the heat exchanger model generation routine (Chapter 8). This generates the internal leak INT-LK fault for both sides of a heat exchanger. Clearly with fluid passing from one side to another the effects will be different for each side. One file is used to model the higher pressure side and the other for the lower pressure side. The two files are

```
F INT-LK       HIGH PRESSURE SIDE
Q    OUT    LO      NONE    REV     ----
G    IN     HI      ----    ----    ----
P    OUT    LO      NONE    REV     ----
R    IN     HI      ----    ----    ----
L    VES    LO      ----    ----    ----
P    VES    LO      ----    ----    ----
M    VES    LO      ----    ----    ----


F INT-LK       LOW PRESSURE SIDE
Q    OUT    HI      ----    ----    ----
G    IN     LO      NONE    REV     ----
P    OUT    HI      ----    ----    ----
R    IN     LO      NONE    REV     ----
L    VES    HI      ----    ----    ----
P    VES    HI      ----    ----    ----
M    VES    HI      ----    ----    ----
```

It is planned to expand this facility to cover other faults which can be grouped together. To do this it is necessary to determine links between types of port for fault characteristics.

5.6 Unit Model Library List

A basic unit model library was developed by Kelly (4) with the first version of the synthesis package. This consisted of a list of process units, instrument units and special units required for modelling. The list was not very comprehensive and each model was only developed when it was required for a particular situation. Once developed it remained in the library.

The content of the library was built up by a process of adding new units as and when required. This resulted in an erratic development and inconsistent numbering system. For a user to find a model in the library requires him to search through the entire list each time.

Using the current library as a base a new taxonomy of units has been devised to give a more representative library. This includes a new numbering system which groups types of units together and should a new model be required also allows these to be added to the group.

The current unit model library list is given in Appendix B.2.

A group by group list of the library appears in the following pages with a brief description of particular features of the models.

5.6.1 Pipe And Pipe Type Units

Short Length of Pipe
Long Length of Pipe

Divider, 2 Outputs   )
Header, 2 Inputs

These units are propagation models used to join units or streams together.

The short length of pipe contains the normal blockage, leak and external temperature faults. This is primarily for use as a connection between pieces of equipment. This model is the base for all models with flow.

The long length of pipe is used to model sections of pipe where it is possible to get surges of flow and water-hammer effects. The model has the basic pipe type faults and the water-hammer fault.

The divider provides a method of splitting a flow stream into multiple streams. For three or more streams the dividers must be nested because the new methodology allows only binary splitting.

The header unit provides a method of combining two streams into one. The unit does not have any spare capacity so if one inlet stream has high flow then the other must have low flow. They can be nested to combine multiple streams.

5.6.2 Pipe End Units

Dummy Head
Dummy Tail

Process Air Supply
Process Steam Supply
Process Water Supply
Cooling Water Supply
Fire Water Supply
Process Nitrogen Supply
Inert Gas Supply
Fuel Gas Supply

Drain
Vent
Flare Relief

These are the start or end models used in the configuration diagram.

The dummy units are used to mark the boundaries of the plant section being studied. They allow initiation and termination of all combinations of pipeline variables and deviations. They do not initiate basic faults but when encountered result in the fault tree branch being traced ending in a diamond event.

The supply units are not utility units because they act similarly to the dummy head but they have initiating faults for supply failures. The models contain flow and pressure equations and the port types allow them to be connected to valves and pipes.

The drain, vent and flare relief are all dummy tail type units but have been given specific names and their own initiating faults.

5.6.3 Valve Type Units

Hand Valve   - Open
            - Closed
Solenoid Valve   - Open
                 - Closed
Remotely Controlled Valve   - Open
                             - Closed

Pressure Relief Valve
Non-Return Valve
Excess Flow Valve
Pressure Reducing Valve
Governor Valve

Control Valve, Fails Open
Control Valve, Fails Closed
3-Way Control Valve, Fails Open
3-Way Control Valve, Fails Closed

Trip Valve, Fails Open   - Open
                          - Closed
Trip Valve, Fails Closed   - Open
                            - Closed

Closed Emergency Shutdown Valve, Fails Open
Open Emergency Shutdown Valve, Fails Closed
Vent Valve   - Open
            - Closed

Manual Vent Valve   - Open
                    - Closed

Control/Trip Valve, Operating   - Fails Open
                                 - Fails Closed
Control/Trip Valve, Not Operating   - Open
                                     - Closed

Variable Resistor

This group contains all the valve models. There is no distinction made between pneumatic and electrical valves. The faults of the two types are similar apart from the signal blockages in the air pipes. To convert to a pneumatic unit, the instrument air pipe model should be used in conjunction with the valve.

Hand valves are the simplest valves and they can be open or closed. Their failures result in the valve being closed or opened respectively.

The solenoid valve is a simple electrically operated valve. Failure of utility results in the valve position not changing.

The remotely operated or automatic valves are similar to hand valves but they are operated from the control room. They can be either open or closed and should the utility supply be lost they will remain in that position.

The relief valve has a built-in set point unit and the faults are the set point being too low or too high. It can also fail to open or fail open and can be blocked or be undersized.

The non-return valve can fail open allowing reverse flow to occur.

The excess flow valve closes should it sense a higher flow than there should be. The valve can fail to operate.

The pressure reducing valve and governor valve fault is its failure to prevent the propagation of high flow or pressure.

The control valves are modulating and their position depends on the signal from the controller. If the utility supply is lost they can fail open or closed.

The 3-way valves are special valves which have three connections for flow, one inlet port and two outlet ports. The valve operates in a similar manner to the control valve except that the excess flow is diverted through the second outlet port. This second outlet forms a by-pass around the unit through which the flow is being controlled. It can fail stuck or fully open/closed to give full flow through the next unit or by-pass system.

The trip valves and the emergency shutdown valves are similar in operation and failure but the signal is received from the trip switch and control room respectively. They can fail stuck or operate unnecessarily.

The vent valve allows a control signal to pass unaltered until a signal is received from a trip switch. The control signal is then vented and prevented from reaching the control/trip valve. There are two versions depending on the required logic.

The manual vent valve is similar to the vent valve but is opened or closed by the operator.

The control/trip valves are control valves which slam shut or fully open on a signal from a trip switch. They should be used with the vent valves.

The variable resistor is specifically for use with the electric current variable. It is modelled in the same way as a control valve receiving a signal from a controller and manipulating the 'flow' of electric current by varying the resistance. It replaces the control valve when the manipulated variable is electric current. The resistor fault is failure to change when directed.

5.6.4 Pipe System Units

Local Sensor
Sample Point
Flushing Link
Drain Point
In-line Filter
Steam Trap

Blank Flange

Bursting Disc

This group contains small units or systems found on pipelines. They are situated between pipe sections but are not required to be connected to any other model.

The local sensor is a sensor unit that has been disconnected or is used for local observations. It does not transmit any signals. The fault is leakage through the gauge.

The sample point is a single unit to replace the divider, closed hand valve and dummy tail set of units. It is used where a sample point appears in the pipeline. The faults are leakage and valve open.

The flushing link is a single unit to replace a header, a closed hand valve and a dummy head set of units. These are found mainly around pumps and are the connections which would be linked to the pump flushing system on a plant. The fault is leakage from the system.

The drain point is the same as the sampler point but is used as the drain for the flushing system or simply as a pipeline drainage point.

The in-line filter is a unit sited in the pipeline which is used to catch any debris that may be in the fluid. The faults are blockage and failure to stop the debris.

The steam trap collects and prevents condensate in a steam line from continuing along the pipe. The faults are blockages caused by debris caught by the built-in filter, leaks and failure to prevent the condensate passing.

The blank flange is often found blocking off a pipeline and is more secure than a closed hand valve. The failures are the flange breaking or being removed.

The bursting disc is the same as the relief valve with a set point built in and similar failures.

5.6.5 Sensor Units

       Multi-purpose Pipe Mounted Sensor
       Multi-purpose Dead Leg Sensor
       Multi-purpose Vessel Mounted Sensor
       Potentiometer on Valve Stem

This group contains all the sensor models. No distinction is made between electronic and pneumatic types.

The faults deal with the sensor failing with a high, low or no output signal.

The pipe mounted sensor converts the output deviation of the flow, pressure, temperature and composition variables from port 2 of the unit to signal deviations from ports 3, 4, 5 and 6 respectively. The port to which the signal link is connected determines the variable which the sensor records. The seventh port is for the utility.

The dead leg sensor converts the input deviation of the pressure, temperature and composition variables from port 1 on the unit to signal deviations from ports 2, 3, and 4 respectively. The port to which the signal link is connected determines the variable which the sensor records. There is no flow through the unit. The fifth port is for the utility.

The vessel mounted sensor is placed on the vessel port of a vessel and it converts the vessel variables of level, pressure, temperature, composition and mass to output signals from ports 2, 3, 4, 5 and 6 respectively. The port to which the signal link is connected determines the variable which the sensor records. The seventh port is for the utility.

The potentiometer is mounted on a valve stem and gives the position of the stem.

5.6.6 Controller And Switch Units

Regular Controller   - Direct Acting
                                   - Reverse Acting

Cascade Controller   - Direct/Direct
                                   - Direct/Reverse
                                   - Reverse/Direct
                                   - Reverse/Reverse

Double Output Controller - Direct/Reverse

Controller on Manual

1-out-of-1 Trip Switch   - High
                                     - Low
                                     - None

1-out-of-2 Trip Switch   - High
                                     - Low
                                     - None

1-out-of-3 Trip Switch   - High
                                     - Low
                                     - None

2-out-of-3 Trip Switch   - High
                                               - Low
                                             - None

Flow Switch
Pressure Switch
Temperature Switch
Level Switch
Limit Switch for Valve

This group contains the decision-making units.

It is assumed that the controller and trip switch are electronic decision-making units. Other units in the system will determine if the operation is partly pneumatic or all electrical. They have their own utility connection.

The direct acting controller transfers the signal from the sensor directly to the valve.

The reverse acting controller reverses the sensor signal before sending it to the valve, ie. a high sensor reading produces a low valve signal.

In a cascade controller, the master loop receives a signal from one sensor and takes either direct or reverse action, this is then used as the set point for the slave loop which receives a signal from the other sensor. The output to the valve is either direct or reversed.

The double output controller is where one input signal is transferred to two valves one direct and one reversed. In the case where the outputs are transferred in the same mode, ie. direct or reversed, the regular controller can be used with the output signal split by a signal splitter.

The controller on manual only allows a control signal to pass to the control valve if the controller has been set back to automatic in error.

A very large number of combinations of trip switch are possible so only the lower order ones have been given. The user will be able to compare the logic used in these to determine the correct way to formulate other combinations. They send a signal to a trip valve when the appropriate deviation is received.

The variable switches are similar to the trip switches but they do not involve a separate sensor unit.

5.6.7 Other Instrument Units

Electric Cable
Instrument Air Pipe
I/P Transducer
P/I Transducer

Signal Splitter, 2 Outputs
Vessel Port Splitter, 2 Outputs
Signal Header, 2 Inputs

High Signal Selector, 2 Inputs
Low Signal Selector, 2 Inputs

Set Point Unit

Process Operator
Process Control Computer

High Alarm
Low Alarm

This group is used for electrical, pneumatic and signal type units.

The electric cable model contains faults particular to wires and cables.

The instrument air pipe is for use on pneumatic control and trip loops. It is used to convert an electrical model loop to a pneumatic model loop. It has failures of partial and complete blockage of the signal transmitted through it.

The transducers convert an electrical signal to a pneumatic signal or vice versa using the instrument air utility.

The splitter models are methods of producing two outputs of the same deviation and type as the input. One divides the signal deviation and the other divides the vessel variable deviation.

The signal header produces the same output signal as either of the input signals.

The selectors compare two input signals to produce one output. One produces the higher deviation to the output and the other gives the lower.

The set point unit is used on the controllers and trip switches and initiates the high and low set point faults during loop failure.

The process operator unit is used to model the responses of humans. The input or inputs is from the alarm unit and the output goes to a remotely controlled valve.

The process control computer is used to model the responses of computers. It is similar to the operator model.

The alarm units are linked to sensors and operate when the signal is high or low. The faults deal with failures to operate or operation when there is no alarm.

5.6.8 Pump And Compressor Units

```
Reciprocating Compressor   - Running
                           - Stopped
Centrifugal Compressor    - Running
                          - Stopped

Reciprocating Pump   - Running
                     - Stopped
Centrifugal Pump    - Running
                    - Stopped
Centrifugal Pump With Water Seal   - Running
                                   - Stopped
```

The pump group gives the units used to mechanically induce a flow or pressure increase in the stream.

Reciprocating pumps and compressors do not allow reverse flow through them, even when stopped. Rotary pumps can be modelled as reciprocating pumps.

Centrifugal pumps and compressors include axial systems and allow reverse flow through them. The centrifugal pump will be the most common type required.

The water seal pump is a special case and has been developed to account for the faults that occur should the high pressure or low pressure seals become faulty.

The running and stopped versions have been given so that complete pump banks can be modelled rather than having dormant pump lines blanked off with dummy tails and heads.

5.6.9 Heat Exchanger And Heater Units

Single Phase   - Shell and Tube, Hot Fluid in Shell
- Shell and Tube, Hot Fluid in Tubes
- Frame and Plate, Hot Fluid in Side A
- Frame and Plate, Hot Fluid in Side B
- Double Pipe, Hot Fluid Inside
- Double Pipe, Hot Fluid Outside
- Finned Tube, Air

Condenser   - Shell and Tube, Partially Condensing Vapour in Shell
- Shell and Tube, Condensing Vapour in Shell
- Shell and Tube, Condensing Vapour in Tubes
- Frame and Plate, Condensing Vapour in Side A
- Frame and Plate, Condensing Vapour in Side B
- Double Pipe, Condensing Vapour Inside
- Double Pipe, Condensing Vapour Outside
- Finned Tube, Air

Reboiler   - Shell and Tube, Partially Boiling Liquid in Shell
- Shell and Tube, Boiling Liquid in Shell
- Shell and Tube, Boiling Liquid in Tubes
- Frame and Plate, Boiling liquid in Side A
- Frame and Plate, Boiling Liquid in Side B
- Double Pipe, Boiling Liquid Inside

Pipe heater
Boiler
Vaporiser
Furnace
Burner

This group is used for basic heat transfer systems where the thermal properties of a stream are changed.

There is no distinction made between heat exchangers containing gases/vapours or liquids that do not change phase because the fault properties and propagation equations are the same.

Due to modelling restrictions on the number of ports, it is not possible to model phase changes on both sides of the heat exchanger.

Four types of heat exchanger are given as illustration. The shell and tube and the finned tube types can only be blocked on one side. The frame and plate and double pipe types can be blocked on both sides. This is the only characteristic which separates the type of heat exchanger otherwise their propagation and fault characteristics are the same.

The pipe heater is a simple pipe type unit with a utility supply, steam or electricity, that heats the contents of the pipe. Should the utility fall below normal rates or fail the exit temperatures will be low.

The boiler is an extension to the pipe heater. It is a vessel with a utility supply to boil the liquid. The failures are low exit temperature and low flow out.

The vaporiser and furnace are similar to the boiler except they have a process supply link for the heating medium as opposed to a utility connection.

The burner unit can be used for a more detailed study of a furnace unit or can be placed on a relief system to simulate a flare or vent stack. The failures are poor burning or no flame which allows material to continue through the unit. This could then be detected in a top event as a cause of flammable material being present.

5.6.10 Utility Units

Instrument Air Supply
Electrical Power Supply
Steam Power Supply

These are the utility units used to power or operate other units. The utility list has been revised and these are the only units grouped in this category. The process supply units fall into the pipeline end units category. These units are connected to other units via the 'ULT' port type.

5.6.11 Containment Vessel Units

Template Models - Open Liquid Containment Vessel
- Closed Liquid Containment Vessel
- Closed Pressurised Gas Containment Vessel
- Closed Vacuum Gas Containment Vessel
- Closed Pressurised Gas and Liquid Containment Vessel

Gas Storage Vessel - Closed, Pressurised
- Gas Holder, Floating Roof

Liquid Storage Vessel, Atmospheric - Open
- Fixed Roof
- Floating Roof

Liquid Storage Vessel, Pressurised

Containment vessels are those where fluids can accumulate and represent storage or buffer vessels.

The template models are multi-purpose units whose characteristics can be varied according to the ports that are connected. They can be used to develop the users' own specific model or can be used as they appear (see Chapter 7).

The storage vessel is like a pipeline end unit because it either has inputs or outputs only. A large capacity means there will always be capacity for more flow in or out.

5.6.12 Solids Handling Units

Storage Silo
Containment Hopper
Power Cylinder
Screw Feeder and Motor
Feeder Belt and Motor
Conveyor Belt
Grinding Mill

This equipment is specifically for solids handling units.

The storage silo is a large capacity solids handling vessel. There are no inputs and an empty vessel is considered an impossible condition. The faults are blockages and solid hold-up.

The containment hopper is a smaller version of the storage hopper and can be used for intermediate storage. It is different from the storage hopper because it has input ports and output ports and has the additional fault of being empty.

The power cylinder is a valve used to maintain the flow of solids from a hopper. A utility supply is connected. The valve can be blocked or become stuck.

The screw feeder transfers the solids from the hopper. It is powered by a motor with a utility supply. Failures can be caused by loss of utility, blockages and overload on motor.

The feeder belt transfers the solids from the hopper to the main transportation device. It is powered by a motor connected to a utility supply. Failures can be caused by loss of utility or the gears system.

The conveyor belt transfers the solids over large distances. A utility supply powers the motor. The failures are a broken or jammed belt in addition to the feeder belt failures.

The grinding mill crushes the solid to a particular size of granule. It can fail due to overload or impediment by large objects.

5.6.13 Reaction Vessel Units

Tubular Reaction Vessel   - Gas Phase
        - Liquid Phase

Continuous Stirred Tank Reaction Vessel

Packed Bed Reaction Vessel   - Gas Phase
              - Liquid Phase
Fluidised Bed Reaction Vessel   - Gas Phase
              - Liquid Phase
Catalytic Gauze Reaction Vessel   - Gas Phase

Semi-Batch Reaction Vessel   - Filling
           - Reacting
           - Emptying

Batch Reaction Vessel   - Filling
         - Reacting
         - Emptying

This group contains the basic reaction vessel types. More research needs to be carried out into the effects of the stoichiometry of the reaction and the thermal properties.

The tubular reaction vessels are continuous and are similar to shell and tube heat exchangers. The reactants are pre-mixed and fed into the tubes where they are immediately brought up to the reaction temperature. The fluid on the shell side serves as a heat transfer medium.

The continuous stirred tank reaction vessel has two reactant input streams. The reactants mix immediately on entry into the vessel. There is a single product outlet and ports for heating/cooling coils.

The packed bed or catalytic reaction vessels have two inlets, which mix together, and an outlet stream. Failures can be blockages caused by foreign bodies or collapse, or changes in reaction rate caused by temperature changes or poisoned catalyst/reactants.

The fluidised bed reaction vessels are similar to the packed bed reaction vessels but there is only one inlet and outlet stream, ie. the reactants are pre-mixed. Failures are blockages caused by foreign bodies, collapse of the packing caused by low inlet flow, changes in reaction rate caused by temperature changes or poisoned catalyst.

The catalytic gauze reaction vessel has two inlets and an outlet stream. The failures are blockages and poisoned catalyst.

The semi-batch reaction vessel model has to be considered in three stages. The first stage is filling with the first reactant. The second stage is continuously adding the second reactant while taking off a product stream. Over the period considered the concentration of the second reactant remains constant giving a constant reaction rate and temperature under steady state. The final stage is where the vessel contents are emptied out. There is also an inlet and outlet port for the heating/cooling coils.

The batch reaction vessel model system is similar to the semi-batch model but both reactants are added in the first stage. They are then brought up to the reaction temperature and pressure to initiate the reaction. The contents are then discharged in the final stage.

5.6.14 Mixing Vessel Units

Binary Mixer, 2 Inputs

The binary mixer combines two streams of different components into a single outlet stream. The two inlet flow legs do not affect each other which is the case with the header unit. The two components do not have to be contained in the same solvent as long as the two solvents are completely miscible.

5.6.15 Separation Vessel Units

Gas-Gas Separation - Absorption Column, Packed
                    - Absorption Column, Plate
                    - Adsorption Column

Gas-Liquid Separation - Wet Scrubber

Liquid-Gas Separation - Steam Stripper

Liquid-Liquid Separation   - Steam Stripper
                              - Distillation Column, Superheated Vapour Feed
                              - Distillation Column, Saturated Vapour Feed
                              - Distillation Column, Boiling Liquid Feed
                              - Distillation Column, Sub-cooled Liquid Feed
                              - Rectifying Column
                              - Stripping Column
                              - Decanting Vessel
                              - Ion Exchanger
                              - Flash Drum

Gas-Solid Separation   - Cyclone
                        - Filter
                        - Wet Scrubber
                        - Electrostatic Precipitator

Liquid-Solid Separation   - Thickener/Clarifier
                             - Hydrocyclone
                             - Centrifuge
                             - Filter

Solid-Liquid Separation   - Rotary Dryer
                            - Crystalising Evaporator
                            - Concentrating Evaporator

Solid-Solid Separation   - Cyclone

The packed absorption column has two inlet and two outlet streams which are free to mix. There is some correlation between the pairs of streams. The failures are blockages due to foreign bodies and disintegration of the packing. There may also be carry-over of impurity due to excessive flow rates.

The plate absorption column is similar to the packed column but only contains blockage failures due to foreign bodies.

The adsorption column has a single inlet and outlet stream but is packed with adsorbent. The failures are blockage and inefficiency due to saturated packing.

The scrubbing column has two inlets and two outlet streams which can be considered to be independent. Only the composition of the pairs changes as the impurities move from the gas to the liquid.

The steam stripper has two inlets and two outlets. The streams can be considered to be independent because only the compositions change as the more volatile component moves to the steam stream. Failures are inefficiency and saturation.

The distillation columns have a single feed inlet, two return inlets and two product outlet ports. They are assumed to be tray columns and have failures of blockage and inefficiency. They are categorised according to the state of the feed.

The rectifying column and stripping column have a feed input and two outputs. One output stream is for the desired product and the other is for the residue. Failures are the same as for the distillation column.

The decanting vessel has a single inlet and two outlet ports. The failure is inefficiency caused by high inlet flow.

The ion exchange vessel has two inlets and two outlets one of each being dead legs under normal operation. The failure is saturation of the ion exchange beads.

The flash drum has a single inlet and two outlet ports, one for liquid and one for vapour. It has faults due to leaks.

The cyclone/hydrocyclone has an inlet and two outlets. High inlet flow causes a deviation in performance.

The filter has an inlet, two outlets and a utility port. One outlet can be linked to the inlet and the other is for the removal of any solids collected. The failure is blockage caused by too much solid.

The electrostatic precipitator has an inlet and an outlet. Since it is used to separate small quantities of material which take a long period of time to build up it is thought unnecessary to have an outlet for the solids. There is a utility port for the electricity and the failure is the loss of this utility.

The thickener/clarifier has an inlet and two outlets. Failure is inefficiency due to high flow.

The centrifuge has an inlet and two outlets and a utility supply to produce the spinning motion required. High inlet flow causes inefficiency.

The rotary dryer has an inlet and two outlets, one for the dried stream and one for the moisture driven off. There is also a utility port and the failure is the loss of this.

The evaporators have an inlet, two outlets and a utility port for the evaporating media. Failure is loss of utility.

5.6.16 Note

It should be noted that a particular condition of a unit may be either a fault or a functional state. For example, a controller may be set to manual control in error, in which case this is a fault condition, or intentionally, in which case it is in a desired state. For each situation a different model is required, respectively, a regular controller and a controller on manual. Other examples are a running pump and a stopped pump.

5.7 Top Event Model Library List

The top event models are kept separate from the configuration data because then it is very easy to use different fault tree top events for the same configuration. The models consist of a named top event which has a set of event statements and/or decision tables describing its causes. The model usually has just one level although it is possible to have more.

5.7.1 Top Event Models For Vessels

There was a problem developing within the top event model library which could have resulted in the library becoming full. The problem was due to the wide variety of vessel models studied and the different variables looked at.

Top event models for pipe type units have port 1 as the inlet and port 2 as the outlet. This poses no problems because the top event can be applied to any unit which has port 1 as the inlet and port 2 as the outlet such as sensors and valves.

However, top events for vessels are different because these are linked directly to the vessel ports. This means a new model has to be produced for each vessel which has a different vessel port number (the vessel port number is determined by the number of inlet ports plus the number of outlet ports plus one). A brief listing of some of the top event names highlights this point.

UNDRTEMP	IMP B HI	OVERTEMP	PRESTEMP	EMPTY 3
OVERPRES	UNDRPRES	OVRFLOW3	IMP A HI	FLOW LO
FLOW HI	SEQ-ABRT	LEV-LO-6	LEV-HI-6	EMPTY 6
TANK-RUP	RUP-TANK	LEV-LO-8	LEV-LO-5	LEV-HI-4
LEV-HI-3	LEV-LO-3	LOPRES-8	HIPRES-8	HIPRES-5
SPHERRUP	VES-RUPT	REV FLOW	NO FLOW	SOME FLO

Those names with a number in them refer to top events specific to vessel models and in particular to vessel models with the vessel port number the same as the number in the name. This is because the top event is linked to a variable deviation which has the port number contained in it. An example of a typical top event for a vessel model is shown in Figure 5.1. The Vessel port is port 3, and there is a single inlet port, port 1 and a single outlet port, port 2. The vessel can be emptied by no flow in AND some flow out.

To prevent the top event model library from becoming full and without the need to delete models no longer in use a way was needed of making the models for vessels independent.

```
                    EMPTY 3
                       │
                      AND
                       │
         ┌─────────────┴─────────────┐
      G1IN NONE                  Q2OUT SOME
```

Figure 5.1 - Old Top Event For Empty Vessel
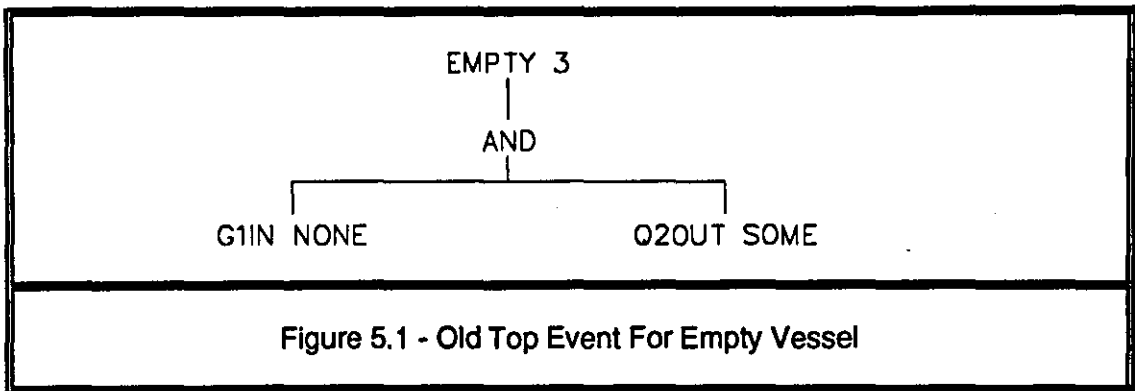
The proposed solution uses context or port independent models. The idea is to create a library of template models for the top event which contain no port numbers. These contain just an event statement linking the event to the variable index, port type, variable deviation and a blank where the port number normally is.

An example of this type of top event is given in Figure 5.2.

```
            ┌─── ── ── ┐── Top Event
            │   EMPTY    │   Model
            │     │      │
            │    OR      │
            │     │      │── Unit Model
      ┌─ ── ┴─ ── ┴ ── ── ┴─ ┴─ ─┐
      │     │  LnVES NONE  │      │
      │     └─── ── ┬─ ── ┘      │
      │           AND            │
      │            │             │
      │     ┌──────┴──────┐      │
      │  G1IN NONE    Q2OUT SOME │
      └─ ── ── ── ── ── ── ── ── ┘
```

Figure 5.2 - New Top Event Model For Empty Vessel
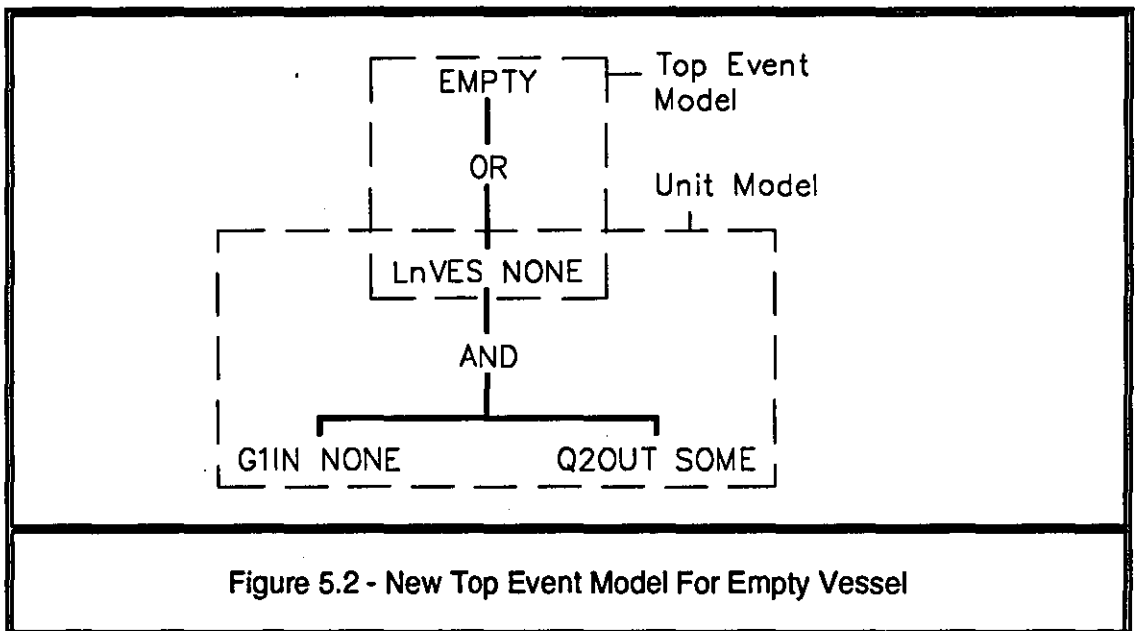
It can be seen that the main difference between this and the one in Figure 5.1 is that the inlet and outlet ports are not used as part of the top event. The new models simply give the corresponding variable deviation for the top event name. The causes of the top event are then retrieved from the vessel model by the synthesis program.

The top event model generation program EVTGEN, which converts the text into a numerical form, was modified to accept this blank character and instead of assigning a positive integer corresponding to the text number it assigns '-1'.

As an additional extra of this method the component letter, when defining a composition deviation, can be left blank for both pipe type events and vessel type events and the program will treat this in the same way. This is advantageous because there can be up to twenty components.

This gives the top event model library a component and port independent set of models.

The configuration data input program MASTER reads in the top event specified by the user. If it is a context independent model, the user must enter the component letter and/or port number required. The program then substitutes the correct index number into the model.

The component letter system also adds the component letter chosen by the user to the end of the top event name which appears in the text at the top of the fault tree. This allows fault trees to be drawn for different components without confusion.

5.7.2 Top Event Models For Pipes And Pipe-Type Units

While working on the project it was noted that there was a difference in the formats used for top event models. The majority of the models used variable descriptions such as Q2OUT and T2OUT. This is expected because Q and T are normally defined at the outlet ports (see Section 5.2.2.3). However, the models for the pressure events used the variable descriptions P1IN and R2OUT. Although the methodology permits this it is not a normal occurrence because P is defined at outlet ports and R at inlet ports.

Clearly it is desirable to maintain a consistent methodology so a study was carried out to determine if there were any reasons for using this format and if there are any reasons for not using the standard methodology. To help with this the configuration in Figure 5.3 was used.

This is a simple flow control loop but due to the relationship between flow and pressure the configuration is sufficient for this example. The top event low pressure in unit 4 will be examined.

It should be noted that pressure P is used to trace deviations to causes in upstream units using the outlet ports of each unit. However, since relief R is transmitted against the normal direction of flow it is used to trace deviations to causes in downstream units using the inlet ports of each unit.

Figure 5.3 - Configuration For Flow Control
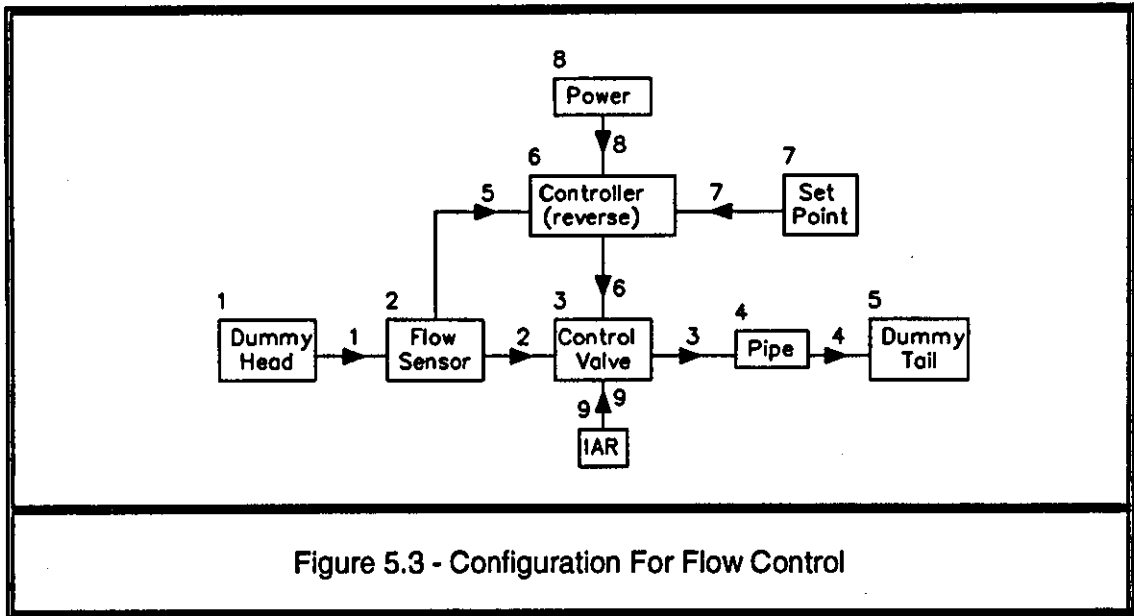
Using the original methodology for the causes we have P1IN LO and R2OUT HI. The former translates to the event P 3 LO which is the inlet connection to the pipe unit. The minitree for this is found in the next unit because the pressure deviation is associated with outlet ports and the connection forms the outlet from the control valve. The latter translates to R 4 HI which is the outlet from the pipe unit. The minitree for this is found in the next unit because relief is associated with inlet ports and the connection forms the inlet to the dummy tail unit. The synthesised fault tree is shown in Figure 5.4.

Figure 5.4 - Fault Tree For Low Pressure Into The Pipe
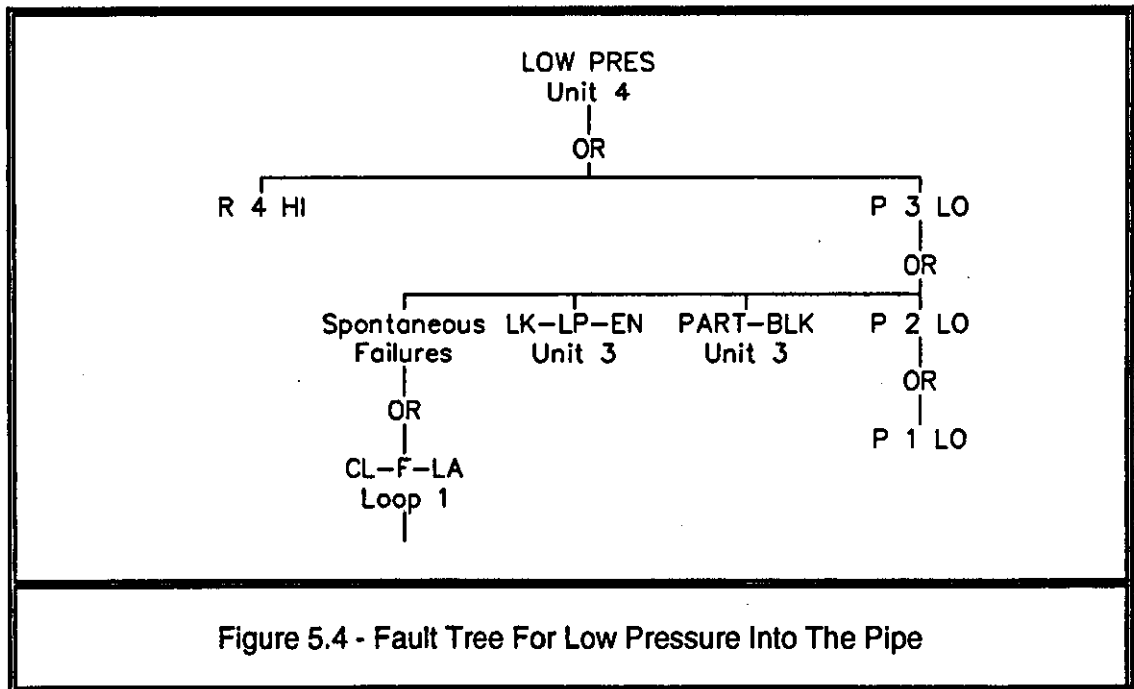
However this system by-passes the unit where the top event occurs.

Next the standard format was used with R1 IN HI and P2OUT LO as the causes of the top event. These are translated to P 4 LO and R 3 HI, respectively, from the configuration diagram. The former is the outlet from the pipe and the latter is the inlet to the unit. Therefore the minitrees for these events are found in the unit model for the pipe and have causes P 3 LO and R 4 HI, respectively. As can be seen these events are on the other side of the unit and effectively the low pressure deviation has passed through the unit. The fault tree is shown in Figure 5.5.

Figure 5.5 - Fault Tree For Low Pressure Out Of The Pipe
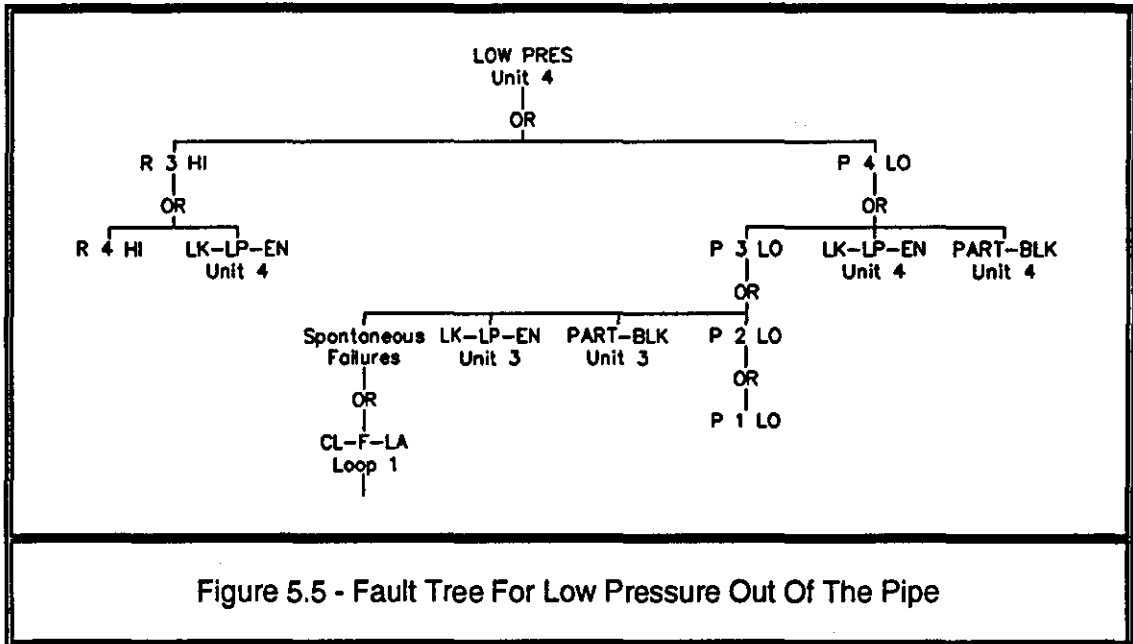
By comparing Figures 5.4 and 5.5 it can be seen that the use of the normally defined models introduces the failures of the pipe unit, unit 4. It is also found that there are no differences in the lower sections of the fault tree and this has been verified using a larger configuration.

Since there are no reasons for keeping the original format and it appears to be advantageous to change it to the standard format, this has been carried out. This also maintains a rigid adherence to the methodology.

However, it should be pointed out that the original format gave top event causes for variable deviations entering the unit, whereas the new format gives causes for variable deviations leaving the unit. This is in compliance with the rest of the top event models which use variable deviations leaving the unit. Should a user wish to examine the causes of variable deviations entering a unit the new standardised format of the top event model library will make it easier to develop the model.

5.7.3 Event Library

The modified top event model library is shown in Table 5.5. Each model number has the prefix UE for undesired event.

Table 5.5A - Top Event Models For Pipe Type Units		
Event Number	Event Name	Cause
1	HIGHFLOW	Q2OUT HI
2	SOMEFLOW	Q2OUT SOME or G1IN SOME
3	LOW FLOW	Q2OUT LO
4	NO FLOW	Q2OUT NONE
5	REV FLOW	Q2OUT REV
6	HIGHPRES	P2OUT HI or R1IN LO or R1IN NONE or R1IN REV
7	SOMEPRES	P2OUT SOME or R1IN SOME
8	LOW PRES	P2OUT LO or R1IN HI
9	NO PRES	P2OUT NONE or R1IN NOP
10	NORELIEF	R1IN NONE or P2OUT NOR
11	REV PRES	P2OUT REV large pressure sink upstream
12	REV REL	R1IN REV large pressure source downstream
13	HIGHTEMP	T2OUT HI and Q2OUT SOME or U1IN HI and G1IN REV
14	LOW TEMP	T2OUT LO and Q2OUT SOME or U1IN LO and G1IN REV
15	HIGHIMPa	Xa2OUT HI and Q2OUT SOME or Ya1IN HI and G1IN REV
16	LOW IMPa	Xa2OUT LO and Q2OUT SOME or Ya1IN LO and G1IN REV

Table 5.5B - Top Event Models For Vessel Type Units		
Event Number	Event Name	Cause
20	OVERLEVL	LvVES HI
21	UNDRLEVL	LvVES LO
22	EMPTY	LvVES NONE
23	OVERPRES	PvVES HI
24	UNDRPRES	PvVES LO
25	OVERTEMP	TvVES HI
26	UNDRTEMP	TvVES LO
27	OVERCONa	XavVES HI
28	UNDRCONa	XavVES LO
29	OVER WT	MvVES HI
30	UNDR WT	MvVES LO

Table 5.5C - Other Top Event Models		
Event Number	Event Name	Cause
50	TANK-RUP	Complex
51	VES-RUP	Complex
99	SEQ-ABRT	SEQ-F-AT or SEQ-F-AF

Notes:

1. The 'a' in the composition events show the position of the blank in the model where the component letter has been omitted.

2. The 'v' in the vessel events show the position of the blank in the model where the vessel port number has been omitted.

3. The top events under other models are models used for specific configurations and the SEQ-ABRT event is for the sequence facility.

5.8 Secondary Failure Model Library List

A secondary failure is an additional failure caused by a variable deviation. An example is low temperature in a pipe causing a blockage by freezing. This effect could be contained within the unit model but it then makes the model less context independent. Hence, the secondary failures are modelled separately in a similar way to top event models.

The original methodology dealt with two specific types of secondary failure. The first involved physical properties and phase change effects. An example is low temperature in a pipe causing low flow due to increased fluid viscosity. The second involved materials failures. An example is a leak in a pipe caused by corrosion as a result of an impurity.

This terminology is no longer applicable as the scope of the secondary failures has widened beyond simple physical properties and phase change effects and materials failures.

New definitions of the secondary failures has been introduced to reflect this increased functionality. The effects of a secondary failure can be modelled in two ways, either as a basic fault or a variable deviation. Effects which use variable deviations are called Type I failures and those which use basic faults are called Type II failures. However, the models for the secondary failures are still given the 'PP' or 'MF' file prefixes in the library.

5.8.1 Type I Failures

Consider a heat exchanger used to cool a nitric acid stream with water. If an internal leak occurs acid may enter the water stream or water enter the acid stream. In either case the result is an impurity in the stream. Considering water in the acid stream then this may cause an exothermic reaction resulting in an increase in stream temperature. The secondary failure in this case is DILUTION and its causes are high concentration of the impurity and its effects are high temperature.

This will appear in the fault tree as a secondary branch under a high temperature deviation. The first branch is labelled 'Normal Causes' and traces the temperature deviation to other causes. The Second branch is labelled 'Type I DILUTION' and traces causes of a high concentration.

Some example models with the prefix PP are given in Table 5.6. It should be noted that names are limited to eight characters.

Table 5.6 - Type I Failure Models			
Event Number	Event Name	Cause	Example Effects
1 #	DILUTION	X2OUT HI and Q2OUT SOME Y1IN HI and G1IN REV	T HI U HI
2	FLASHING	P2OUT REV R1IN HI and G1IN REV	T LO U LO
3	VAPORISE	T2OUT HI and Q2OUT SOME U1IN HI and G1IN REV	P HI
4	CONDENSE	T2OUT LO and Q2OUT SOME U1IN LO and G1IN REV	P LO
5	LIQUEFY	P2OUT HI R1IN REV and G1IN REV	T HI U HI
6	THICKEN	T2OUT LO and Q2OUT SOME U1IN LO and G1IN REV	Q LO G LO

The secondary event models contain just the causes of the event linked to specific port numbers and port types. When the secondary failure is identified by the configuration data input program MASTER the user is prompted for the effects of the event. This input consists of the variable letter and the deviation only. The port numbers and types are not required for the effects because they are dealt within a plant context rather than a unit model context. Some example effects are given in Table 5.6.

The event with the '#' already exists in the model library. The others have been given as examples of the type of event that can be used.

5.8.2 Type II Failures

This is used to extend the fault events from a simple base event to its variable deviation causes. It was initially used to represent leaks due to failures in materials of construction by corrosion (hence the MF prefix) but has been extended to cover all basic faults.

Consider a simple pipe line containing a liquid which is close to its freezing point. Should the temperature drop then the liquid freezes and causes a partial blockage. The secondary failure in this case is FREEZING and its causes are low temperature and its effect is PART-BLK.

This will appear in the fault tree as a secondary branch under the PART-BLK fault. The fault becomes an intermediate event with a branch to the basic fault of the same name and a branch labelled 'Type II FREEZING'. The latter branch traces causes of low temperature.

Some example models with the prefix MF and are given in Table 5.7. It should be noted that names are limited to eight characters. Again the models contain just the causes. The effect is entered by the user in the form of a basic fault. Some example effects are given in Table 5.7.

Table 5.7 - Type II Failure Models			
Event Number	Event Name	Cause	Example Effects
1 #	CORROSN	X2OUT HI and Q2OUT SOME Y1IN HI and G1IN REV	LK-LP-EN
2 #	FREEZING	T2OUT LO and Q2OUT SOME U1IN LO and G1IN REV	PART-BLK COMP-BLK
3 #	FREEZE1	E3IN LO	PART-BLK
4 #	FREEZE2	E3IN NONE	COMP-BLK
5	POLYRISE	T2OUT HI and Q2OUT SOME U1IN HI and G1IN REV	PART-BLK

The events with the '#' already exist in the model library. The other has been given as an example of different causes of the same effect.

5.9 Summary

This chapter has given an insight into the library lists used by the synthesis package. Discussion has been given on changes and improvements made during the project. These include the addition of two new variables for mass and electric current, the standardisation of the unit model library, the standardisation of the top event model library and a revision to the secondary failures model library.

5.10 References

1. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University of Technology; 1987.

2. Mullhi J. S.; Rules For The Computer-Aided Synthesis Of Fault Trees For Process Plants; PhD Thesis, Loughborough University of Technology; 1990.

3. Kelly B. E., Lees F. P.; The Propagation Of Faults In Process Plants: 1. Modelling Of Fault Propagation; Reliability Engineering, Volume 16, Number 1; Elsevier Applied Science Publishers; 1986; pp 12–14.

4. Kelly B. E., Lees F. P.; Faultfinder, An Interactive Computer-based Facility For Fault Propagation In Process Plants: Model And Event Library Manual, Version 1; Plant Engineering Group, Department of Chemical Engineering, Loughborough University Of Technology; 1983.

Chapter 6

Rules For Modelling Flow And Pressure

6.1 Introduction

This chapter looks at the current methodology used for modelling flow and pressure. This forms the base upon which fault propagation is achieved.

6.2 Review Of Flow And Pressure Modelling In Pipes

The basic principles for modelling flow and pressure were developed by Andow (1), Martin-Solis (2) and Kelly (3, 4). This has remained unchanged but additions have been made to deal with problems and situations which have occurred since the initial conception.

6.2.1 The Modelling Of Flow

The effects of flow faults propagate both upstream and downstream from the location of the fault. For example, a leak in a pipe will result in high flow into the pipe and low flow out of the pipe. This will cause high flow out of a unit upstream and low flow into a unit downstream of the pipe.

The variable Q can only propagate the flow deviations in one direction, so another is required for the other direction. Initially pressure P was chosen but confusion was encountered in the size of the absolute pressure required to cause no flow and reverse flow. However, pressure gradient G is ideal because it has a direct relationship with flow. For example, high flow is accompanied by a high pressure gradient and reverse flow (flow against the normal direction) is accompanied by a reverse pressure gradient.

The two equations used to model flow in a pipe type unit are

$$G1IN= F(Q1IN, Q2OUT)$$
$$Q2OUT= F(G1IN, G2OUT)$$

The first equation gives the pressure gradient at the inlet port as a function of the flow rate into the unit and of the flow rate through the outlet port. The second equation gives the flow rate through the outlet port as a function of the pressure gradients at the inlet port and at the outlet port. It is necessary to use both the inlet and the outlet ports because of the way two-way fault propagation is obtained.

This pair must be used in all pipe type units where the inlet flow is linked directly to the outlet flow of the unit and where there is no accumulation of mass. Additional variable names may be used inside the function part to simulate other effects such as a signal from a controller.

For low flow the minitrees will take the form shown in Figure 6.1.


```
         G  1  LO                          Q  2  LO
         (G1IN)                            (Q2OUT)
            |                                 |
            OR                                OR
            |                                 |
      ┌─────┴─────┐                     ┌─────┴─────┐
   Q  1  LO    Q  2  LO              G  1  LO    G  2  LO
   (Q1IN)      (Q2OUT)              (G1IN)       (G2OUT)

         (a)                               (b)
```

Figure 6.1 - Minitrees For Low Flow In A Pipe
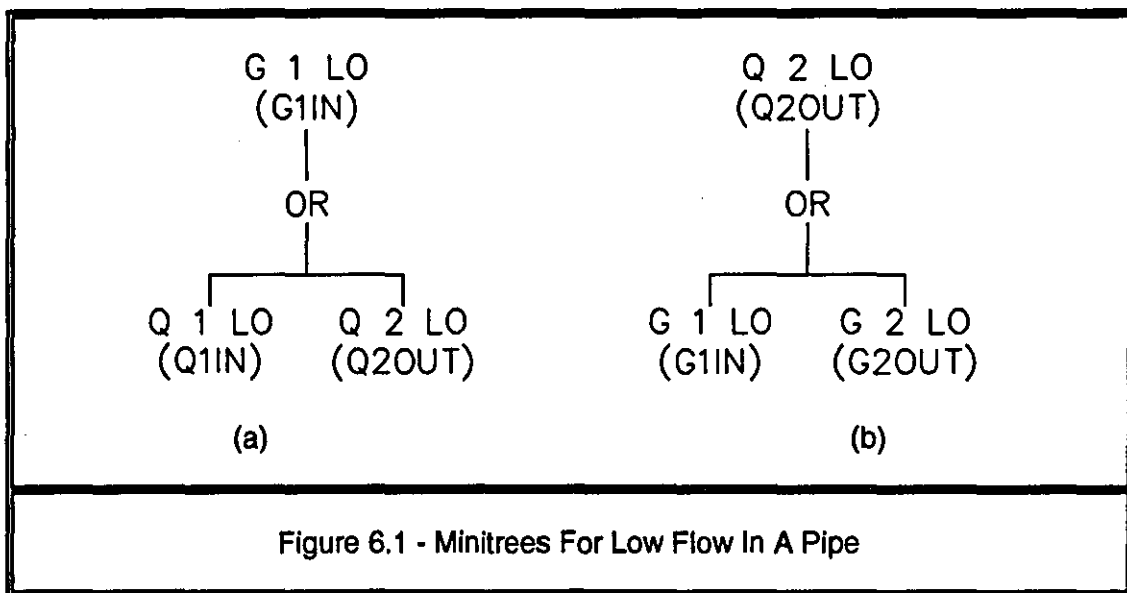
The port type identifier is not used in the fault tree. The characters in brackets are the propagation equation names so that the link between the minitree and the equations can be seen.

To show how the two-way fault propagation is obtained consider the four units linked together as shown in Figure 6.2.

Figure 6.2 - Simple Flow Section Of Plant
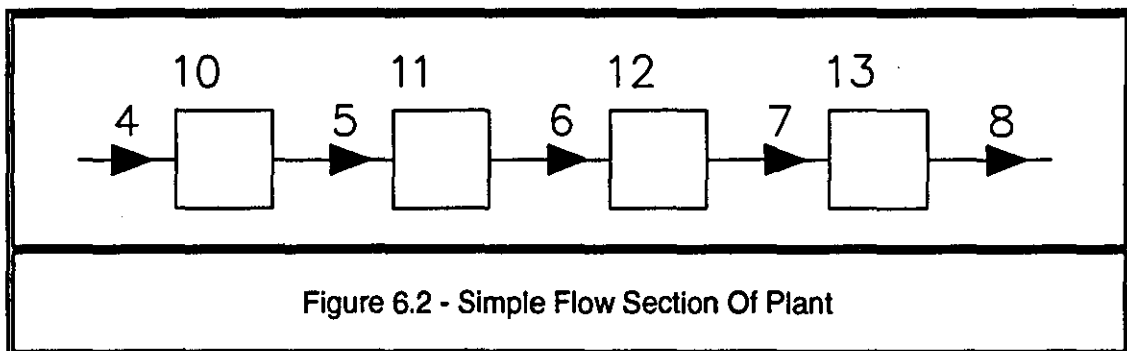
The top event is low flow out of unit 11. The outlet variable is Q, therefore the top event translates to Q 6 LO. The relevant minitree is found in the model for unit 11 and is equivalent to Figure 6.1 (b). The port numbers used in the minitrees are changed to the

appropriate connection number. In this case port 2 is connection 6 and port 1 is connection 5. The two events under the tree at this stage are G 5 LO and G 6 LO as shown in Figure 6.3.

Q 6 LO
|
OR

G 5 LO   G 6 LO
| |
OR   OR

Q 6 LO   Q 5 LO   Q 7 LO   Q 6 LO
          |
          OR   OR

G 5 LO   G 4 LO   G 7 LO   G 6 LO
          |
          OR   OR

Q 5 LO   Q 4 LO   Q 8 LO   Q 7 LO
          | |

Branch tracing low flow deviations to upstream causes

Branch tracing low flow deviations to downstream causes
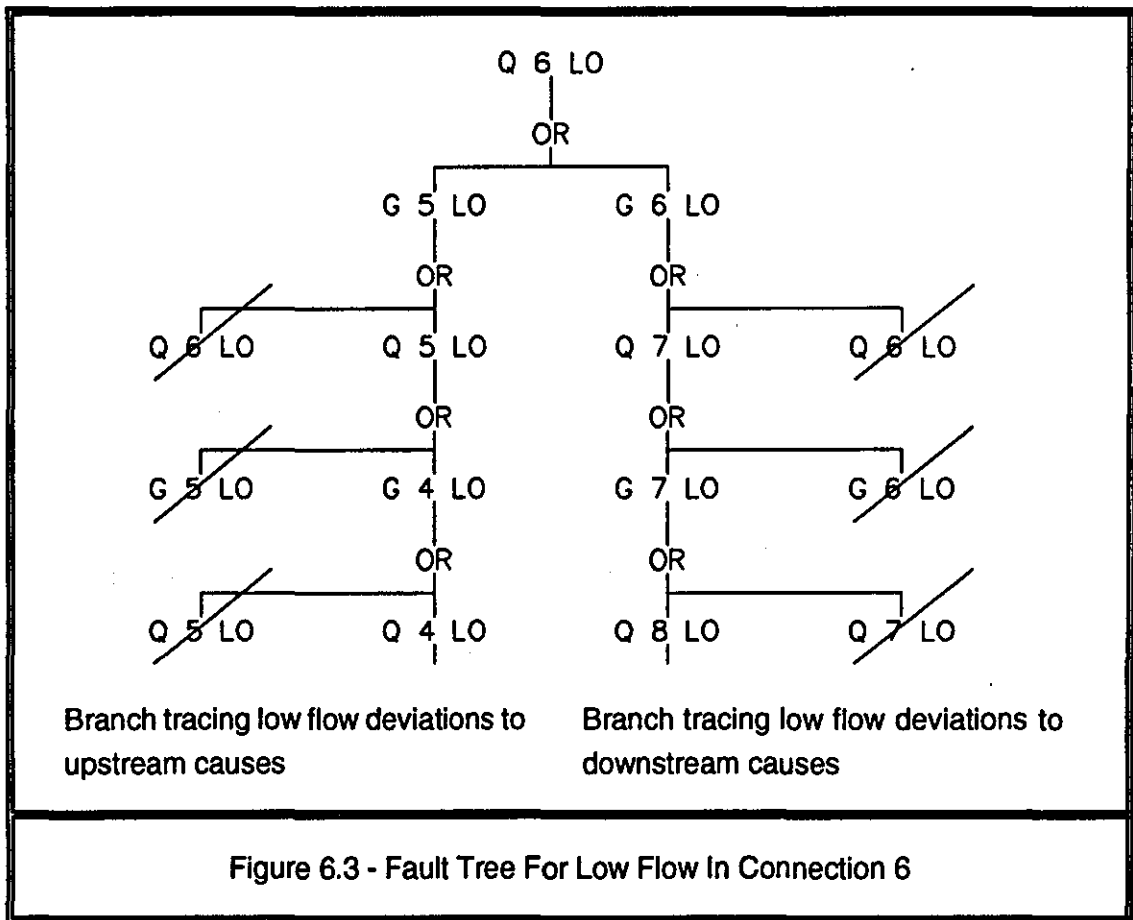
Figure 6.3 - Fault Tree For Low Flow In Connection 6

It can be seen that the branch for G 5 LO has started to trace the deviation to an upstream cause. The G variable is defined at inlet ports so the relevant minitree (Figure 6.1 (a)) will also be found in the model for unit 11. This will place the events Q 6 LO and Q 5 LO under an OR gate. However, the event Q 6 LO already exists higher up the branch. This is therefore immediately deleted by the synthesis program, leaving just Q 5 LO. The Q variable is defined at outlet ports and therefore the relevant minitree will be found in the model for unit 10. Assuming the two propagation equations are the same in each model, then the next row of the branch will be G 5 LO and G 4 LO. In this case the G 5 LO term exists higher up and is therefore deleted.

The G 6 LO term is used to trace the deviation to a cause downstream. The G variable is defined at inlets so the minitree will be contained in the model for unit 12. This provides the events Q 7 LO and Q 6 LO. However, the Q 6 LO term already exists further up the

branch and must be deleted. The Q 7 LO term is developed using the appropriate minitree from the model for unit 12. The next events are G 7 LO and G 6 LO of which the latter is deleted because it exists further up the branch.

For tracing deviations to upstream causes the Q1IN and G1IN terms are used from the propagation equations. For tracing deviations to causes downstream the Q2OUT and G2OUT terms are used from the propagation equations. The daisy-chain effect between input and output variables and G and Q continues in both directions until termination.

6.2.2 The Modelling Of Pressure

Deviations of pressure, like flow, propagate upstream and downstream, therefore here again two variables are needed. In this case the pressure variable P and the relief variable R are used. Pressure is a presence of a pressure source and relief is a presence of a pressure sink. However, the deviations of the P and R variables are not related in the same way as the deviations of the G and Q variables for flow. Therefore, the daisy-chain effect is not used on this pair.

In a process plant flow is from a point of high pressure to one of low pressure. Thus pressure deviations are transmitted in the normal direction of flow. A particular unit will experience pressure effects from units upstream. Relief deviations are transmitted against the normal direction of flow. A particular unit will experience relief effects from units downstream.

In order to model pressure the variable P is defined as propagating out of normal outlet ports and the variable R is defined as propagating out of normal inlet ports. The P OUT variable is used to trace pressure and relief deviations that are in units downstream, to causes upstream. The R IN variable is used to trace pressure and relief deviations that are in units upstream to causes downstream. This is used because a deviation propagating upstream from a downstream cause will propagate out of normal inlet ports. The two equations used to model pressure in a pipe type unit are

R1IN =F(R2OUT)
P2OUT =F(P1IN)

The first equation gives the relief at the inlet port as a function of the relief into the unit. This is linked to the relief at the inlet of the next unit downstream. The second equation gives the pressure at the outlet port as a function of the pressure at the inlet port. This is linked to the pressure at the outlet port of the next unit upstream.

The possible deviations for the variables are given in Table 6.1.

Table 6.1 - The Deviations Of Pressure And Relief	
Deviation	Description
P OUT HI	source of pressure upstream is high
P OUT LO	source of pressure upstream is low
P OUT NONE	source of pressure upstream is negligible
P OUT SOME	source of pressure upstream is significant
P OUT REV	a pressure relief sink exists upstream
P OUT NOR	no pressure relief sink exists upstream
R IN HI	sink for pressure relief downstream is high
R IN LO	sink for pressure relief downstream is low
R IN NONE	sink for pressure relief downstream is negligible
R IN SOME	sink for pressure relief downstream is significant
R IN REV	a pressure source exists downstream
R IN NOP	no pressure source exists downstream

It can be seen that certain combinations have the same effect. These are

Some relief is present if the deviation R1IN SOME or P2OUT REV occurs.
Some pressure is present if the deviation R1IN REV or P2OUT SOME occurs.
No relief is present if the deviations R1IN NONE and P2OUT NOR occur.
No pressure is present if the deviations R1IN NOP and P2OUT NONE occur.

There is no linkage between the variables P and R in individual models. The propagation of pressure P is traced upstream from the point of interest and the propagation of relief R is traced downstream from that point. Hence, both must be specified in the top event model in order for pressure faults to be traced in both directions. For example, the top event HIGHPRES, representing high pressure in a pipe, involves the following elements

events upstream -   there exists a connection to a high pressure source, represented by P1IN HI

events downstream -   there exists insufficient relief, no relief at all or a back pressure from downstream (R2OUT LO, R2OUT NONE or R2OUT REV)

These must be linked together under an OR gate from which the deviations can be traced to their causes in the appropriate direction.

6-5

6.2.3 The Modelling Of Total Component Flow

It is sometimes necessary to model the flow of a particular component as opposed to the overall flow. Consider the reaction vessel shown in Figure 6.4 which has a diluted oxygen feed. If too much oxygen enters the reaction vessel a runaway reaction may occur. Causes of too much oxygen may be a high concentration of oxygen or a high flow rate into the reaction vessel. The oxygen is diluted by an inert gas. Therefore, a low flow of inert gas may cause a high oxygen concentration or a high flow of inert gas may cause a high flow rate into the reaction vessel. Neither, however, results in too much oxygen entering the reaction vessel.

Figure 6.4 - Simple Reaction Vessel
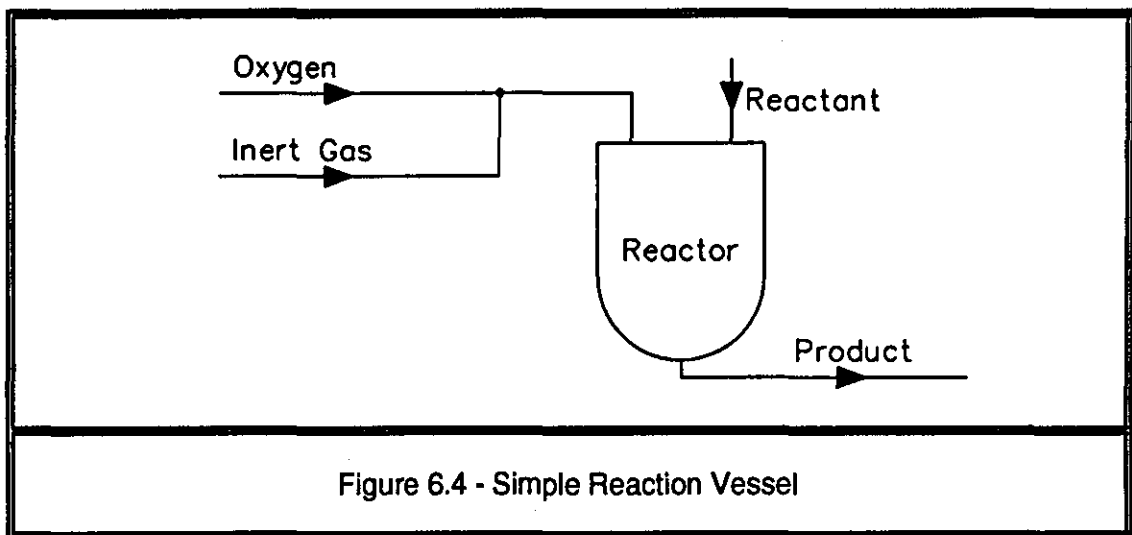
Total component flow is designed to overcome this problem by removing the contradictory events. It is the product of the flow rate and composition of a component in a stream. The total component flow operator '⊗', takes the form

T3VES= F(G1IN ⊗ X1IN)

The propagation equation is resolved into minitrees where G1IN and X1IN have the same deviations. The total component flow term can also be used in event statements which take the form

V G1IN ⊗ X1IN LO: effects

However, the algorithm used to resolve the event statements and decision tables into numerical minitrees means that the total component flow operator cannot be combined with an AND gate because this results in all the other terms being affected by the operator as well.

6.3 Pressure Deviations In Vessels

The version 2 methodology of Mullhi (5) for modelling pressure uses the same set of deviations for vessels as for pipe type units. This gives the set of variable and deviation combinations for vessels given in Table 6.2.

Table 6.2 - The Old Deviations For Pressure In Vessels	
Deviation	Description
P VES HI	high pressure inside vessel
P VES LO	low pressure inside vessel
P VES NONE	no pressure inside vessel
P VES SOME	some pressure inside vessel
P VES REV	reverse pressure inside vessel
P VES NOR	no pressure relief sink inside vessel

Not all of these combinations are appropriate to vessels. The 'some' deviation can be removed by assuming that a vessel will normally always have a certain degree of pressure. The 'none' deviation is a more extreme form of low pressure. When applied to vessels the 'reverse' deviation has no physical meaning and should be removed. The 'nor' deviation is a more extreme form of high pressure.

The methodology for pressure in vessels has been modified to remove these anomalies. The new deviation list is given in Table 6.3.

Table 6.3 - The New Deviations For Pressure In Vessels	
Deviation	Description
P VES NOR	no pressure relief sink inside vessel
P VES HI	high pressure inside vessel
P VES LO	low pressure inside vessel
P VES NONE	no significant pressure inside vessel

In order for the new methodology to be incorporated some alterations had to be made to the synthesis package. A separate file for the vessel pressure deviations had to be created. This simply contains a list of deviations which can be coupled with the pressure variable and vessel port type. The general reference files giving pointer information had a new line added to them giving pointer numbers specifically for the vessel pressure. The

programs were updated to make use of this new information. This involved changing the index number for pressure to the index number for vessel pressure if the port type of the variable deviation was a vessel port.

This has eliminated some of the problems encountered when pipe type deviations for pressure were applied to vessel ports.

6.4 The Relationship Between Flow And Pressure In Vessels

Vessels are defined as units in which there is an accumulation of mass and/or energy. This may affect the deviations of variables at the inlet and/or outlet ports. For example, the liquid level may affect the flow in and the flow out of the vessel.

To model vessels it is necessary to be able to model certain internal variables. By definition, variables can only exist at ports, so an internal port is required. This is called the vessel port since it is mainly used on vessels. Variables associated with this port are pressure, temperature, composition, level, mass and flow between phases. The latter is used in reboilers and condensers and is described in Chapter 8.

Due to the accumulation effects deviations do not propagate directly from the inlet port to the outlet port or vice versa. The incoming variable deviations are transmitted to the vessel port variables and these affect the out going variables.

6.5 Original Methodology For Modelling Flow And Pressure In Vessels

The original methodology for modelling vessels was very sketchy on which variables affect what. The next four sub-sections give a brief account of the principles involved for four types of vessel. These describe an open vessel and closed vessels containing liquid, gas/vapour or both.

The variables used for modelling were the level and the internal pressure. The level represented the accumulation of liquid in the vessel and also the pressure on a port below the liquid surface equivalent to the head of liquid above that port. The internal pressure was simply the gas/vapour pressure above any liquid level in the vessel.

6.5.1 Open Vessel For Liquids

Open vessels only contain liquid. Flow into and out of the vessel was modelled using the level. These vessels do not have an internal pressure but the liquid above the port opening exerts a pressure equivalent to the head of liquid. Therefore, the pressure into and out of the vessel was modelled using the level of liquid. The level was modelled using the flows in and out of the vessel. The following propagation equations were used in the original model

```
G1IN= F( Q1IN, L3VES)
R1IN= F( -L3VES)
Q2OUT= F( G2OUT, L3VES)
P2OUT= F( L3VES)
L3VES= F( G1IN, -Q2OUT)
```

6.5.2 Closed Vessel For Gases/Vapours

A closed vessel containing only gas/vapour does not have a level term. The flows and pressures into and out of the vessel were modelled using the internal pressure. The internal pressure was modelled using the gas/vapour flow in and out of the vessel. However, this caused problems when modelling pressures in a plant.

It is clear that high flow rate of gas into the vessel may cause a high internal pressure but also if the gas entering the vessel is at a high pressure then this may also cause high internal pressure. The model containing just the propagation equation for pressure adequately modelled the effect of flow on the internal vessel pressure but not the incoming gas pressure. In order to model pressure effects more fully it was necessary to use an event statement and a decision table. The event statement simply related high inlet

pressure to high internal pressure and the decision table gave the effect of no relief into the vessel and out of the vessel as a cause of high internal pressure. The following propagation equations, event statement and decision table were used in the original model

    G1IN= F(Q1IN, -P3VES)
    R1IN= F(-P3VES)
    Q2OUT= F(G2OUT, P3VES)
    P2OUT= F(P3VES)
    P3VES= F(G1IN, -Q2OUT)

    V P1IN HI: P3VES HI
    V P1IN NOR AND V R2OUT NONE: P3VES HI

6.5.3 Closed Vessel For Liquids With·A Confined Vapour Space

A closed vessel containing a liquid and a confined vapour space only has ports for the liquid leaving the vapour trapped. This vessel has both a level term and an internal pressure term. The flows and pressures into and out of the vessel were modelled using the internal pressure because the level term was incorporated in the pressure term. The level was modelled in the same way as the open vessel, as a function of the flows of liquid into and out of the vessel. There are no gas flows but it was assumed that the internal pressure was dependent on the level of liquid and the temperature of the vessel. The level was used because as the level rises the vapour will be compressed increasing the internal pressure. The temperature was used because the vapour above the liquid is usually the gaseous phase of the liquid. This means that the quantity and pressure of the vapour will be dependent on the temperature of the vessel. As the temperature rises the vapour pressure of the liquid will increase. The following propagation equations were used in the original model

    G1IN= F(Q1IN, -P3VES)
    R1IN= F(-P3VES)
    Q2OUT= F(G2OUT, P3VES)
    P2OUT= F(P3VES)
    L3VES= F(G1IN, -Q2OUT)
    P3VES= F(L3VES, T3VES)

6.5.4 Closed Vessel For Liquids With Gas/Vapour Space Connections

A closed vessel containing both liquids and gases/vapours has five ports, an inlet and outlet for the liquid, an inlet and outlet for the gas/vapour and a vessel port. The liquid ports were assumed to always be below the level of the liquid and the gas/vapour ports always above it. Again the flows and pressures of all the ports were modelled using the internal pressure. The level was modelled as before as a function of the flows of liquids into and out of the vessel. The internal pressure was modelled using the flows of the gas/vapour in addition to the level and temperature. The same event statement and decision table was used to model the pressure effects as in the gas/vapour vessel. The following propagation equations, event statements and decision tables were used in the original model

```
G1IN= F( Q1IN, -P5VES)
R1IN= F( -P5VES)
Q2OUT= F( G2OUT, P5VES)
P2OUT= F( P5VES)
G3IN= F( Q3IN, -P5VES)
R3IN= F( -P5VES)
Q4OUT= F( G4OUT, P5VES)
P4OUT= F( P5VES)
L5VES= F( G1IN, -Q2OUT)
P5VES= F( G3IN, -Q4OUT, L5VES, T5VES)


V P3IN HI: P5VES HI
V P3IN NOR AND V R4OUT NONE: P5VES HI
```

6.5.5 Problems With The Original Methodology

The four vessel models described in the previous sections use a different methodology for the different types of vessel. The level is used to model liquids and the internal pressure is used for gases/vapours. In turn these internal variables affect either the liquid flows or the gas/vapour flows. The relief and pressure deviations are also affected by either the level or pressure.

Clearly it is desirable to combine these differences into a single term such that a systematic approach can be used for automatic vessel model generation.

6.6 The Effective Pressure In Vessels

It is possible to simplify the methodology by redefining the pressure term in a vessel. Instead of using the internal pressure, which refers to the gas/vapour space, the 'effective vessel pressure' is used and is applied to both liquids and gases/vapours.

The principles used are similar to the original methodology for a vessel containing both liquids and gases/vapours. The internal pressure was used for gases/vapours and represented the pressure exerted by this phase. The effective vessel pressure incorporates the effects the liquid in the vessel has on a port. There are three sets of terms used in the effective vessel pressure propagation equation

- the flows of gases/vapours into and out of the vessel, if these ports exist
- the level of the liquid in the vessel, if liquid is present
- the internal temperature of the vessel, if a there is a gas/vapour space

These translate to

P5VES= F(G3IN, -Q4OUT, and/or -L5VES, and/or T5VES)

The level is still defined as a function of the flow rate of liquid into and out of the vessel.

L5VES= F(G1IN, -Q2OUT)

The flows and reliefs into the vessel are negative functions of the effective vessel pressure, if this applies, for all port types.

G1IN= F(Q1IN, -P5VES)
G3IN= F(Q3IN, -P5VES)

R1IN= F(-P5VES)
R3IN= F(-P5VES)

The flows and pressures out of the vessel are positive functions of the effective pressure, if this applies, for all port types.

Q2OUT= F(G2OUT, P5VES)
Q4OUT= F(G4OUT, P5VES)

P2OUT= F(P5VES)
P4OUT= F(P5VES)

This makes all the flow and pressure propagation equations identical for all the vessels and the level propagation equation is the same for all vessel with liquids. The only changeable propagation equation is for the effective pressure and this is made up from a combination of four variables which should be used for different situations.

In the following sections it is convenient to consider ports to be either in the gas/vapour space or at the bottom of the liquid space. A port in the liquid space but above the bottom simply has an effective pressure equal to one at the bottom of the liquid space less the effective pressure equivalent to the height of the port above the bottom. However, deviations of the level will cause the same deviations in the pressure regardless of the positions of the port providing it remains submerged.

Some important distinctions on effective pressure for any inlet or outlet port are

Vessel open, liquid only: liquid level

Vessel closed, liquid plus gas/vapour space: liquid level +
                                               gas/vapour temperature

Vessel closed, gas/vapour only: gas/vapour pressure +
                                               gas/vapour temperature

Vessel closed, liquid and gas/vapour   liquid level +
                                               gas/vapour pressure +
                                               gas/vapour temperature

6.7 Rules For Modelling Flow And Pressure In Vessels

The previous sections have shown that modelling flow and pressure in vessels is quite complex. The introduction of the effective vessel pressure has reduced the problem and the next sections deal with simplifying the process by grouping the vessels under different headings. Specific propagation equations can then be applied along with their associated event statements and decision tables.

In order to describe the flow and pressure relationship it is necessary to make distinctions between various types of vessel and port. These are

    Vessel fluid   - Liquid
                    - Gas/Vapour
                    - Liquid and Gas/Vapour

    Vessel closure   - Vessel open (to atmosphere)
                      - Vessel closed

    Vessel port location   - Liquid space, at bottom
                            - Liquid space, above bottom
                            - Gas/Vapour space

    Flow to connecting unit   - Dependent
                                - Independent

Dependent is defined here to mean that flow is affected by the effective pressure of the vessel.

6.7.1 Unit Connections

For flow into or out of the port of a vessel the driving force is the pressure difference between the vessel and the next unit. Two types of unit have been considered, another vessel and a pressure raising device (compressor or pump). The next unit may be upstream or downstream. This gives rise to the arrangements of unit connections given in Table 6.4 and Figure 6.5.

Table 6.4 - Basic Unit Connections	
Upstream Unit	Downstream Unit
Vessel	Vessel
PRC	Vessel
PRR	Vessel
Vessel	PRC
Vessel	PRR

Notes:

PRC = Pressure raiser, centrifugal
PRR = Pressure raiser, reciprocating, rotary

Figure 6.5 - Basic Unit Connections
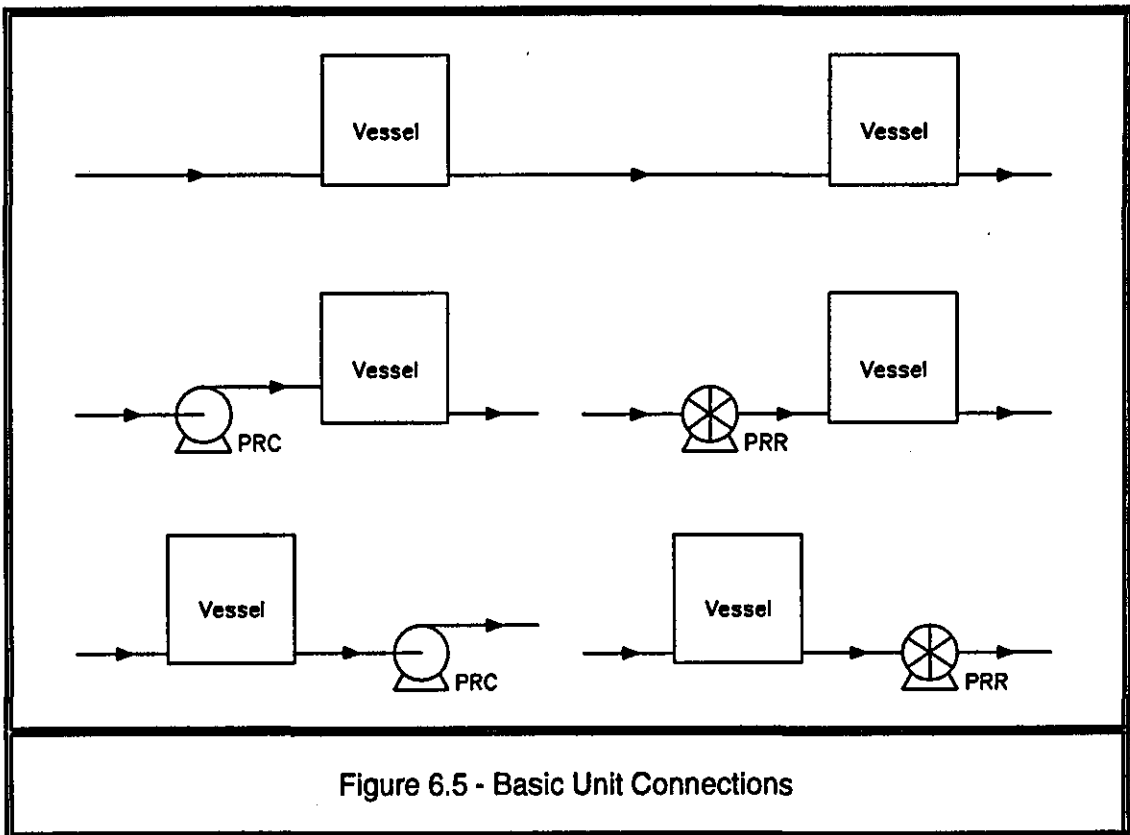
The difference in pressure also determines if the flow into or out of the unit is affected by changes in the effective pressure of the vessel. Therefore it is necessary to define the connections which are based on the vessel and port types and the unit connections. The distinctions made give rise to the arrangements of flow connections shown in Table 6.5.

Table 6.5 - Flow Connections				
Case	Upstream Unit	Downstream Unit	Fluid	Flow Connection
1	Open vessel	Open vessel	Liquid	Dependent
2	Open vessel	Open vessel	Liquid	Independent
3	Open vessel	Closed vessel	Liquid	Dependent
4	Closed vessel	Closed vessel	Liquid	Dependent
5	Closed vessel	Closed vessel	Gas/vapour	Dependent
6	Closed vessel	Open vessel	Liquid	Dependent
7	Closed vessel	Open vessel	Liquid	Independent
8	PRC	Open vessel	Liquid	Dependent
9	PRC	Open vessel	Liquid	Independent
10	PRC	Closed vessel	Liquid	Dependent
11	PRC	Closed vessel	Liquid	Independent
12	PRC	Closed vessel	Gas/vapour	Dependent
13	PRR	Open vessel	Liquid	Independent
14	PRR	Closed vessel	Liquid	Independent
15	PRR	Closed vessel	Gas/vapour	Independent
16	Open vessel	PRC	Liquid	Independent
17	Closed vessel	PRC	Liquid	Independent
18	Closed vessel	PRC	Gas/vapour	Independent
19	Open vessel	PRR	Liquid	Independent
20	Closed vessel	PRR	Liquid	Independent
21	Closed vessel	PRR	Gas/vapour	Independent

In some cases the flow can be dependent or independent. This occurs with open vessels where the elevation of the downstream unit can be above or below the upstream unit. In cases 10 and 11, the effective pressure difference between the pressure raiser and the closed vessel may be small or large and consequently changes in the effective pressure of the vessel may or may not affect the flow. Further information is given in Section 7.2.2.

6.7.2 Propagation Equations And Associated Event Statements

Having defined the arrangements to be considered, it is now possible to develop the corresponding propagation equations. The equations of interest are those for flow Q, pressure gradient G, pressure P, relief R and level L.

For a vessel with one inlet, one outlet and a vessel port, the propagation equations may involve the following terms.

Inlet port

G1IN= F(Q1IN and -P3VES)
R1IN= F(-P3VES)
U1IN= F(T3VES)
Y1IN= F(X3VES)

Outlet port

Q2OUT= F(G2OUT and P3VES)
P2OUT= F(P3VES)
T2OUT= F(T3VES)
X2OUT= F(X3VES)

Vessel port

L3VES= F(G1IN, -Q2OUT)
P3VES= F(G1IN, -Q2OUT and/or L3VES and/or T5VES)
T3VES= F(T1IN)
X3VES= F(X1IN)

Notes

1. These do not include the terms for pressure raising devices.
2. The temperature and composition equations have been included for completeness,
   they are the same for all types. No further references to them are made.
3. The P3VES term is the effective pressure.

Event statements have to be used to supplement the propagation equations. Those used
depend on the type of vessel and the type of port. The event statements are given in the
sections for each port type.

6.7.2.1 Inlet Ports

The inlet port of interest is on the downstream unit. This is connected to the outlet port
on the upstream unit. The flow characteristic between the two units depends on the units
themselves and is given in Table 6.5. Cases 16 to 21 are not used because these are
pressure raising devices.

The basic equation for pressure gradient G is

G1IN= F(Q1IN)

This is applied to units which have independent flow.

If there is a dependence on the effective pressure the propagation equation is

G1IN= F(Q1IN, -P3VES)

The propagation equation for relief R is the same for all the vessel type units. It shows that the relief and pressure into the vessel is dependent on the effective pressure.

R1IN= F(-P3VES)

An event statement is used to give the causes of no or reverse flow or pressure as high effective pressure.

V P3VES HI: G1IN NONE, G1IN REV, R1IN NONE, R1IN REV

In the cases considered in Table 6.5, the propagation equation for G at the inlet port of the downstream unit is shown in Table 6.6.

Table 6.6 - Inlet Ports	
Case	Propagation Equations
1	G1IN= F(Q1IN, -P3VES)
2	G1IN= F(Q1IN)
3	G1IN= F(Q1IN, -P3VES)
4	G1IN= F(Q1IN, -P3VES)
5	G1IN= F(Q1IN, -P3VES)
6	G1IN= F(Q1IN, -P3VES)
7	G1IN= F(Q1IN)
8	G1IN= F(Q1IN, -P3VES)
9	G1IN= F(Q1IN)
10	G1IN= F(Q1IN, -P3VES)
11	G1IN= F(Q1IN)
12	G1IN= F(Q1IN, -P3VES)
13	G1IN= F(Q1IN)
14	G1IN= F(Q1IN)
15	G1IN= F(Q1IN)

6.7.2.2 Outlet Ports

The outlet port of interest is on the upstream unit. This is connected to the inlet port on the downstream unit. The flow characteristic between the two units depends on the units themselves and is given in Table 6.5. Cases 8 to 15 are not used because these units are pressure raising devices.

The basic equation for flow Q is

   Q2OUT= F(G2OUT)

This is applied to units which have independent flow.

If there is a dependence on the effective pressure then the P3VES term must be added to give

   Q2OUT= F(G2OUT, P3VES)

The propagation equation for pressure P is the same for all vessel type units. It gives the pressure at the outlet port as being dependent on the effective pressure.

   P2OUT= F(P3VES)

In the cases considered in Table 6.5, the propagation equations for Q at the inlet port of the downstream unit is shown in Table 6.7.

There is one event statement that is necessary to supplement the propagation equations in some of the cases. This deals with no pressure causing reverse flow and pressure into the vessel.

   V P3VES NONE: Q2OUT REV, P2OUT REV

The cases with the '(ES)' indicate that the event statement is optional because reverse flow may not be possible through a reciprocating pump.

There are also two decision tables that are necessary to deal with the causes of no level and the consequences of no level. These are

   V G1IN NONE V Q2OUT SOME T L3VES NONE
   V G1IN NONE V L3VES NONE T Q2OUT NONE, P2OUT NONE

The right hand column of Table 6.7 indicates whether the event statement or decision tables are necessary for the particular case number.

Table 6.7 - Outlet Ports		
Case	Propagation Equation	Event Statement/ Decision Table
1	Q2OUT= F(G2OUT, P3VES)	DT
2	Q2OUT= F(G2OUT)	DT
3	Q2OUT= F(G2OUT, P3VES)	DT
4	Q2OUT= F(G2OUT, P3VES)	ES and DT
5	Q2OUT= F(G2OUT, P3VES)	ES
6	Q2OUT= F(G2OUT, P3VES)	ES and DT
7	Q2OUT= F(G2OUT)	ES and DT
16	Q2OUT= F(G2OUT)	DT
17	Q2OUT= F(G2OUT)	ES and DT
18	Q2OUT= F(G2OUT)	ES
19	Q2OUT= F(G2OUT)	DT
20	Q2OUT= F(G2OUT)	(ES) and DT
21	Q2OUT= F(G2OUT)	(ES)

6.7.2.3 Vessel Ports

The vessel of interest in this section is the downstream unit and contains either liquid or gas/vapour, as defined in Table 6.5, but not both. Only cases 1 to 15 are considered because the others are pressure raising devices which do not have vessel ports.

The temperature and composition variables are the same for all cases. They are

    T3VES= F(T1IN)
    X3VES= F(X1IN)

These propagation equations only deal with flow in the normal direction. In order to use flow in the reverse direction the equations have to be supplemented with the following decision tables

V Q2OUT REV V U2OUT HI T T3VES HI
V Q2OUT REV V U2OUT LO T T3VES LO
V Q2OUT REV V Y2OUT HI T X3VES HI
V Q2OUT REV V Y2OUT LO T X3VES LO

For a liquid only vessel the level is a function of the liquid flows into and out of the vessel. The propagation equation is

L3VES= F(G1IN, -Q2OUT)

This can only produce minitrees for events which are directly linked. Therefore it is necessary to supplement it with event statements for the more extreme variable deviations of the flows and the level. The following event statements are required

V G1IN NONE: L3VES LO
V G1IN REV: L3VES LO, L3VES NONE
V Q2OUT NONE: L3VES HI
V Q2OUT REV: L3VES HI

The second column in Table 6.8 indicates whether the level equation and associated event statements are needed in the model.

The effective pressure term depends on the type of unit being considered.

For vessels with gas/vapour only the propagation equation is a function of the gas/vapour flows into and out of the vessel and the temperature of the gas/vapour.

P3VES= F(G1IN, -Q2OUT, T3VES)

Event statements and a decision table are required to deal with the extreme deviations of the flows and additional causes of high vessel pressure.

V G1IN NONE: P3VES LO
V G1IN REV: P3VES LO
V Q2OUT NONE: P3VES HI
V Q2OUT REV: P3VES HI

This is set A in Table 6.8.

For open vessels which will contain just liquids the propagation equation is

P3VES= F(L3VES)

An additional cause of low pressure is no level which requires an event statement

V L3VES NONE: P3VES LO

This is set B in Table 6.8.

For closed vessels containing just liquids the propagation equation also has the temperature term.

P3VES= F(L3VES, T3VES)

The level/pressure event statement is also required for this vessel type.

V L3VES NONE: P3VES LO

This is set B again.

This section has dealt with three types of vessel, open containing liquid, closed containing liquid and closed containing gas/vapour. This is sufficient for modelling individual phases. However, if a vessel contains both a liquid and a gas/vapour it is necessary to change the port numbers. The propagation equations and associated event statements are the same apart from this. In the following example the vessel layout has an inlet and an outlet port for the liquid (ports 1 and 2, respectively) and an inlet and an outlet port for the gas/vapour (ports 3 and 4, respectively). The level equation is

L5VES= F(G1IN, -Q2OUT)

and has the following event statements

V G1IN NONE: L5VES LO
V G1IN REV: L5VES LO, L5VES NONE
V Q2OUT NONE: L5VES HI
V Q2OUT REV: L5VES HI

The pressure equation is

P5VES= F(G3IN, -Q4OUT, T5VES)

6-22

with the following event statements

V G3IN NONE: P5VES LO
V G3IN REV: P5VES LO
V Q4OUT NONE: P5VES HI
V Q4OUT REV: P5VES HI
V L5VES NONE: P5VES LO

If this layout is used in the model it should be noted that the port numbers must be changed accordingly in the other propagation equations and event statements used in the model.

Case	Propagation Equations		Set
1	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
2	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
3	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES, T3VES)	B
4	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES, T3VES)	B
5	-	P3VES= F(G1IN, -Q2OUT, T3VES)	A
6	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
7	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
8	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
9	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
10	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES, T3VES)	B
11	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES, T3VES)	B
12	-	P3VES= F(G1IN, -Q2OUT, T3VES)	A
13	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES)	B
14	L3VES= F(G1IN, -Q2OUT)	P3VES= F(L3VES, T3VES)	B
15	-	P3VES= F(G1IN, -Q2OUT, T3VES)	A

Table 6.8 - Vessel Ports

6.8 Summary

This chapter has given a revision of the methodology used for modelling flow and pressure in pipe type units. A new methodology has been given for the modelling of flow and pressure in vessel type units. This process has been adapted in the next chapter on automatic generation of complete vessel models.

6.9 References

1. Andow P. K.; A Method For Process Computer Alarm Analysis; PhD Thesis, Loughborough University Of Technology; 1973; pp 40–53.

2. Martin-Solis G. A.; Fault Tree Synthesis For Real Time And Design Applications On Process Plant; PhD Thesis, Loughborough University Of Technology; 1978; pp 50–69.

3. Kelly B. E., Lees F. P.; The Propagation Of Faults In Process Plants: 1. Modelling Of Fault Propagation; Reliability Engineering, Volume 16, Number 1; Elsevier Applied Science Publishers; 1986; pp 8–9.

4. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1987; pp 3-18–3-21, 4-7–4-9, 4-11–4-15.

5. Mullhi J. S.; Rules For The Computer-Aided Synthesis Of Fault Trees For Process Plants; PhD Thesis, Loughborough University Of Technology; 1990.

Chapter 7

Rules For The Generation Of Vessel Models

7.1 Introduction

In an ideal situation any computer aided fault tree synthesis package would contain a unit model library or minitree database sufficient enough to cover every possible situation. However, this is very unlikely due to the large variety of different units found in the process industry. Experience gained from studying different plant systems indicates that the most likely unit requiring a new model to be developed is the vessel or tank storage unit. For this particular class of unit the wide variety is due to the multiple combinations of inlet and outlet ports and the particular types of port used. This results in the user requiring an understanding of the complex methodology used for vessel models. Hence the processing required in developing a new model can be time consuming.

Two approaches have been used to make the process easier for the user, a vessel model generation routine and a set of vessel model template examples. They were developed in parallel so that the generation routine produced the same result as the template and the template was used to check that each vessel type and port type produced was correct.

7.2 Vessel Characteristics

In order for a systematic approach to be used to generate vessel models it is necessary to categorise the different vessels. Once the characteristics have been identified, any vessel can be sorted and modelled appropriately.

The data required to model the vessel is collected in two sections, the first is the general information about the overall vessel layout and the second is information about each port. All of this information is available on a normal piping and instrumentation diagram.

7.2.1 Overall Vessel Layout

The overall vessel information deals with the basic vessel type and its contents. These include atmospheric influences and the number and type of ports. Although there are numerous vessel types as given in Perry (2) and Coulson (4), for the purpose of fault modelling they can be grouped into either an open or a closed vessel category.

7-1

The vessel contents can be either liquid, gas/vapour or both. It is assumed that any solids that may be contained in the fluid will not affect the modelling for which this information is required. For a liquid only container it is necessary to determine if the vessel is open to the atmosphere (this includes floating roof vessels) or if it is a closed container. All vessels containing gas/vapour are assumed to be closed to the atmosphere since the gas/vapour pressure is likely to introduce pressure influences. This information is shown in Figure 7.1.

Figure 7.1 - Overall Vessel Layout
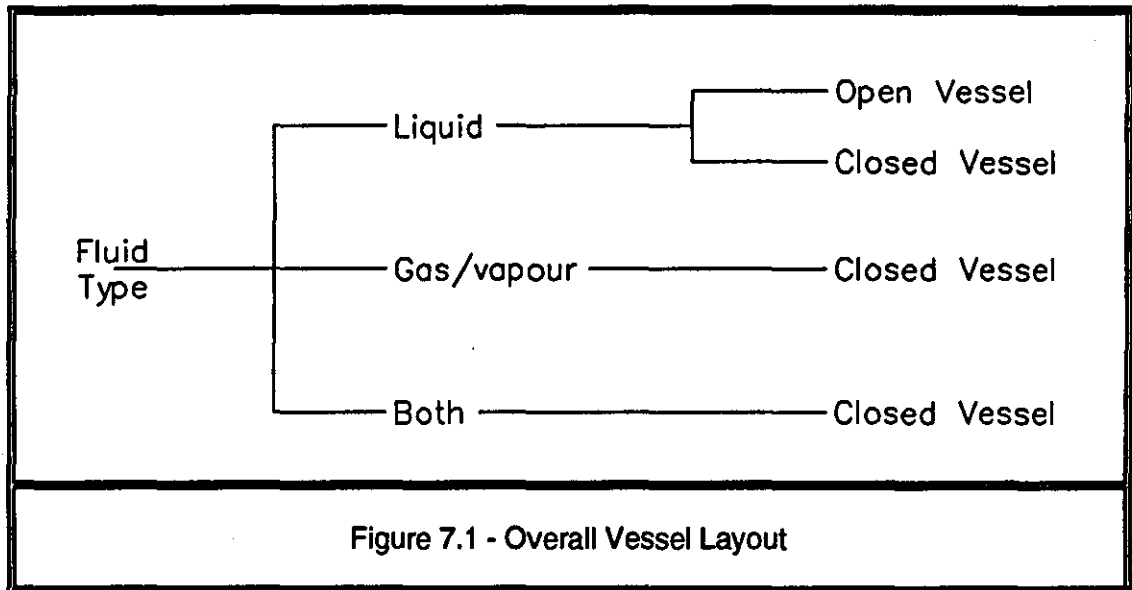
It has been assumed that a closed vessel containing liquid only is not filled to the top and therefore contains a certain amount of vapour or gas above the liquid surface. The design of the vessel is such that this vapour is unable to escape through any of the ports and must remain inside until it condenses (or dissolves in the case of a gas). For the purpose of modelling it is assumed that the mass of gas/vapour remains constant. A vessel of this type has no gas/vapour ports.

This information is required for modelling the flow through the ports and for modelling the pressure within the vessel. Since the characteristics cover open and closed vessels for liquids and gases/vapours the vessel pressure term used is really an effective vessel pressure. It deals with the pressure exerted at the ports by the liquid and/or the gas/vapour. It may depend on any combination of the gas/vapour flows in and out, the level of the liquid and the temperature of the gas/vapour in the gas/vapour space. The liquid level is considered because this will exert a pressure directly at the liquid ports proportional to the head of liquid. An increase in level will result in an increase in the effective vessel pressure. The temperature of the gas/vapour in the gas/vapour space is considered because as the temperature rises, the gas/vapour pressure rises increasing the vessel pressure.

The reason for the large numbers of different vessels is due to the enormous variations in numbers and combinations of inlet and outlet ports. The ports must be defined as either an inlet port or an outlet port. In some cases there may be situations where the port on the vessel is a non-flow port under normal operating conditions, ie. there will be normally no flow through the port. In these cases it must be decided if the port could act as an inlet port, for example the port may be connected to a redundant inlet pump, or an outlet port such as a drain line. There may, however, be instances where it is impossible to determine the port type and in these cases the port should be treated as an outlet port because it could become the source of a leak. Care should be taken with this assumption because on a vessel which operates under vacuum pressure a port which leaked would become an inlet port but if above atmospheric pressure it would become an outlet port.

7.2.2 Port Information

The information required about each port is whether it has flow or not, if it is in the gas/vapour or liquid side and how the fluid is or would be transported to or from the vessel.

In order to determine if the *no flow* variable deviation is a fault condition or a normal condition the port must be specified as having flow or not having flow under normal operating conditions. The latter case will probably be due to a closed hand valve further upstream, or downstream, as appropriate.
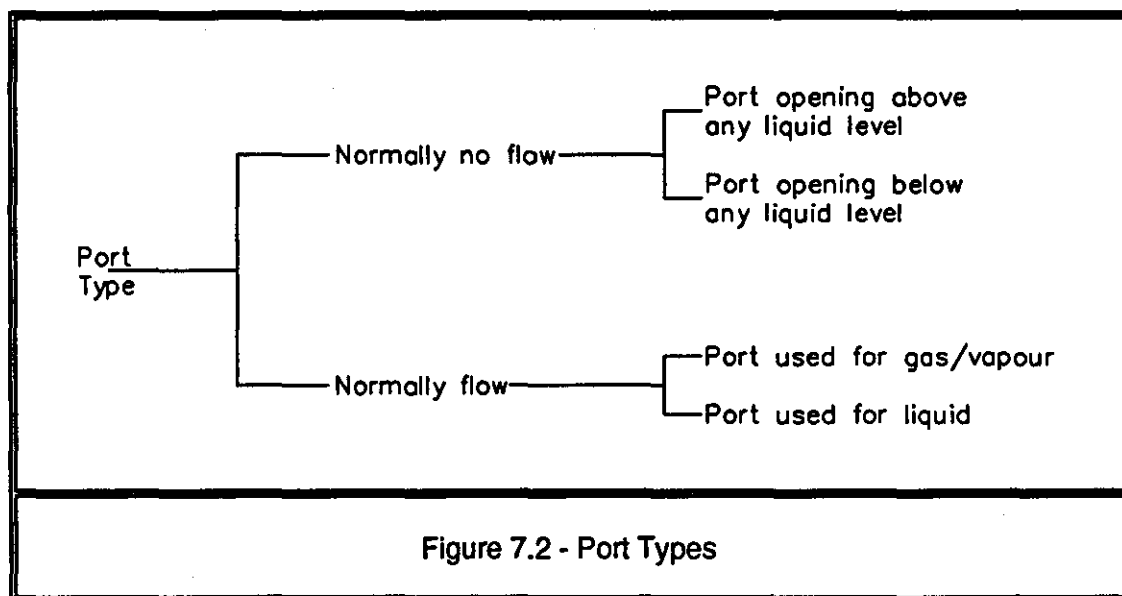
Figure 7.2 - Port Types

The flow properties of liquids and gases/vapours are modelled differently and their effects on the vessel properties are different, therefore, the type of fluid passing through the port, under normal conditions, must be specified. In situations where the stream connected to a port has two phase flow it is necessary to determine whether the fluid contains mainly

liquid or mainly gas/vapour and must be defined accordingly. An option for defining a port with two phases has not been used because this complicates the modelling required since varying degrees of saturation will result in varying effects on the level and pressure and the flow rate. For non-flow ports, the port opening may be defined as above or below the level of the liquid. This is used should a leak or flow occur through the port. This information is shown in Figure 7.2.

The flow behaviour through each of the ports is dependent on two factors. These are the vessel pressure and the ability for reverse flow. As a means of illustration three modes of transportation have been used. There is reciprocating pumping, centrifugal pumping or gravitational/ back-pressure effects. The assumptions used to categorise the pumping systems are based on information in Perry (1) and Coulson (3) and design data contained in Coulson (4). No distinction has been made between a pump and a compressor.

A reciprocating pump can generate very large pump pressures, therefore, it is assumed that the flow at this type of port is not dependent on the vessel pressure. It is also necessary to indicate if this port can have reverse flow through it. It has been assumed that this type of pump has been designed such that reverse flow through them is very unlikely if not impossible. It is, therefore, reasonable to assume that reverse flow through the port is also impossible. However, leaks may occur between the pump and the vessel which could result in reverse flow. If the port has been defined as a reciprocating pump, it is necessary to determine if reverse flow can occur. There are some rules which can be applied. An open vessel may have reverse flow at the inlet port because liquid will run out of the vessel. On this principle reverse flow cannot occur at the outlet ports as a result of a leak because flow would be in the normal direction. The same reasoning is used for a closed vessel where the internal pressure is above atmospheric pressure in that flow will be out of the vessel should a leak occur. For vessels at lower than atmospheric pressure (ie. vacuum vessels) the flow will be into the vessel, therefore, it is possible for an outlet port to have reverse flow.

Centrifugal pumps are more complicated because the modelling depends on the fluid and port type. All outlet ports to a centrifugal pump are assumed to be independent of the vessel pressure. Inlet ports for gases/vapours will usually be connected to a compressor and can be assumed to be affected by the vessel pressure. For liquids at inlet ports, the effect depends on the difference between the maximum pressure the pump can generate and the vessel pressure. For systems where the maximum pump pressure greatly exceeds the vessel pressure then clearly it will behave similarly to a reciprocating pump and flow will not be affected by the pressure. This is defined as a high pressure centrifugal pump. If the difference between pressures is small then fluctuations in the vessel pressure could

7-4

affect the flow so in this case there is a dependence on the vessel pressure. This is defined as a low pressure centrifugal pump. All centrifugal pumps are assumed to allow reverse flow.

The final category is for flow which is not induced mechanically, this may be liquids which flow under gravity or gases/vapours which flow due to a pressure difference between the vessel and another unit. In these cases there is obviously a dependence on the vessel pressure and reverse flow is possible.

The information required for each port is summarised in Table 7.1.

Table 7.1 - Port Information			
Port Type	Pressure Effects	Reverse Flow	Mode used
Gas/vapour inlet	Yes	Yes	Back-pressure
Gas/vapour inlet	Yes	Yes	Centrifugal
Gas/vapour inlet	No	Yes	Reciprocating
Gas/vapour inlet	No	No	Reciprocating
Gas/vapour outlet	Yes	Yes	Back-pressure
Gas/vapour outlet	No	Yes	Centrifugal
Gas/vapour outlet	No	Yes	Reciprocating
Gas/vapour outlet	No	No	Reciprocating
Liquid inlet	Yes	Yes	Gravity
Liquid inlet	Yes	Yes	Centrifugal (low)
Liquid inlet	No	Yes	Centrifugal (High)
Liquid inlet	No	Yes	Reciprocating
Liquid inlet	No	No	Reciprocating
Liquid outlet	Yes	Yes	Gravity
Liquid outlet	No	Yes	Centrifugal
Liquid outlet	No	Yes	Reciprocating
Liquid outlet	No	No	Reciprocating

7.3 Vessel Model Generation Program

An automatic vessel model generation subroutine has been written into the unit model generation program MODGEN. The user has the option of using this facility after the model number and name have been entered. The vessel model information described in the previous section is collected through a series of questions. These questions and, therefore, the user interaction are kept to a minimum by tailoring and only prompting for data which cannot be derived from previous answers. A simple example of this is that all vessels that contain any gases/vapours are automatically assumed to be closed.

The process is split into four subroutines, the first collects the data from the user, generates a description of the model from the data (for reference purposes), and calls the three subroutines to generate the rest of the model. The second subroutine generates the propagation equations, a third the event statements and the fourth is for the decision tables. The models produced are based on those developed by Kelly, Mullhi and Hunt as described by Hunt and Lees (5) but have a greater variation of configurations.

7.3.1 Data Collection

The vessel must be identified as containing gas/vapour, liquid or both and then, if only liquid is selected, whether it is open or closed to the atmosphere. Vessels containing any gases/vapours are automatically given a closed specification. The number of inlet and outlet ports are given as between one and seven which means that there must be at least one inlet port and one outlet port, although in some circumstances either of these may be later defined as a non-flow port. In addition to these port types it is necessary to define another port to model the internal characteristics of the vessel. This port is called the vessel port and is used to model the level, pressure, temperature and composition of the vessel as a whole. There is a restriction imposed by the FAULTFINDER code such that only nine ports can be given in total, including the vessel port. A check is carried out to ensure that these rules are adhered to.

The characteristics of each port are then determined. The user defines each port as being a flow port or a non-flow port under normal operating conditions. If the vessel contains both liquid and gas/vapour it is necessary for the user to distinguish between ports for gases/vapours and ports for liquids. The methodology used is such that the fluid passing through any port can be of only one type, ie liquid or gas/vapour. Therefore, a port specified as liquid is assumed to have its opening below the normal operating level of the vessel and all gas/vapour port openings are above any liquid level.

There may be some situations where it is desirable to model the effects of break-through or carry-over of gas/vapour or liquid to a port defined as being for the other. This scenario has not been considered in the methodology because it represents a change in the configuration of the vessel. The causes of such an occurrence can be modelled using the *high level* top event to mean high enough for liquid carry-over or the *low level* top event to mean low enough for gas/vapour break-through. The effects of having a carry-over or break-through can be modelled using the composition variable where in this case it refers to the gas/vapour or liquid as an impurity.

Finally the mode of transportation of the fluid is specified. Three options are given, the flow may be induced by gravity (or a pressure difference between the vessel and another unit), pumped by a centrifugal pump or pumped by a reciprocating pump. The option used should be determined by examining Table 7.1. Additional information may be required for different port types used. A liquid inlet port can have a low pressure centrifugal pump or a high pressure centrifugal pump for which the former has pressure effects and the latter does not. If a positive displacement pump is selected the user has to declare whether the port can or cannot have reverse flow through it. All other port types are assumed to allow reverse flow.

All the information required by the generation routines is now contained in various arrays and it is only a matter of applying the rules to them.

7.3.2 Generation Of Propagation Equations

The propagation equations are generated for each port in turn. This includes the non-flow ports because these will behave as flow ports should a fault occur that allows flow. For this reason all non-flow port variables are also included in the equations for other ports, where necessary.

There are four equations associated with each inlet port and they take the form,

$$GmIN = F (QmIN)$$ flow
$$RmIN = F (-PvVES)$$ pressure
$$UmIN = F (TvVES)$$ temperature
$$YmIN = F (XvVES)$$ composition

where 'm' is the inlet port number and 'v' is the vessel port number.

Note :   The temperature and composition propagation equations are only used if reverse flow is possible through the port.

The pressure, temperature and composition equations are the same for all types of inlet port. However, the flow equation contains a vessel pressure term if pressure effects are to be considered. The equation becomes,

$$GmIN= F (QmIN, -PvVES) \qquad \text{flow.}$$

There are four similar equations for the outlet ports and they take the form,

$$QnOUT= F (GnOUT) \qquad \text{flow}$$
$$PnOUT= F (PvVES) \qquad \text{pressure}$$
$$TnOUT= F (TvVES) \qquad \text{temperature}$$
$$XnOUT= F (XvVES) \qquad \text{composition}$$

where 'n' is the outlet port number.

If the outlet flow is affected by the pressure then the flow equation needs to contain the vessel pressure term because there is a dependence on the vessel pressure. The equation becomes,

$$QnOUT= F (GnOUT, PvVES) \qquad \text{flow.}$$

The pressure, temperature and composition equations are the same for all outlet port types.

The vessel port equations are slightly more complicated because the format depends on the nature of the vessel and the fluid in the ports. If liquid is present then the vessel port level equation is needed. The level is a positive function of all the liquid inlet flow port flows and a negative function of all the liquid outlet flow port flows. It takes the form,

$$LvVES= F (GmIN,..., -QnOUT,...) \qquad \text{liquid flow ports.}$$

The vessel port pressure term has four main formats. For an open vessel it is a positive function of the liquid level because the effective pressure of the vessel is equivalent to the pressure exerted by the liquid head above the port. In a closed vessel containing only liquid the effective pressure is a positive function of the liquid level and the temperature of the gas/vapour in the gas/vapour space above the liquid level. This is because variations in temperature will cause variations in the pressure of the gas/vapour. For a gas/vapour only vessel the pressure is a positive function of the flow rates at the inlet gas/vapour flow ports and a negative function of the flow rates at the outlet gas/vapour flow ports, plus a temperature term. The equation for vessels containing both liquid and gas/vapour contains all four terms, inlet, outlet, level and temperature. The equations are:

PvVES= F (LvVES)   liquid only, open
PvVES= F (LvVES, TvVES)   liquid only, closed
PvVES= F (GmIN,..., -QnOUT,..., TvVES)   gas/vapour only, closed,
                                                         gas/vapour flow ports
PvVES= F (GmIN,..., -QnOUT,..., LvVES, TvVES)   gas/vapour and liquid, closed,
                                                         gas vapour flow ports.

It may appear inappropriate to use the vessel pressure propagation equation for open vessels because a pressure does not really exist. However, the term is required to give a systematic approach to the model and enable automatic generation. The overall result is the same if the vessel pressure is removed and substituted for the level term in the propagation equations. The only difference is that a pressure term appears between the flow term and the level term of fault trees which use the automatically generated vessel models.

The vessel port temperature term is given as a positive function of the inlet port temperature terms only. The outlet port temperature terms require the flow to be reversed at that port in order for the vessel port temperature to be affected. Therefore the outlet port temperature deviations are ANDed with reverse flow in the decision tables given later. The equation is

TvVES= F (TmIN,...)   temperature.

The vessel port composition term is similar to the temperature term and is a positive function of the inlet port composition terms. The outlet port composition terms are dealt with in the decision tables because they require reverse flow. The equation is

XvVES= F (XmIN,...)   composition.

This completes the propagation equations required for vessel models.

7.3.3 Generation Of Event Statements

For the event statements each deviation and event is considered in turn and applied to the appropriate ports.

The first set of event statements deal with the effect of the inlet ports on the level and /or the pressure. The exact configuration depends on the type of vessel and the type of fluid.

7-9

If the port is a non-flow port then some flow through the port may cause a high level and/or pressure. The various combinations for this deviation are:

V GmIN SOME: LvVES HI   liquid non-flow port, open vessel
V GmIN SOME: LvVES HI, PvVES HI liquid non-flow port, closed vessel
V GmIN SOME: PvVES HI   gas/vapour non-flow port.

No flow through any of the normal flow ports may cause low level and/or low pressure. Deviations relating to no level are dealt with later.

V GmIN NONE: LvVES LO   liquid flow port, open vessel
V GmIN NONE: LvVES LO, PvVES LO liquid flow port, closed vessel
V GmIN NONE: PvVES LO   gas/vapour flow port.

If the port can have reverse flow then this may be a cause of low or no level and/or low pressure.

V GmIN REV: LvVES LO, LvVES NONE liquid port, open vessel, reverse flow
V GmIN REV: LvVES LO, LvVES NONE,
     PvVES LO   liquid port, closed vessel reverse flow
V GmIN REV: PvVES LO   gas/vapour port, reverse flow.

The intermediate dummy event C(DUMMY) forms part of the causes of no outlet flow and combines the no flow deviations of all the liquid flow ports. If there is more than one flow port then the deviations form a decision table generated later. However, if there is only one flow port then an event statement is used. The statement is

V GmIN NONE: C(DUMMY)   only one liquid inlet port.

The intermediate dummy event C(DUMMY) also gives the causes of no level.

I C(DUMMY): LvVES NONE   liquid inlet ports.

The next set of event statements deal with the effects of the outlet ports on the level and pressure in the vessel. If the port is non-flow then some flow will cause a low level and/or a low pressure.

V QnOUT SOME: LvVES LO   liquid non-flow port, open vessel
V QnOUT SOME: LvVES LO, PvVES LO liquid non-flow port, closed vessel
V QnOUT SOME: PvVES LO   gas/vapour non-flow port.

High level and pressure may be caused by no flow out of a normal flow port or reverse flow through any outlet port, if this is possible. The event statements are:

  V QnOUT NONE: LvVES HI   liquid flow port, open vessel
  V QnOUT NONE: LvVES HI, PvVES HI liquid flow port, closed vessel
  V QnOUT NONE: PvVES HI   gas/vapour flow port

  V QnOUT REV: LvVES HI   liquid port, open vessel, reverse flow
  V QnOUT REV: LvVES HI, PvVES HI liquid port, closed vessel, reverse flow
  V QnOUT REV: PvVES HI   gas/vapour port, reverse flow.

The following set of event statements deals with the effects of deviations at the vessel ports. High effective pressure may cause none or reverse flow or relief at the inlet ports, if this is possible.

  V PvVES HI: GmIN NONE, RmIN NONE,... flow port
  V PvVES HI: GmIN REV, RmIN REV,...   reverse flow.

Reverse flow through an outlet port may be caused by a no effective pressure in the vessel. The event statements for ports allowing reverse flow are,

  V PvVES NONE: QnOUT REV, PnOUT REV,... reverse flow.

An additional event statement is required to indicate that a *no level* deviation may cause a low pressure.

  V LvVES NONE: PvVES LO   vessel containing liquid.

The basic faults used in vessel models are the leaks, LK-LP-EN and LK-HP-EN, and the temperature effects, EXT-HEAT and EXT-COLD. The standard data files used in the Version 2 methodology for event statement generation as developed by Mullhi (7) are used to generate the event statements for these faults. Only the vessel port is specified as being affected by the fault.

A leak to a low pressure environment, LK-LP-EN, applies to most vessels and simply models the effects of fluid leaking out of the vessel. This fault may cause the level to be low or none, if liquid is present, and the vessel pressure to be low if the vessel is closed.

F LK-LP-EN: LvVES LO, LvVES NONE   open vessel
F LK-LP-EN: LvVES LO, LvVES NONE, PvVES LO   closed vessel containing
                                             liquid
F LK-LP-EN: PvVES LO   closed vessel containing
                                             gas/vapour only.

A leak from a high pressure environment, LK-HP-EN, applies only to vessels which are closed and where the internal pressure is below atmospheric pressure. The fault models the in-flow of air to the vessel which results in an impurity being present. The fault is given in vessels where the internal pressure is above atmospheric pressure because of the systematic approach used for the process, in these cases the fault can be deleted from the model later or ignored. The result of the fault is a higher vessel pressure and a high deviation to the composition variable. Multi-components are not required for this fault because the composition variable is used to model the composition of the component of interest which in this case would be the air impurity. A secondary effect is also given which deals with a large leak sufficient enough to cause such a high vessel pressure that relief into the vessel is not possible. The event statement is

    F LK-HP-EN: PvVES HI, PvVES NOR, XvVES HI   closed vessel.

An external heat source, EXT-HEAT, models the effects of things like an external fire. It may cause the vessel temperature to become high and in a closed vessel the vessel pressure may also become high. An external cold source, EXT-COLD, may cause the vessel temperature to become low and in a closed vessel the vessel pressure may also become low.

    F EXT-HEAT: TvVES HI   open vessel
    F EXT-HEAT: PvVES HI, TvVES HI   closed vessel
    F EXT-COLD: TvVES LO   open vessel
    F EXT-COLD: PvVES LO, TvVES LO   closed vessel.

Vessel models are different from many other models because the propagation equations for flow do not link up the inlet and outlet ports. In models such as the length of pipe, the outlet flow is a function of the inlet flow and from this the model generation program MODGEN, derives a minitree indicating that some flow out may be caused by some flow in. However, the flow equations for a vessel give the flow as a function of the level and/or the vessel pressure. These terms belong to the vessel port where the *some* deviation has no meaning and is not defined. This results in the *some* deviation for flow at the port having no causes and being marked as impossible. This deviation and others above it in the fault tree branch are deleted according to a set of rules for impossible events. However, this gives an incorrect fault tree because the *some* deviation may be a normal

event. It is therefore necessary to introduce the *some* deviation to the model to prevent this happening. The S NORMAL event statement is used to indicate to the synthesis program which deviations in the model do not require a fault to cause them. The *some* deviation is added to this for flow ports and the *none* deviation for non-flow ports. When this event in encountered by the synthesis algorithm the event is marked as certain and a different deletion process is carried out than for impossible events.

S NORMAL: GmIN SOME, RmIN SOME,...	inlet flow ports
S NORMAL: QnOUT SOME, PnOUT SOME,...	outlet flow ports
S NORMAL: GmIN NONE, RmIN NONE,...	inlet non-flow ports
S NORMAL: QnOUT NONE, PnOUT NONE,...	outlet non-flow ports.

For similar reasons as the S NORMAL event statement the S IMPOSS event statement is used to define which deviations are impossible within the domain of the vessel unit. These prevent deviations propagating through the vessel which have been defined in the methodology or by the user as impossible events. These are reverse flow and pressure through a port where it has been specified as impossible by the user.

S IMPOSS: GmIN REV, RmIN REV,...	inlet ports, no reverse flow
S IMPOSS: QnOUT REV, PnOUT REV,...	outlet ports, no reverse flow.

It should be noted that the S NORMAL and S IMPOSS event statements are only used in models with vessel ports.

This completes the event statements for the vessel model.

7.3.4 Generation Of Decision Tables

This first set of decision table rules are only applied if there is liquid present because they deal with the influences of the liquid level and the causes of its deviations. If there is more than one liquid inlet with flow then a decision table is required to indicate that there must be no flow through each of these ports to cause the intermediate dummy event C(DUMMY). The effects of this are dealt with later. If there are more than three of these ports then the intermediate dummy event D(DUMMY) has to be used because the decision table becomes too long. The decision tables are:

V GmIN NONE ... T C(DUMMY)	more than one liquid inlet flow port

or

V GmIN NONE ... I D(DUMMY) T C(DUMMY)   more than three liquid
        V GmIN NONE ... T D(DUMMY)   inlet flow ports.

The effect of no level in the vessel is obviously no flow or pressure out but it is possible
that flow can occur if there is flow in, so a decision table has to be generated that combines
no flow in and no level. The intermediate dummy event C(DUMMY) combines all the
liquid inlet ports for no flow, therefore, for each liquid outlet flow port we have:

    I C(DUMMY) V LvVES NONE T QnOUT NONE, PnOUT NONE   causes of no
                                                            flow.

The pressure term is included in this because if there is no flow of liquid then there can
be no pressure transmission since this is a liquid port.

The causes of the vessel temperature and composition deviations can be generated by
the propagation equations for the inlet ports because these involve flow in the normal
direction. The outlet ports are more difficult because for flow through these to cause
deviations requires the flow to be reversed. For each outlet port that allows reverse flow
the following set of decision tables is required:

    V QnOUT REV V UnOUT HI T TvVES HI   outlet port with reverse flow
    V QnOUT REV V UnOUT LO T TvVES LO   outlet port with reverse flow
    V QnOUT REV V YnOUT HI T XvVES HI   outlet port with reverse flow
    V QnOUT REV V YnOUT LO T XvVES LO   outlet port with reverse flow.

7.3.5 Additional Information

To complete the model some additional information has to be provided to aid the program
and other users. A facility has been added to generate a limited number of engineering
assumptions and a description of the model. This essentially gives the type of vessel,
open or closed, and what it stores, liquid, gas/vapour or both. Then for each port it gives
the normal flow state, whether it is for liquid or gas/vapour, if this applies, and whether it
is an inlet or outlet port. It also gives the type of flow and any other information such as
reverse flow or a high pressure pump.

The program generates the normal state character string such that it contains a list of
ports which normally have flow.

The generation routine only deals with single component flow or vessels where mass transfer between phases does not occur so the multi-component variable is set to "N" for no.

The model generation is then complete and the routine returns to the main program where the data file is written and the option given to examine it before proceeding to the minitree generation phase. Section 7.5 deals with problems encountered during this phase.

7.4 Vessel Model Template Examples

The vessel model templates have been created to give the user an example of the type of vessel model that can be developed. They give the user the ability to create a number of common configurations by deleting certain elements from the model.

The set of vessel model templates were developed in parallel with the automatic vessel model generation algorithm. They were developed from the rules for the vessel models and then used to test the output from the generation algorithm. The models can be used as a base or an example for developing other vessel models. All the models appear as they would after generation using the algorithm except that the description has details on editing the model.

Each model can be used as it is with the user connecting other units to a proportion of the existing ports and ignoring the rest. Although this procedure will work satisfactorily it is not strictly in keeping with the methodology which states that all ports must be connected to a unit. A solution to this would be to use Dummy Heads and Dummy Tails but this would then alter the configuration diagram. Therefore, within the description of the model is an explanation on to how to remove an unwanted port by giving an example.

The configurations for the models have been chosen in an attempt to show as many different types of vessel and port connections as possible. There is an open vessel, a closed vessel for liquid only, a pressurised gas/vapour vessel, a vacuum gas/vapour vessel and a vessel containing gas/vapour and liquid. A brief description of each model and the model listing is given in Appendix C.

7.4.1 Example Editing Of A Template Model

The open vessel model, number 90 (Appendix C.1), has the following description for editing,

PORTS CAN BE REMOVED IF SO DESIRED BY FOLLOWING THIS EXAMPLE.

IF PORT 2 IS NOT REQUIRED, DELETE PROPAGATION EQUATIONS 5 TO 8. DELETE THE G2IN, T2IN AND X2IN TERMS FROM EQUATIONS 33, 35 AND 36 RESPECTIVELY. DELETE EVENT STATEMENTS 3 AND 4. DELETE THE G2IN NONE, R2IN NONE, G2IN REV AND R2IN REV TERMS FROM EVENT STATEMENTS 16 AND 17. DELETE THE G2IN SOME AND R2IN SOME TERMS FROM EVENT STATEMENT 24. WHEN THE G2IN NONE TERM IS REMOVED FROM THE FIRST DECISION TABLE THIS LEAVES ONE ELEMENT SO DELETE THE DECISION TABLE AND ADD V G1IN NONE: C(DUMMY) TO THE LIST OF EVENT STATEMENTS.

ALTER THE NUMBERS FOR THE NUMBER OF PROPAGATION EQUATIONS, EVENT STATEMENTS AND DECISION TABLES AS NECESSARY AT THE TOP OF THE MODEL.

HAVING CHANGED THE MODEL IT MUST BE GIVEN A NEW FILE NAME AND CORRESPONDING MODEL NUMBER.

Considering port 2 as the redundant port then the above instructions have to be carried out. Propagation equations 5 to 8 are those which have a '2' on the left hand side of the '=' sign, ie. G2IN, R2IN, U2IN and Y2IN. These need to be deleted. The 'G2IN,', 'T2IN,' and 'X2IN,' terms have to be deleted from the L9VES, T9VES and X9VES propagation equations, respectively.

The third and fourth event statements can be deleted because the 'G2IN NONE' and 'G2IN REV' terms no longer exists. The '2' terms in the P9VES HI and S NORMAL state event statements must also be removed. This is the section ',G2IN NONE' and 'G2IN REV,' in the two P9VES HI event statements and the 'G2IN SOME, R2IN SOME,' in the first S NORMAL event statement.

The 'V G2IN NONE' term in the first decision table must be removed but this leaves the decision table with only one term,

    V G1IN NONE T C(DUMMY)

which should not be used as part of the model. It is necessary to convert this decision table into an event statement. It will become

    V G1IN NONE: C(DUMMY)

and must be placed in the list of event statements.

The assumptions, propagation equations, event statements and decision tables should be counted and the correct numbers placed at the end of the 'NO. OF' lines near the top of the model. The normal state line in the supplementary information section should have the '2,' removed from it.

Finally the model number must be changed and the model saved to a file with that number in it, ie. changing the model number to 99 requires the model to be saved as MDAT99.DAT.

The new model is shown in the following pages.

```
1)        MODEL   NUMBER   NAME
             99              VESSEL MODEL (LIQUID, OPEN)


NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 9
NO. OF PROPAGATION EQUATIONS: 32
NO. OF EVENT STATEMENTS: 26
NO. OF DECISION TABLES: 10
NO. OF FAILURE MODES: 1


2)     ENGINEERING ASSUMPTIONS AND DESCRIPTIONS


THIS IS AN OPEN VESSEL USED TO STORE LIQUID, ALL PORTS ARE BELOW THE LEVEL OF THE
LIQUID SURFACE.
PORT 1: NORMAL FLOW INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 3: NO FLOW INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 4: NO FLOW INLET FROM HIGH PRESSURE CENTRIFUGAL PUMP
PORT 5: NORMAL FLOW OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 6: NORMAL FLOW OUTLET UNDER GRAVITY
PORT 7: NO FLOW OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 8: NO FLOW OUTLET UNDER GRAVITY.


3)     PROPAGATION EQUATIONS


G1IN= F (Q1IN)
R1IN= F (-P9VES)
U1IN= F (T9VES)
Y1IN= F (X9VES)
G3IN= F (Q3IN)
R3IN= F (-P9VES)
U3IN= F (T9VES)
Y3IN= F (X9VES)
G4IN= F (Q4IN)
R4IN= F (-P9VES)
U4IN= F (T9VES)
Y4IN= F (X9VES)
```

Q5OUT= F (G5OUT)
P5OUT= F (P9VES)
T5OUT= F (T9VES)
X5OUT= F (X9VES)
Q6OUT= F (G6OUT, P9VES)
P6OUT= F (P9VES)
T6OUT= F (T9VES)
X6OUT= F (X9VES)
Q7OUT= F (G7OUT)
P7OUT= F (P9VES)
T7OUT= F (T9VES)
X7OUT= F (X9VES)
Q8OUT= F (G8OUT, P9VES)
P8OUT= F (P9VES)
T8OUT= F (T9VES)
X8OUT= F (X9VES)
L9VES= F (G1IN, -Q5OUT, -Q6OUT)
P9VES= F (L9VES)
T9VES= F (T1IN, T3IN, T4IN)
X9VES= F (X1IN, X3IN, X4IN)

4)   EVENT STATEMENTS

V G1IN NONE: L9VES LO
V G1IN REV: L9VES LO, L9VES NONE
V G3IN SOME: L9VES HI
V G3IN REV: L9VES LO, L9VES NONE
V G4IN SOME: L9VES HI
V G4IN REV: L9VES LO, L9VES NONE
I C(DUMMY): L9VES NONE
V Q5OUT NONE: L9VES HI
V Q6OUT NONE: L9VES HI
V Q6OUT REV: L9VES HI
V Q7OUT SOME: L9VES LO
V Q8OUT SOME: L9VES LO
V Q8OUT REV: L9VES HI
V P9VES HI: G1IN NONE, R1IN NONE
V P9VES HI: G1IN REV, R1IN REV, G3IN REV, R3IN REV
V P9VES HI: G4IN REV, R4IN REV
V P9VES NONE: Q6OUT REV, P6OUT REV, Q8OUT REV, P8OUT REV
V L9VES NONE: P9VES LO
F LK-LP-EN: L9VES LO, L9VES NONE
F EXT-HEAT: T9VES HI
F EXT-COLD: T9VES LO
S NORMAL: G1IN SOME, R1IN SOME, G3IN NONE, R3IN NONE
S NORMAL: G4IN NONE, R4IN NONE, Q5OUT SOME, P5OUT SOME, Q6OUT SOME, P6OUT SOME
S NORMAL: Q7OUT NONE, P7OUT NONE, Q8OUT NONE, P8OUT NONE

S IMPOSS: Q5OUT REV, P5OUT REV, Q7OUT REV, P7OUT REV

V G1IN NONE: C(DUMMY)

5)   DECISION TABLES

I C(DUMMY) V L9VES NONE T Q5OUT NONE, P5OUT NONE

I C(DUMMY) V L9VES NONE T Q6OUT NONE, P6OUT NONE

V Q6OUT REV V U6OUT HI T T9VES HI

V Q6OUT REV V U6OUT LO T T9VES LO

V Q6OUT REV V Y6OUT HI T X9VES HI

V Q6OUT REV V Y6OUT LO T X9VES LO

V Q8OUT REV V U8OUT HI T T9VES HI

V Q8OUT REV V U8OUT LO T T9VES LO

V Q8OUT REV V Y8OUT HI T X9VES HI

V Q8OUT REV V Y8OUT LO T X9VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 5 AND 6.

NO MULTI-COMPONENT FEATURES

7.5 Special Notes

When the template models are compiled into minitree format by the unit model generation program, the program prompts the user for more information. This happens as a result of the program's built-in minitree generation routines developed by Mullhi (6), which create, from the propagation equations, some minitrees which would otherwise have to be specified as event statements. This system produces questions for the user to answer which requires a degree of knowledge of the methodology but since the generation routine is an aid to the user it is not really appropriate for this to occur. Therefore, all the information has been incorporated into the event statements and decision tables.

The prompts given by the program deal with the validity of a variable deviation as a cause of another deviation. A simple "Y" or "N" answer is required. For each inlet port the following prompts appear:

THE EVENT:	GmIN	REV
CAN BE CAUSED BY:	P9VES	NOR
IS THIS A VALID MINITREE,	Y/N ?	

and

THE EVENT:	RmIN	REV
CAN BE CAUSED BY:	P9VES	NOR
IS THIS A VALID MINITREE,	Y/N ?	

It has been assumed that no relief into the vessel cannot cause a reverse pressure gradient or reverse relief through the port, so these are not valid minitrees. In any case some of these deviations may have been labelled impossible by the model.

The other type of prompt that occurs concerns the outlet ports:

THE EVENT:	L9VES	NONE	
CAN BE CAUSED BY:	QnOUT	HI	liquid ports
IS THIS A VALID MINITREE,	Y/N ?		

This implies that high flow through a single outlet port may cause the liquid level to fall to zero. This situation has been dealt with in the event statements and decision tables because other factors are necessary. A high flow will cause the liquid level to be low but in order for the level to fall to zero there must be no inlet flow either. For this reason the minitree is not required.

The advice given to a user is to reply with "N" because all the information required for the vessel is contained in the model.

7.6 Vessel Model Summary Tables

Table 7.2 - Options	
Vessel type:	Open Closed
Vessel storage fluid :	Liquid Gas/vapour Liquid and gas/vapour
Number of inlet ports :	1 to 7
Number of outlet ports : (NB Total number is 8)	1 to 7
Port flow types :	Normally flow Normally no flow Pressure effects No pressure effects
Reverse flow through ports :	Optional
Multi-component features :	No

Table 7.3 - Faults	
External heat source	F EXT-HEAT
External cold source	F EXT-COLD
Leak to a low pressure environment	F LK-LP-EN
Leak from a high pressure environment	F LK-HP-EN

7.7 References

1. Perry R. H., Chilton C. H.; Chemical Engineers' Handbook, 5th Edition; McGraw-Hill; 1974; pp 6-5–6-32.

2. Perry R. H., Chilton C. H.; Chemical Engineers' Handbook, 5th Edition; McGraw-Hill; 1974; pp 6-85–6-89.

3. Coulson J. M., Richardson J. F.; Chemical Engineering, Volume 1, 3rd Edition; Pergamon; 1977; pp 124–144, pp 147–150.

4. Coulson J. M., Richardson J. F.; Chemical Engineering, Volume 6; Pergamon; 1983; pp 376–380.

5. Hunt A., Lees F. P.; Faultfinder, An Interactive Computer-based Facility For Fault Propagation In Process Plants: Unit Model Library Listing, Version 2 ; Plant Engineering Group, Department of Chemical Engineering, Loughborough University Of Technology; 1990.

6. Mullhi J. S.; Rules For The Computer-Aided Synthesis Of Fault Trees For Process Plants; PhD Thesis, Loughborough University Of Technology; 1990; pp 4/1–4/8.

7. Mullhi J. S.; Rules For The Computer-Aided Synthesis Of Fault Trees For Process Plants; PhD Thesis, Loughborough University Of Technology; 1990; pp 4/11–4/17.

Chapter 8

Rules For The Generation Of Heat Exchanger Models

8.1 Introduction

The previous chapter has dealt with the rules for generating vessel models which involved a process of modelling the internal characteristics as a flow from the inlet or outlet ports to the internal *vessel* port and then back to the inlet or outlet ports. This produced a relatively simple model generation routine.

This chapter deals with the generation of heat exchanger models. While many heat exchangers are simply used to transfer heat from one process fluid to another, some involve phase changes (either complete or partial) such as condensers and reboilers. The simple heat exchangers can be modelled without using internal variables but the ones involving phase changes require internal variables for flow transfer and heat transfer.

8.2 Heat Exchanger Characteristics

Heat transfer equipment covers a wide range of types as given by Perry (1). These cover simple heat exchangers for two process fluids and air-cooled systems to evaporator systems and systems for solids. However some of this equipment is very specialised and does not lend itself to characterisation and automatic model generation. Therefore, the process has been kept to the principal types of heat exchanger found in the process industries. These are the shell and tube type, frame and plate (or plate) type, air-cooled finned tube type and the double pipe type heat exchangers given by Coulson (3).

These four types can be further reduced into two specific categories based on one particular fault. This is made possible because the basic heat transfer fundamentals are similar for all equipment as regard to fault initiation and propagation. The only difference occurs with the blockage faults. In a shell and tube or finned tube heat exchanger it can be assumed very unlikely that the shell side or air side will become blocked or partially blocked to an extent that the flow will be affected. In the plate or double pipe heat exchanger it is possible for both sides to become blocked or partially blocked. This will ultimately affect the heat transfer in the unit. Therefore two types of heat exchanger can be modelled, the shell and tube type and the plate type. The former can only be blocked on the tube side whereas the latter can be blocked on both sides. This is shown in Figure 8.1.

Figure 8.1 - Heat Exchanger Types
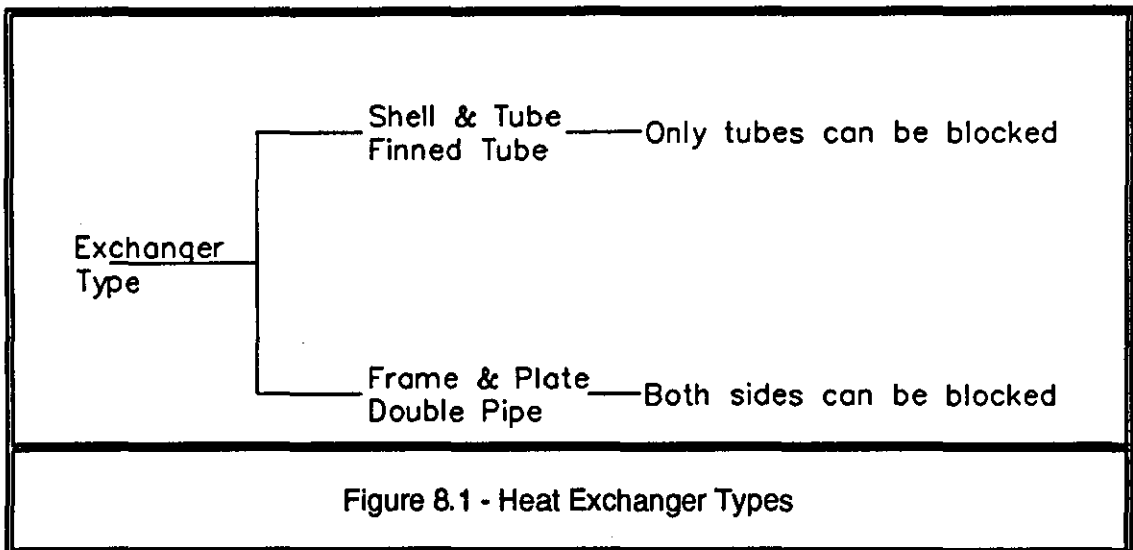
Different designs of heat exchanger mean there is a variety of ways in which the two process streams can flow relative to each other. There is counter-current flow, co-current flow, double pass, triple pass, etc. However the basic heat transfer equation gives the amount of heat transferred as proportional to the area and the temperature difference. In the above heat exchanger types they can all be given the same area by design and Coulson (2) shows the temperature difference expressions to be the same. It should be noted that although the expression is the same the value will depend on the inlet and outlet temperatures and the difference between the two streams at the exit points, resulting in a difference in the quantity of heat transferred. However, for the purpose of modelling we are only interested in the effects of changes in the flow and the temperature variables which will be the same. Therefore no distinction is necessary for the flow type.

The propagation equations for the temperature at the ports of a simple heat exchanger were derived by Kelly (4). This showed the expression for the first side containing the cold fluid as

T2OUT= F (-G1IN, T1IN, G3IN, T3IN)

which indicates the first side outlet temperature to be proportional to the first side inlet temperature and the second side inlet flow and temperature, and inversely proportional to the first side inlet flow.

If the first side contains the hot fluid then the expression is,

T2OUT= F (G1IN, T1IN, -G3IN, T3IN)

giving the first side outlet temperature as proportional to the first side inlet flow and temperature and second side inlet temperature, and inversely proportional to the second side inlet flow.

Therefore it is necessary to label the first side as hot or cold.

In order to model any leaks which may occur internally it is necessary to identify the higher pressure side such that the direction of flow of fluid from one side to the other can be determined. This allows the effects of contamination of one stream by the other to be modelled. If the pressure difference is not significant, then the internal leak fault is not modelled.

Heat exchangers also include reboilers and condensers where one or both of the streams undergo a phase change. Phase changes can be modelled relatively easily but they require the use of additional ports called vessel ports to model the internal characteristics of the flow and heat transfer. There is a restriction imposed by FAULTFINDER of a maximum of nine ports in a unit. In addition to the four flow ports (or five in the case of a partial phase change), a complete phase change has two vessel ports and a partial phase change has three vessel ports. For this reason the phase change is only considered on one side, which has to be specified, and any change on the other is ignored as not important. This may create problems if there is a change on both sides but when tracing fault propagation the top event of interest will only ever be concerned with one of the phase changes. If another top event is then concerned with the other phase change a new model can easily and quickly be generated and the data file altered to use this model. This restriction has been identified as an area for improvement for which the duration of this project has not allowed.

If there is a complete phase change then the composition of the end phase will be the same as the initial phase, however, in a partial phase change not all the fluid will change phase. If there are components of different volatility then different quantities of each will change phase. Therefore, for a partial phase change, it is necessary to define the system as multi-component if there is more than one component as shown in Figure 8.2.

To enable the flow to be modelled through each port it is necessary to define the type of fluid transportation method. There are a variety of types but essentially they break down into categories of pressure and reverse flow effects. The pressure effects are due to the vessel pressure influences on the flow and the reverse flow capability depends on the type of pump connected to the port. The assumptions used are based on those applied for the vessel model generation rules (see Section 7.2.2).

```
                                   ┌──Complete────── Single component
                   ┌──Yes──────────┤
                   │               │            ┌── Single component
Phase              │               └──Partial───┤
Change  ───────────┤                            └── Multi-component
                   │
                   │
                   └────── No ──────────None──────────── Single component
```
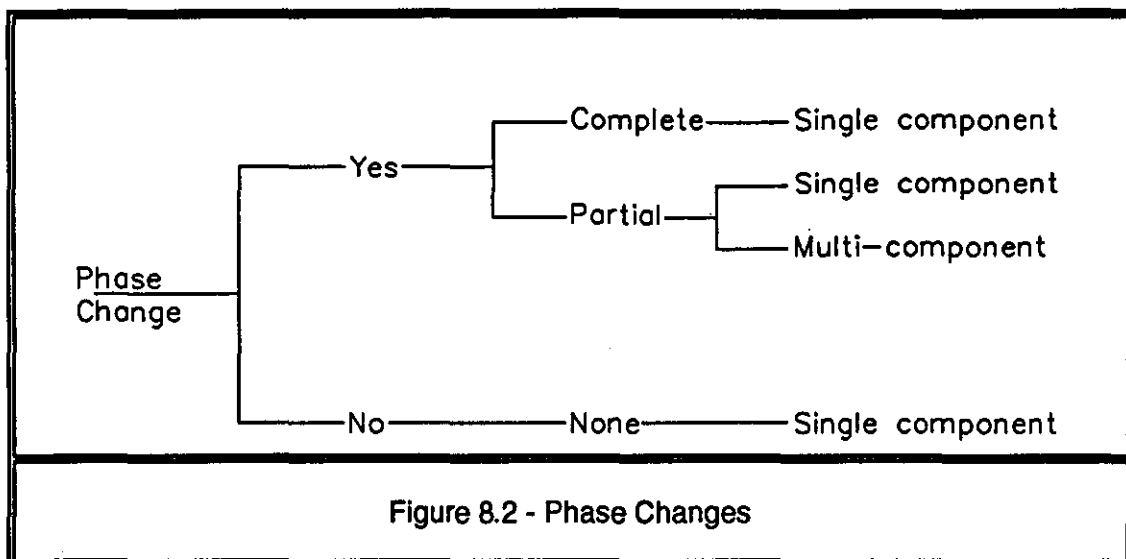
Figure 8.2 - Phase Changes

For the no phase change side the flow is assumed to behave similarly to that in a simple pipe type model. In these situations the pressure is simply transmitted through the unit and the flow is not affected by the vessel pressure. There is an option to indicate if reverse flow can occur through the ports.

If a phase change occurs then changes in the vessel pressure are likely. For this reason four port types have been considered for an inlet and three for an outlet port. An inlet port defined as a gravity/back-pressure fed flow port is assumed to be affected by the vessel pressure. The same is assumed for an inlet port which is defined as connected to a low pressure centrifugal pump which has a pump pressure close to the vessel pressure. An inlet port defined as connected to a high pressure centrifugal pump or a reciprocating pump is not affected by the vessel pressure. All ports are assumed to allow reverse flow unless the user defines a reciprocating pump where an option is given to suppress the reverse flow deviation. Outlet ports are split into gravity/back-pressure flows and pumped flows. Gravity/back-pressure flows are assumed to be affected by the vessel pressure and pumped flows are not. The pumped flows are divided into centrifugal and reciprocating pumps with the latter having the ability to suppress reverse flow.

On some heat exchanger equipment relief ports and drain ports are used for safety and maintenance reasons so these have been integrated into the layout should they be required. These are simply expressed as non-flow ports and are modelled as being affected by the vessel pressure and allowing reverse flow.

The information required for each port is summarised in Table 8.1.

Table 8.1 - Port Information			
Port Type	Pressure Effects	Reverse Flow	Mode used
No phase change inlet	No	Yes	Normally flow
No phase change inlet	No	No	Normally flow
No phase change outlet	No	Yes	Normally flow
No phase change outlet	No	No	Normally flow
Phase change inlet	Yes	Yes	Gravity/back-pressure
Phase change inlet	Yes	Yes	Centrifugal (low)
Phase change inlet	No	Yes	Centrifugal (High)
Phase change inlet	No	Yes	Reciprocating
Phase change inlet	No	No	Reciprocating
Phase change outlet	Yes	Yes	Gravity/back-pressure
Phase change outlet	No	Yes	Centrifugal
Phase change outlet	No	Yes	Reciprocating
Phase change outlet	No	No	Reciprocating
Relief	Yes	Yes	Normally no flow
Drain	Yes	Yes	Normally no flow

These are the only characteristics required for generating heat exchanger models.

8.3 Heat Exchanger Model Generation Program

The heat exchanger model generation subroutine has been written into the unit model generation program MODGEN. The option to use this facility is given after the model number and name have been entered. The information required by the subroutine is collected by a set of questions with optional answers given to the user. The questions asked depend on previous options selected and, therefore, the user interaction is kept to a minimum. Most of the information can be found on a piping and instrumentation diagram although some such as the fluid states may need to be calculated from the temperature and pressure data.

The generation process is split into four subroutines, a master subroutine collects the data, generates a description of the model and then calls the other subroutines. There is one to generate the propagation equations, one for the event statements and one for the decision tables.

As well as generating all the propagation equations, event statements and decision tables the routine generates the list of engineering assumptions and description and prints it on the screen for the user to check. This information is saved with the model and can be used as a reference when looking for a suitable model.

8.3.1 Data Collection

The generation subroutine has been developed to deal with two types of process heat exchanger, one which can be blocked on only one side, such as the tube side of a shell and tube heat exchanger and one which can be blocked on both sides such as in a plate heat exchanger. The type required must be identified as one of these. Should the heat exchanger be of another type the model must be specified as being of the type with the appropriate blockage properties. The fault propagation models are the same for all types of heat exchanger.

The restriction imposed by FAULTFINDER on the maximum number of ports in a unit means the user must specify if there is a phase change or not. Any phase change must be confined to one side, which must be identified, and specified as a complete or partial phase change. The side containing the phase change is hereafter referred to as the second side because it is dealt with after the no phase change side.

If a partial phase change occurs then the user has to specify if there is more than one component in that particular side. This is necessary because different components have

different rates of condensing or boiling and, therefore, the liquid or vapour produced will have a different composition. If this is specified then multi-component modelling must be used.

There are two sets of temperature propagation equations, one for the hot fluid and one for the cold fluid. Therefore the first side must be specified as containing the hot or the cold fluid. There is no difference between the flow regimes of counter-current, co-current, double-pass, etc.

It is necessary to indicate if there is a significant pressure difference between the two sides and if so in which side the higher pressure is located. This is used to model internal leaks.

If there are sufficient ports remaining, a relief port and/or drain port can be specified for each side. These are defined as normally having no flow.

Assumptions have been made about the flows to and from the heat exchanger. For no phase change the vessel pressure does not affect the flow and the behaviour is the same as in a simple pipe. This is standard for the first side where it is defined as having no phase change. If a phase change occurs on the second side then the user is prompted for the type of flow which occurs through the port. This may or may not be affected by the vessel pressure. Table 8.1 shows the flow type combinations and the corresponding method of transportation. The ports can be defined as being under gravity flow, low pressure or high pressure centrifugal pumped flow or reciprocating pumped flow.

Reverse flow is allowed in all flow ports but the user can change this by choosing the no reverse flow option which is given by the program.

This is all the information required by the routine and the model can now be generated using the rules incorporated in the routine.

8.3.2 Model Ports And Nomenclature

The layout of the models has been standardised as much as possible to reduce any confusion. Port 1 is always the first side inlet stream and port 2 the outlet stream. Any additional ports on the first side, such as the relief or drain ports, are allocated the next port or ports as necessary otherwise they are used by the second side. Hence port 3 may be the relief or drain port for the first side or the inlet for the second side. Therefore

port 4 may be the first side drain port, if there is also a relief port, or the second side inlet, if there is only one additional port on the first side, or the outlet on the second side, if there are no additional first side ports.

As can be seen the logic here gets quite complicated and some variables have been set up to overcome this. FLAG (3) is set to 0, 1 or 2 for the number of additional ports on the first side and if this is 1 then FLAG (4) is set to 1 for a relief or 2 for a drain port. Similar variables have been set up for the second side. It is also necessary to define a complete set of character variables which contain the port numbers of the second side ports used. The port numbers are allocated in the following order if they are used,

PORCH (1)	Inlet
PORCH (2)	Outlet or, if there is a partial phase change, liquid outlet
PORCH (3)	Vapour outlet for a partial phase change
PORCH (4)	Relief if there is one
PORCH (5)	Drain if there is one

The following are used for the vessel ports if a phase change occurs,

VESCH (1)	Vessel port or, for a partial phase change, vessel port relating to the liquid
VESCH (2)	Vessel port relating to the vapour for a partial phase change
VESCH (3)	Vessel port relating to the internal flow from liquid to vapour or vapour to liquid.

The last vessel port is used to model the internal flow of fluid from one phase to the other. For a reboiler the port is used to model the flow from the liquid to the vapour and for a condenser it is used to model the flow from the vapour to the liquid. This means the flow will always be positive and the model has been defined such that reverse flow in this port is impossible.

The following sections on the modelling of the heat exchanger uses the following nomenclature. The sides of the exchanger have been defined as:

first side	no phase change
second side	phase change if required.

If no phase change occurs then the two sides will be similar and the order will not matter.

After each rule or set of rules is defined a list of example propagation equations, event statements or decision tables is given. These use a set of keywords to describe the type of heat exchanger to which they apply. The terms are:

none	for no phase change
complete	for a complete phase change
partial	for a partial phase change
hot	current side contains the hot fluid
cold	current side contains the cold fluid
single	no multi-component features used
multi	multi-component features are used
reverse	reverse flow is possible through the port.

The 'hot' and 'cold' keywords refer to the *current* side of the heat exchanger in which the deviation or fault occurs. This is easy to determine for deviations such as 'G1IN=' (first side propagation equation) and 'L5VES HI:' (second side event statement) but for basic faults it is not as obvious. For basic faults such as 'INT-LK' the side referred to is given by the heading prior to the list of event statements. For the 'PART-BLK', 'FOULING', 'FROTHING' and 'VAP-BLKT' faults there are no headings and reference is made to the first side.

The example propagation equations, event statements and decision tables are based on three types of model which have the following port types:

No phase change
    port 1   first side inlet
    port 2   first side outlet
    port 3   second side inlet
    port 4   second side outlet.

Complete phase change
    port 1   first side inlet
    port 2   first side outlet
    port 3   second side inlet
    port 4   second side outlet
    port 5   vessel port for variables
    port 6   vessel port for flow.

Partial phase change
    port 1   first side inlet
    port 2   first side outlet

port 3	second side inlet
port 4	second side liquid outlet
port 5	second side vapour outlet
port 6	vessel port for liquid
port 7	vessel port for vapour
port 8	vessel port for flow.

These models are not the same as the illustrative example models given in Appendix D. For simplicity the above models have no relief or drain ports but the example models do.

There are rules given for relief and drain ports and in these cases one port has been added to the side of interest and is noted in the example.

8.3.3 Generation Of Propagation Equations

This section deals with the rules used to develop the propagation equations. Each port is taken in turn and the equation for the flow, pressure, temperature and composition defined as necessary.

The flow and relief propagation equations for the inlet port on the first side are taken directly from the standard pipe model. These are:

G1IN= F (Q1IN, Q2OUT)	flow
R1IN= F (R2OUT)	relief

The temperature propagation equation for the inlet port on the first side, if reverse flow can occur, depends on whether the first side is for the hot or cold fluid. The equation is a function of the inlet flow on the second side, the reverse temperature on the first side and the inlet temperature on the second side. However, if a phase change occurs on the second side the vessel port temperature is used instead of the second side inlet temperature. The matter is further complicated if the phase change is only partial because there are two vessel port temperatures, one for the liquid and one for the vapour. The vapour vessel port temperature is used if the first side is for the cold fluid and the liquid vessel port is used if it is the hot fluid. This is because generally the exit port for the first side on a cooling system will be in the vapour space, whereas for a heating system it will be in the liquid space. In fact the question of which to use is not important because the two temperatures will be very similar anyway and the deviation from the normal will be in the same direction. The equation will have one of the following forms:

U1IN= F (U2OUT, -G3IN, T3IN)   none, hot, reverse
U1IN= F (U2OUT, G3IN, T3IN)   none, cold, reverse
U1IN= F (U2OUT, -G3IN, T5VES)   complete, hot, reverse
U1IN= F (U2OUT, G3IN, T5VES)   complete, cold, reverse
U1IN= F (U2OUT, -G3IN, T6VES)   partial, hot, reverse
U1IN= F (U2OUT, G3IN, T7VES)   partial, cold, reverse.

The composition term, if reverse flow can occur, is a straightforward function of the reverse exit composition, if this is also possible. If the multi-component feature is used then the format has to be altered slightly by incorporating a 'C' which refers to the component (or component of interest) in the first side.

Y1IN= F (Y2OUT)   single, reverse
YC1IN= F (YC2OUT)   multi, reverse.

The flow and pressure equations for the outlet port on the first side are also taken directly from the standard pipe model. These are:

Q2OUT= F (G1IN, G2OUT)   flow
P2OUT= F (P1IN)   pressure.

The form of the equation used for the temperature at the outlet port is similar to those for the reverse temperature term. The equation is a function of the inlet flows and temperatures on both sides with similar adaptations for the phase changes. The equation will have one of the following forms:

T2OUT= F (G1IN, T1IN, -G3IN, T3IN)   none, hot
T2OUT= F (-G1IN, T1IN, G3IN, T3IN)   none, cold
T2OUT= F (G1IN, T1IN, -G3IN, T5VES)   complete, hot
T2OUT= F (-G1IN, T1IN, G3IN, T5VES)   complete, cold
T2OUT= F (G1IN, T1IN, -G3IN, T6VES)   partial, hot
T2OUT= F (-G1IN, T1IN, G3IN, T7VES)   partial, cold.

The composition term is a straightforward function of the inlet composition.

X2OUT= F (X1IN)   single
XC2OUT= F (XC1IN)   multi.

The equations for any additional ports are the same for both the relief and the drain port. Since there are no vessel ports on the first side the internal pressure has been defined as the pressure at the inlet port. The flow equation for a relief or drain port incorporates

pressure effects and the pressure equation is a straight-forward function of the inlet pressure. In the following two equations, used to illustrate the use of additional ports, port 3 has been defined as a relief or drain port with port 1 remaining the inlet port.

  Q3OUT= F (G3OUT, P1IN)   relief or drain flow
  P3OUT= F (P1IN)   relief or drain pressure.

The temperature and composition propagation equations are similar to those for the outlet port. Again, port 3 has been defined as a relief or drain port with port 1 remaining the inlet port. Each of the second side port numbers are increased by one to allow for the additional port on the first side.

  T3OUT= F (G1IN, T1IN, -G4IN, T4IN) none, hot
  T3OUT= F (-G1IN, T1IN, G4IN, T4IN) none, cold
  T3OUT= F (G1IN, T1IN, -G4IN, T6VES) complete, hot
  T3OUT= F (-G1IN, T1IN, G4IN, T6VES) complete, cold
  T3OUT= F (G1IN, T1IN, -G4IN, T7VES) partial, hot
  T3OUT= F (-G1IN, T1IN, G4IN, T8VES) partial, cold
  X3OUT= F (X1IN)   single
  XC3OUT= F (XC1IN)   multi.

The first side equations are not complex or numerous because it has been defined as having no phase change. The second side equations will be very similar if no phase change takes place except that the port numbers will be different. Explanation, therefore, is limited to phase change situations only but the equations have still been given in the lists.

The flow equation for the second side inlet port depends on whether pressure effects are to be considered. This is defined by the choice of transportation mode used in Table 8.1.

  G3IN= F (Q3IN, Q4OUT)   none
  G3IN= F (Q3IN)   complete or partial, no pressure effects
  G3IN= F (Q3IN, -P5VES)   complete, pressure effects
  G3IN= F (Q3IN, -P7VES)   partial, pressure effects.

Note the use of different vessel ports for the effective vessel pressure between a complete phase change and a partial phase change. This is because there is an extra flow port and the vapour vessel port is the second vessel port (the first being the liquid vessel port) for a partial phase change.

The relief equation for the second side inlet port is a function of the pressure at the appropriate vessel port.

R3IN= F (R4OUT)   none
R3IN= F (-P5VES)   complete
R3IN= F (-P7VES)   partial.

The temperature equation, if reverse flow is possible, is a function of the temperature at the appropriate vessel port.

U3IN= F (G1IN, T1IN, U4OUT)   none, cold, reverse
U3IN= F (-G1IN, T1IN, U4OUT)   none, hot, reverse
U3IN= F (T5VES)   complete, reverse
U3IN= F (T6VES)   partial, cold, reverse
U3IN= F (T7VES)   partial, hot, reverse.

The composition equation for the second side inlet port, if reverse flow is possible, is a function of the composition at the appropriate vessel port. If there is more than one component in the second side then the multi-component feature must be used. Two equations are required and the 'A' refers to the more volatile component which is modelled in the vapour phase and 'B' is that remaining and is modelled in the liquid phase.

Y3IN= F (Y4OUT)   none, reverse
Y3IN= F (X5VES)   complete, reverse
Y3IN= F (X6VES)   partial, single, cold, reverse
Y3IN= F (X7VES)   partial, single, hot, reverse
YA3IN= F (-XB6VES)   partial, multi, cold, reverse
YB3IN= F (XB6VES)   partial, multi, cold, reverse
YA3IN= F (XA7VES)   partial, multi, hot, reverse
YB3IN= F (-XA7VES)   partial, multi, hot, reverse.

For no phase change or a complete phase change there is only one outlet flow port but for a partial phase change one is required for the liquid outlet and one for the vapour outlet. The flow equation for the second side outlet or liquid outlet port contains a pressure term if pressure effects are to be considered.

Q4OUT= F (G3IN, G4OUT)   none
Q4OUT= F (G4OUT)   complete or partial, no pressure effects
Q4OUT= F (G4OUT, P5VES)   complete, pressure effects
Q4OUT= F (G4OUT, P7VES)   partial, pressure effects.

The pressure equation for the second side outlet or liquid outlet port is a function of the pressure at the appropriate vessel port.

```
P4OUT= F (P3IN)                          none
P4OUT= F (P5VES)                         complete
P4OUT= F (P7VES)                         partial.
```

The temperature equation for the second side outlet or liquid outlet port is a function of the temperature at the appropriate vessel port.

```
T4OUT= F (G1IN, T1IN, -G3IN, T3IN)       none, cold
T4OUT= F (-G1IN, T1IN, G3IN, T3IN)       none, hot
T4OUT= F (T5VES)                         complete
T4OUT= F (T6VES)                         partial.
```

The composition equation for the second side outlet or liquid outlet port is a function of the composition at the appropriate vessel port.

```
X4OUT= F (X3IN)                          none
X4OUT= F (X5VES)                         complete
X4OUT= F (X6VES)                         partial, single
XA4OUT= F (-XB6VES)                       partial, multi
XB4OUT= F (XB6VES)                        partial, multi.
```

For a partial phase change a second outlet port is required for the vapour. This is modelled in the same way as the other second side outlet port except that the partial phase change options and the vapour vessel port are used. The equations are:

```
Q5OUT= F (G5OUT)                         partial, no pressure effects, flow
Q5OUT= F (G5OUT, P7VES)                  partial, pressure effects, flow
P5OUT= F (P7VES)                         partial, pressure
T5OUT= F (T7VES)                         partial, temperature
X5OUT= F (X7VES)                         partial, single, composition
XA5OUT= F (XA7VES)                        partial, multi, composition
XB5OUT= F (-XA7VES)                       partial, multi, composition.
```

The flow and pressure equations for any additional ports are similar to the additional ports on the first side except that the vessel port terms for the pressure are used for phase change situations. In these examples there is only one additional port and that is on the

8-14

second side. For no phase change or a complete phase change the additional port is port 5 but for a partial phase change it is port 6. It may be either a relief or drain port because the results are the same.

Q5OUT= F (G5OUT, P3IN)	none
Q5OUT= F (G5OUT, P6VES)	complete
Q6OUT= F (G6OUT, P8VES)	partial
P5OUT= F (P3IN)	none
P5OUT= F (P6VES)	complete
P6OUT= F (P8VES)	partial

The temperature and composition propagation equations are similar to those for the outlet port. Again, the same port numbers are used for the relief or drain port. It should be noted that there is a difference between a relief port and a drain port, with one being for the vapour and the other for the liquid, respectively. The appropriate vessel ports must be used.

T5OUT= F (G1IN, T1IN, -G3IN, T3IN)	none, cold
T5OUT= F (-G1IN, T1IN, G3IN, T3IN)	none, hot
T5OUT= F (T6VES)	complete
T6OUT= F (T7VES)	partial, drain
T6OUT= F (T8VES)	partial, relief
X5OUT= F (X3IN)	none
X5OUT= F (X6VES)	complete
X5OUT= F (X7VES)	partial, single, drain
X5OUT= F (X8VES)	partial, single, relief
XA6OUT= F (-XB7VES)	partial, multi, drain
XB6OUT= F (XB7VES)	partial, multi, drain
XA6OUT= F (XA8VES)	partial, multi, relief
XB6OUT= F (-XA8VES)	partial, multi, relief.

The last sets of equations deal with the vessel ports and are only necessary if there is a phase change. The first vessel port is used either to model all the variables for a complete phase change or just the variables associated with the liquid side for a partial phase change.

The level term is generated from a mass balance of all the liquid streams entering and leaving the second side. The actual terms depend on whether the second side is the hot or cold side. The complete selection of terms is; liquid inlet flow, G3IN, liquid outlet flow, Q4OUT, and the internal flow, QvVES. This becomes,

L5VES= F (G3IN, -Q6VES)   complete, cold
L5VES= F (-Q4OUT, Q6VES)   complete, hot
L6VES= F (G3IN, -Q4OUT, -Q8VES)   partial, cold
L6VES= F (-Q4OUT, Q8VES)   partial, hot.

The pressure term is used at this port only if there is a complete phase change and is a mass balance on the vapour streams entering and leaving the second side. The possible terms are; vapour inlet flow, G3IN, vapour outlet flow, Q4OUT, and the internal flow, Q6VES. The level term, L5VES, is also used in this because as the level rises so will the pressure and the head of liquid above the port will also exert a pressure.

P5VES= F (-Q4OUT, L5VES, Q6VES)   complete, cold
P5VES= F (G3IN, L5VES, -Q6VES)   complete, hot

The temperature term has been developed from those derived by Kelly (5). For a single component system it is a function of the pressure and the internal flow. For a system with multi-component features the pressure and composition of the liquid are used.

T5VES= F (P5VES, -Q6VES)   complete
T6VES= F (P7VES, -Q8VES)   partial, single
T6VES= F (P7VES, XB6VES)   partial, multi.

The composition term again comes from those developed by Kelly. For a single component system it is a function of the inlet composition. A system with multi-component features is more complicated and since this is the liquid outlet port the equation makes reference to the less volatile component or components. For a cold fluid it is a function of the inlet composition and the internal flow but for a hot fluid it is a function of the composition at the vapour vessel port (since only the less volatile component is modelled at this port, a negative relationship is used) and the internal flow.

X5VES= F (X3IN)   complete
X6VES= F (X3IN)   partial, single
XB6VES= F (XB3IN, Q8VES)   partial, multi, cold
XB6VES= F (-XA7VES, -Q8VES)   partial, multi, hot.

A second vessel port is required for a partial phase change to model the vapour phase. The pressure term is a mass balance of the vapour flows into and out of the second side. The possible terms are; vapour inlet flow, G3IN, vapour outlet flow, Q4OUT, the internal flow, Q8VES, and the level term, L6VES.

    P7VES= F (-Q5OUT, L6VES, Q8VES)   partial, cold
    P7VES= F (G3IN, -Q5OUT, L6VES, -Q8VES)   partial, hot.

The temperature for the second vessel port as derived by Kelly is a function of the pressure and internal flow for single components and a function of the pressure and the composition for multi-component features.

    T7VES= F (P7VES, -Q87VES)   partial, single
    T7VES= F (P7VES, -XA7VES)   partial, multi.

The composition term for the second vessel port depends on the number of components and whether the second side is hot or cold. For single components only the inlet composition is used. For multi-component features the inlet composition and internal flow are used for the hot side and the composition at the liquid vessel port and internal flow are used for the cold side.

    X7VES= F (X3IN)   partial, single
    XA7VES= F (-XB6VES, -Q8VES)   partial, multi, cold
    XA7VES= F (XA3IN, Q8VES)   partial, multi, hot.

The last vessel port is used to model the internal flow between the liquid and vapour phases. The equation depends on whether it is for the hot or cold side and is a function of the flow and the temperature into the first side, the composition (if a partial phase change occurs) and the pressure.

    Q6VES= F (G1IN, T1IN, -P5VES)   complete, cold
    Q6VES= F (G1IN, -T1IN, P5VES)   complete, hot
    Q8VES= F (G1IN, T1IN, -P7VES)   partial, single, cold
    Q8VES= F (G1IN, -T1IN, P7VES)   partial, single, hot
    Q8VES= F (G1IN, T1IN, -XB6VES, -P7VES)   partial, multi, cold
    Q8VES= F (G1IN, -T1IN, -XA7VES, P7VES)   partial, multi, hot.

This completes all the necessary propagation equations for the heat exchanger model.

8.3.4 Generation Of Event Statements

For the event statements each deviation or event is considered in turn and applied to each appropriate port.

A high flow through the second side liquid outlet port may cause the level to fall to zero for a partial phase change. A high flow through the second side outlet port of a hot system with a complete phase change may also cause the level to fall to zero.

> V Q4OUT HI: L5VES NONE   complete, hot
> V Q4OUT HI: L6VES NONE   partial.

The effect of some flow out through any relief or drain port has to be considered. This increases the inlet flow and reduces the outlet flow and even reverses it if this can occur through the outlet. For the second side where a phase change occurs the effects are on the pressure for a relief port and on the level and the pressure for a drain port. For the first side port 3 is the additional port and for the second side it is port 5 (or 6 for a partial phase change).

> First side

> V Q3OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV,
>   R1IN HI, P2OUT LO, P2OUT NONE, P2OUT REV

> Second side

> V Q5OUT SOME: G3IN HI, Q4OUT LO, Q4OUT NONE, Q4OUT REV,
>   R3IN HI, P4OUT LO, P4OUT NONE, P4OUT REV   none
> V Q5OUT SOME: P6VES LO   complete, relief
> V Q5OUT SOME: L6VES LO, L6VES NONE, P6VES LO   complete, drain
> V Q6OUT SOME: P8VES LO   partial, relief
> V Q6OUT SOME: L7VES LO, L7VES NONE, P8VES LO   partial, drain.

The results of reverse flow through the ports for a no phase change situation are modelled by the propagation equations. However, for a phase change situation on the second side where vessel ports are involved the process must be handled the same as that in vessel models (see section 7.3.3). For the inlet ports, reverse flow may cause low or no level and/or low pressure depending on whether the second side was the hot or cold side.

V G3IN REV: L5VES LO, L5VES NONE, P5VES LO   complete, cold
V G3IN REV: P5VES LO   complete, hot
V G3IN REV: L6VES LO, L6VES NONE, P7VES LO   partial, cold
V G3IN REV: P7VES LO   partial, hot.

For the outlet ports, reverse flow may cause high level and/or pressure.

V Q4OUT REV: P5VES HI   complete, cold
V Q4OUT REV: L5VES HI, P5VES HI   complete, hot
V Q4OUT REV: L6VES HI, P7VES HI   partial
V Q5OUT REV: P7VES HI   partial.

The effect of reverse flow through any relief or drain port has to be considered. This decreases the inlet flow (it may even reverse it) and increases the outlet flow. For the second side where a phase change occurs the effects are on the pressure for a relief port and on the level and the pressure for a drain port. Impurities may also be introduced from these ports. For the first side port 3 is the additional port and for the second side it is port 5 (or 6 for a partial phase change). The component 'D' refers to an impurity entering through any of the additional ports, for example air.

First side

V Q3OUT REV: G1IN LO, G1IN NONE, G1IN REV, Q2OUT HI
             R1IN LO, R1IN NONE, R1IN REV, P2OUT HI
             Y1IN HI, X2OUT HI   single
V Q3OUT REV: G1IN LO, G1IN NONE, G1IN REV, Q2OUT HI
             R1IN LO, R1IN NONE, R1IN REV, P2OUT HI
             YD1IN HI, XD2OUT HI   multi.

Second side

V Q5OUT REV: G3IN LO, G3IN NONE, G3IN REV, Q4OUT HI
             R3IN LO, R3IN NONE, R3IN REV, P4OUT HI
             Y3IN HI, X4OUT HI   none
V Q5OUT REV: P6VES HI, X6VES HI   complete, relief
V Q5OUT REV: L6VES HI, P6VES HI, X6VES HI   complete, drain
V Q6OUT REV: P8VES HI, X8VES HI   partial, relief, single
V Q6OUT REV: L7VES HI, P8VES HI, X7VES HI   partial, drain, single
V Q6OUT REV: P8VES HI, XD8VES HI   partial, relief, multi
V Q6OUT REV: L7VES HI, P8VES HI, XD7VES HI   partial, drain, multi.

If a phase change occurs then the various deviations of the level term may have different effects on other variables. These also depend on whether the second side is the hot or cold side. The effects are in the outlet temperature of the first side and the internal flow.

V L5VES HI: T2OUT HI, Q6VES LO   complete, hot
V L5VES HI: T2OUT LO, Q6VES HI   complete, cold
V L6VES HI: T2OUT HI, Q8VES LO   partial, hot
V L6VES HI: T2OUT LO, Q8VES HI   partial, cold

V L5VES LO: T2OUT LO, Q6VES HI   complete, hot
V L5VES LO: T2OUT HI, Q6VES LO   complete, cold
V L6VES LO: T2OUT LO, Q8VES HI   partial, hot
V L6VES LO: T2OUT HI, Q8VES LO   partial, cold

V L5VES NONE: T2OUT LO, P5VES LO, Q6VES HI   complete, hot
V L5VES NONE: T2OUT HI, T5VES HI, P5VES LO,
                Q6VES NONE   complete, cold
V L6VES NONE: T2OUT LO, P7VES LO, Q8VES HI   partial, hot
V L6VES NONE: T2OUT HI, T7VES HI, P7VES LO,
                Q8VES NONE   partial, cold.

If there is a phase change then the vessel pressure may affect the flows into and out of the second side. The reverse flow deviation is only used if the particular port allows reverse flow.

V P5VES HI: G3IN NONE, R3IN NONE, G3IN REV, R3IN REV   complete
V P7VES HI: G3IN NONE, R3IN NONE, G3IN REV, R3IN REV   partial

V P5VES NONE: Q4OUT REV, P4OUT REV   complete
V P7VES NONE: Q4OUT REV, P4OUT REV,
                Q5OUT REV, P5OUT REV   partial

V P6VES NONE: Q5OUT REV, P5OUT REV   complete, relief or drain port
V P8VES NONE: Q6OUT REV, P6OUT REV   partial, relief or drain port.

The effects of no internal flow have also to be considered for phase changes. This affects the level and the pressure depending on whether the second side is the hot or cold side.

```
V Q6VES NONE: L5VES LO, P5VES HI        complete, hot
V Q6VES NONE: L5VES HI, P5VES LO        complete, cold
V Q8VES NONE: L6VES LO, P7VES HI        partial, hot
V Q8VES NONE: L6VES HI, P7VES LO        partial, cold.
```

These are all the variable deviations that are considered, the rest of the event statements are used to incorporate the basic faults.

The external heat source or cold source faults affect the inlet and outlet temperatures on both sides and the vessel port temperatures for a phase change. The affect on the pressure is also considered. In the examples all ports have reverse flow.

```
F EXT-HEAT: U1IN HI, T2OUT HI, U3IN HI, T4OUT HI            none
F EXT-HEAT: U1IN HI, T2OUT HI, T5VES HI, P5VES HI           complete
F EXT-HEAT: U1IN HI, T2OUT HI, T6VES HI, P7VES HI,
            T7VES HI                                        partial


F EXT-COLD: U1IN LO, T2OUT LO, U3IN LO, T4OUT LO            none
F EXT-COLD: U1IN LO, T2OUT LO, T5VES LO, P5VES LO           complete
F EXT-COLD: U1IN LO, T2OUT LO, T6VES LO, P7VES LO,
            T7VES LO                                        partial.
```

A leak to a low pressure environment deals with leaks to the atmosphere and it is assumed that a leak from only one side of the heat exchanger occurs. The leak will affect the inlet and outlet flows or the vessel ports, including reverse flow if this is allowed.

First side

```
F LK-LP-EN: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV,
            R1IN HI, P2OUT LO, P2OUT NONE, P2OUT REV.
```

Second side

```
F LK-LP-EN: G3IN HI, Q4OUT LO, Q4OUT NONE, Q4OUT REV,
            R3IN HI, P4OUT LO, P4OUT NONE, P4OUT REV      none
F LK-LP-EN: L5VES LO, L5VES NONE, P5VES LO                complete
F LK-LP-EN: L6VES LO, L6VES NONE, P7VES LO                partial.
```

Internal leaks are assumed only to occur if there is a significant pressure difference between the two sides of the heat exchanger. From the high pressure side there is increased flow in but reduced flow out or even reverse flow. The lower pressure side has reduced and possibly reversed flow in but a high flow out. Temperature effects are considered and depend on the *current* side indicated being hot or cold and the direction of flow, for example, a hot fluid entering the cold side will cause the cold side exit temperature to be high and the other sides to be low. Composition is also considered as one side being an impurity in the other. In the example event statements the *current* side is the high pressure side. The effects shown are for flow and pressure on the first side then flow and pressure on the second side followed by the effects on temperature and composition.

High pressure first side

F INT-LK: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV,
         R1IN HI, P2OUT LO, P2OUT NONE, P2OUT REV

F INT-LK: G3IN LO, G3IN NONE, G3IN REV, Q4OUT HI,
         R3IN LO, R3IN NONE, R3IN REV, P4OUT HI   none
F INT-LK: T2OUT LO, U3IN HI, T4OUT HI, Y3IN HI, X4OUT HI   none, hot
F INT-LK: T2OUT HI, U3IN LO, T4OUT LO, Y3IN HI, X4OUT HI   none, cold

F INT-LK: L5VES HI, P5VES HI   complete
F INT-LK: T2OUT LO, T5VES HI, X5VES HI   complete, hot
F INT-LK: T2OUT HI, T5VES LO, X5VES HI   complete, cold

F INT-LK: L6VES HI, P7VES HI   partial
F INT-LK: T2OUT LO, T6VES HI, T7VES HI, X6VES HI,
         X7VES HI   partial, single, hot
F INT-LK: T2OUT HI, T6VES LO, T7VES LO, X6VES HI,
         X7VES HI   partial, single, cold
F INT-LK: T2OUT LO, T6VES HI, T7VES HI, XC6VES HI,
         XC7VES HI   partial, multi, hot
F INT-LK: T2OUT HI, T6VES LO, T7VES LO, XC6VES HI,
         XC7VES HI   partial, multi, cold.

High pressure second side

F INT-LK: G1IN LO, G1IN NONE, G1IN REV, Q2OUT HI,
          R1IN LO, R1IN NONE, R1IN REV, P2OUT HI

F INT-LK: G3IN HI, Q4OUT LO, Q4OUT NONE, Q4OUT REV,
          R3IN HI, P4OUT LO, P4OUT NONE, P4OUT REV   none

F INT-LK: U1IN HI, T2OUT HI, T4OUT LO, Y1IN HI, X2OUT HI   none, hot

F INT-LK: U1IN LO, T2OUT LO, T4OUT HI, Y1IN HI, X2OUT HI   none, cold

F INT-LK: L5VES LO, P5VES LO   complete

F INT-LK: U1IN HI, T2OUT HI, T5VES LO, Y1IN HI, X2OUT HI   complete, hot

F INT-LK: U1IN LO, T2OUT LO, T5VES HI, Y1IN HI, X2OUT HI   complete, cold

F INT-LK: L6VES LO, P7VES LO   partial

F INT-LK: U1IN HI, T2OUT HI, T6VES LO, T7VES LO,
          Y1IN HI, X2OUT HI   partial, single, hot

F INT-LK: U1IN LO, T2OUT LO, T6VES HI, T7VES HI,
          Y1IN HI, X2OUT HI   partial, single, cold

F INT-LK: U1IN HI, T2OUT HI, T6VES LO, T7VES LO,
          YA1IN HI, YB1IN HI, XA2OUT HI, XB2OUT HI   partial, multi, hot

F INT-LK: U1IN LO, T2OUT LO, T6VES HI, T7VES HI,
          YA1IN HI, YB1IN HI, XA2OUT HI, XB2OUT HI   partial, multi, cold.

The partial blockage fault has been included to simulate a blockage occurring in some of the tubes or a partial blockage in the frame or plate side. It is assumed that a blockage of the shell side of any significance is impossible. This fault affects the flow rates and the temperatures depending on if the first side is the hot or cold side.

First side flow and pressure effects

F PART-BLK: G1IN LO, Q2OUT LO, R1IN LO, P2OUT LO

Second side flow and pressure effects

F PART-BLK: G3IN LO, Q4OUT LO, R3IN LO, P4OUT LO

F PART-BLK: G3IN LO, Q4OUT LO, Q5OUT,
          R3IN LO, P4OUT LO, P5OUT LO   partial.

Temperature effects

F PART-BLK: U1IN LO, T2OUT LO, U3IN LO, T4OUT LO   none, hot
F PART-BLK: U1IN HI, T2OUT HI, U3IN HI, T4OUT HI   none, cold
F PART-BLK: U1IN LO, T2OUT LO, T5VES LO   complete, hot
F PART-BLK: U1IN HI, T2OUT HI, T5VES HI   complete, cold
F PART-BLK: U1IN LO, T2OUT LO, T6VES LO, T7VES LO   partial, hot
F PART-BLK: U1IN HI, T2OUT HI, T6VES HI, T7VES HI   partial, cold.

The next set of event statements deal with faults specific to heat exchangers, these are: fouling, frothing and vapour blanket 'VAP-BLKT'. The faults are all associated with a reduction in the heat transfer efficiency. The effects are high or low temperature in the inlet and outlet streams depending on whether the first side is the hot or cold side. When a phase change occurs the faults also reduce the transfer rate between phases. The frothing fault represents liquid entrained in the vapour phase and the vapour blanket fault represents vapour trapped in the liquid phase. These two are used only if there is a phase change.

F FOULING: U1IN HI, T2OUT HI, U3IN LO, T4OUT LO   none, hot
F FOULING: U1IN LO, T2OUT LO, U3IN HI, T4OUT HI   none, cold
F FOULING: U1IN HI, T2OUT HI, T5VES LO,
            Q6VES LO, Q6VES NONE   complete, hot
F FOULING: U1IN LO, T2OUT LO, T5VES HI,
            Q6VES LO, Q6VES NONE   complete, cold
F FOULING: U1IN HI, T2OUT HI, T6VES LO,
            T7VES LO, Q8VES LO, Q8VES NONE   partial, hot
F FOULING: U1IN LO, T2OUT LO, T6VES HI,
            T7VES HI, Q8VES LO, Q8VES NONE   partial, cold

F FROTHING: U1IN HI, T2OUT HI, T5VES LO, Q6VES LO   complete, hot
F FROTHING: U1IN LO, T2OUT LO, T5VES HI, Q6VES LO   complete, cold
F FROTHING: U1IN HI, T2OUT HI, T6VES LO,
            T7VES LO, Q8VES LO   partial, hot
F FROTHING: U1IN LO, T2OUT LO, T6VES HI,
            T7VES HI, Q8VES LO   partial, cold

```
F VAP-BLKT: U1IN HI, T2OUT HI, T5VES LO, Q6VES LO          complete, hot
F VAP-BLKT: U1IN LO, T2OUT LO, T5VES HI, Q6VES LO          complete, cold
F VAP-BLKT: U1IN HI, T2OUT HI, T6VES LO,
            T7VES LO, Q8VES LO                             partial, hot
F VAP-BLKT: U1IN LO, T2OUT LO, T6VES HI,
            T7VES HI, Q8VES LO                             partial, cold.
```

If there is a phase change vessel ports are used to link inlet and outlet ports. However, the *some* deviation is not defined at vessel ports (see the end of section 7.3.3) and it is necessary to use the S NORMAL event statement. These introduce the *some* deviation for ports on the second side with a phase change. They are also used to define any relief or drain ports as having no flow under normal conditions.

```
S NORMAL: G3IN SOME, Q4OUT SOME, R3IN SOME, P4OUT SOME  complete
S NORMAL: G3IN SOME, Q4OUT SOME, Q5OUT SOME,
          R3IN SOME, P4OUT SOME, P5OUT SOME              partial
S NORMAL: Q3OUT NONE, Q6OUT NONE                         relief port.
```

The S IMPOSS event statements are used to define which deviations are impossible as given by the user and assumed in the methodology. This covers reverse flow at certain ports and reverse flow from liquid to vapour or vapour to liquid.

```
S IMPOSS: G1IN REV, Q2OUT REV, G3IN REV, Q4OUT REV      optional
S IMPOSS: Q6VES REV                                     complete
S IMPOSS: Q8VES REV                                     partial.
```

This completes all the necessary event statements for the heat exchanger model.

8.3.5 Generation Of Decision Tables

The decision tables are used to combine temperature and composition deviations at ports where a flow condition other than the normal state needs to occur before the deviation can propagate. This is generally reverse flow through the second side outlet ports where a phase change occurs.

The causal events for the vessel temperature deviations are reverse temperature at the outlet port ANDed with reverse flow through that port.

V U4OUT HI V Q4OUT REV T T5VES HI	complete
V U4OUT LO V Q4OUT REV T T5VES LO	complete
V U4OUT HI V Q4OUT REV T T6VES HI	partial
V U4OUT LO V Q4OUT REV T T6VES LO	partial
V U5OUT HI V Q5OUT REV T T7VES HI	partial
V U5OUT LO V Q5OUT REV T T7VES LO	partial.

The ports considered also include the relief or drain ports.

V U5OUT HI V Q5OUT REV T T6VES HI	complete, relief or drain
V U5OUT LO V Q5OUT REV T T6VES LO	complete, relief or drain
V U6OUT HI V Q6OUT REV T T8VES HI	partial, relief
V U6OUT LO V Q6OUT REV T T8VES LO	partial, relief
V U6OUT HI V Q6OUT REV T T7VES HI	partial, drain
V U6OUT LO V Q6OUT REV T T7VES LO	partial, drain.

The vessel composition is affected by the reverse composition at the outlet ports only if accompanied by reverse flow.

V Y4OUT HI V Q4OUT REV T X5VES HI	complete
V Y4OUT LO V Q4OUT REV T X5VES LO	complete
V Y4OUT HI V Q4OUT REV T X6VES HI	partial, single
V Y4OUT LO V Q4OUT REV T X6VES LO	partial, single
V Y5OUT HI V Q5OUT REV T X7VES HI	partial, single
V Y5OUT LO V Q5OUT REV T X7VES LO	partial, single
V YA4OUT HI V Q4OUT REV T XB6VES LO	partial, multi
V YA4OUT LO V Q4OUT REV T XB6VES HI	partial, multi
V YB4OUT HI V Q4OUT REV T XB6VES HI	partial, multi
V YB4OUT LO V Q4OUT REV T XB6VES LO	partial, multi
V YA5OUT HI V Q5OUT REV T XA7VES HI	partial, multi
V YA5OUT LO V Q5OUT REV T XA7VES LO	partial, multi
V YB5OUT HI V Q5OUT REV T XA7VES LO	partial, multi
V YB5OUT LO V Q5OUT REV T XA7VES HI	partial, multi.

A similar set of decision tables is obtained for deviations through the relief or drain ports.

V Y5OUT HI V Q5OUT REV T X6VES HI	complete
V Y5OUT LO V Q5OUT REV T X6VES LO	complete

```
V Y6OUT HI V Q6OUT REV T X8VES HI        partial, single, relief
V Y6OUT LO V Q6OUT REV T X8VES LO        partial, single, relief
V Y6OUT HI V Q6OUT REV T X7VES HI        partial, single, drain
V Y6OUT LO V Q6OUT REV T X7VES LO        partial, single, drain


V YD6OUT HI V Q6OUT REV T XD8VES HI      partial, multi, relief
V YD6OUT LO V Q6OUT REV T XD8VES LO      partial, multi, relief
V YD6OUT HI V Q6OUT REV T XD7VES HI      partial, multi, drain
V YD6OUT LO V Q6OUT REV T XD7VES LO      partial, multi, drain.
```

This completes all the necessary decision tables for the heat exchanger model.

8.3.6 Additional Information

There are two final sections of the model to be generated. The first is used by the main program when converting the text form of the model to the numerical form. The character variable used to indicate if the model has multi-component features is redefined to the text string "There are no multi-component features" or "There are multi-component features". This also serves the purpose of informing a user if the model uses this feature.

The second section is purely for user information and is not used by the programs. A text string is set up which is used to indicate the normal state of the model. This contains a list of ports which normally have flow.

The generation of the heat exchanger model is now complete.

8.4 List Of Example Models

Six examples are given to illustrate the rules given in this chapter. There is a partial reboiler, a partial condenser, a total reboiler, a total condenser, a heater unit and a cooler unit. These models are contained in Appendix D and the engineering description gives details on the port types.

8.5 Heat Exchanger Summary Tables

Table 8.2 - Options	
Heat exchanger type :	Shell & tube and Air-cooled finned Frame & plate and Double pipe
First side temperature :	Hot Cold
First side phase change :	None
Second side phase changes :	None Partial Complete
Multi-component features :	Optional for partial phase change
Pressure difference :	First side at higher pressure Second side at high pressure Insignificant pressure difference
Additional ports (each side) :	Relief and/or drainage port
First side port flow type :	As for pipe type unit Optional reverse flow
Second side port flow types :	Pressure effects No pressure effects Optional reverse flow
Additional port flow type :	Normally no flow Pressure effects Reverse flow

Table 8.3 - Faults	
External heat source	F EXT-HEAT
External cold source	F EXT-COLD
Leak to a low pressure environment	F LK-LP-EN
Internal leak	F INT-LK
Partial blockage	F PART-BLK
Fouling	F FOULING
Frothing	F FROTHING
Vapour blanketing	F VAP-BLKT

8.6 References

1. Perry R. H., Chilton C. H.; Chemical Engineers' Handbook, 5th Edition; McGraw-Hill; 1974; pp 11-3, pp 11-20–11-27, pp 11-41–11-42.

2. Coulson J. M., Richardson J. F.; Chemical Engineering, Volume 1, 3rd Edition; Pergamon; 1977; pp 170–171.

3. Coulson J. M., Richardson J. F.; Chemical Engineering, Volume 6; Pergamon; 1983; pp 511.

4. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1987; pp 3-29–3-32.

5. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1987; pp 5-19–5-23.

Chapter 9

Rules For The Generation Of Reaction Vessel Models

9.1 Introduction

This chapter deals with the generation of reaction vessel unit models. These are probably the most diverse items of equipment, around which all the other units are based. Ideally the generation of reaction vessels would be of great advantage to the user but the variety of different reactions makes it very difficult to sort out any type of taxonomy.

However, a reaction vessel can be broken into a combination of two types of process unit, a holding vessel and a heat exchanger (since the majority of reactions require a heat input system to start and maintain a reaction or a removal system to prevent a runaway reaction). Systems for categorising vessels and heat exchangers have already been discussed in the previous chapters. The next step is to look at characteristics particular to reaction vessels.

9.2 Reaction Vessel Characteristics

There are many different types of reaction vessel of all sizes which makes a characterisation process seem very difficult. However, Coulson (2) has looked at the design procedure used for reaction vessels and has put forward three classes to characterise them. To enable fault modelling to be carried out an additional class has been added giving the following four classes,

1. Mode of operation: batch, continuous or hybrid
2. Reaction vessel geometry: stirred tank, tubular or packed/ fluidised bed
3. Phases present: homogeneous or heterogeneous
4. Heat transfer requirements: none, cooling or heating, internal or external

These are explained more fully by Coulson (1) and in the following sections.

9.2.1 Mode Of Operation

A batch process operates in three distinct stages. There is a reaction vessel filling or charging stage when all the reactants are added to the vessel, a reaction stage when the chemical reaction takes place and a reaction vessel emptying or discharge phase when the products and any unconverted reactants are removed.

Reaction vessels situated in a batch operation process are used mainly for very slow reactions and specialist chemical production. They require less auxiliary equipment but are less economical to run for mass production. They are particularly useful for polymerisation and biochemical reactions because the time between batches can be used to clean the equipment. A disadvantage is that all the reactants are contained in the reaction vessel which prevents control of the reaction rate by control of the reactant feed. There is a strong possibility of a runaway reaction.

A batch process is shown schematically in Figure 9.1.

Figure 9.1 - Batch Reaction Process
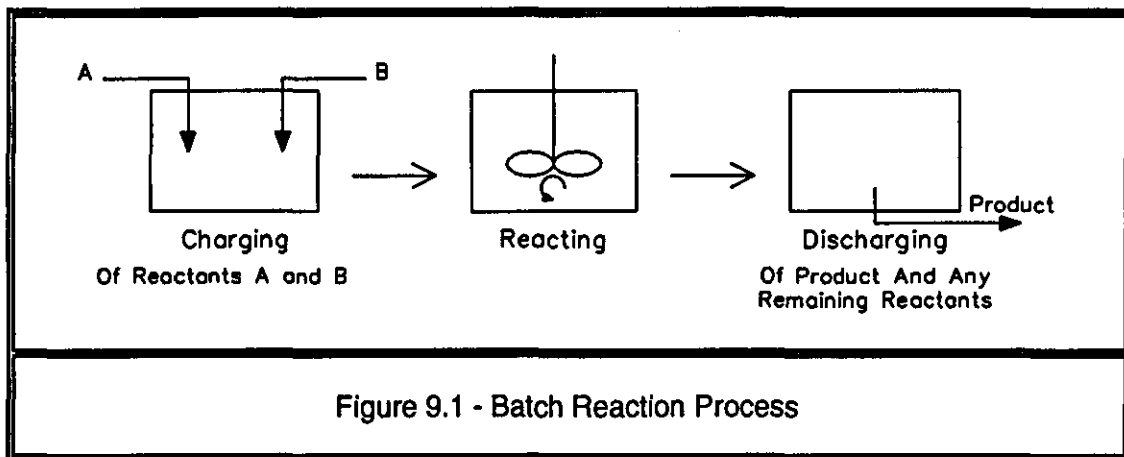
A continuous process is where all the reactants are continually added to the reaction vessel and the product and any unreacted material continually removed. The entire reaction process may occur in several reaction vessels placed in series.

This is very economical for large scale production of chemicals. The disadvantage lies in the capital cost of the control and other safety systems. A continuous process is shown schematically in Figure 9.2.

Figure 9.2 - Continuous Reaction Process
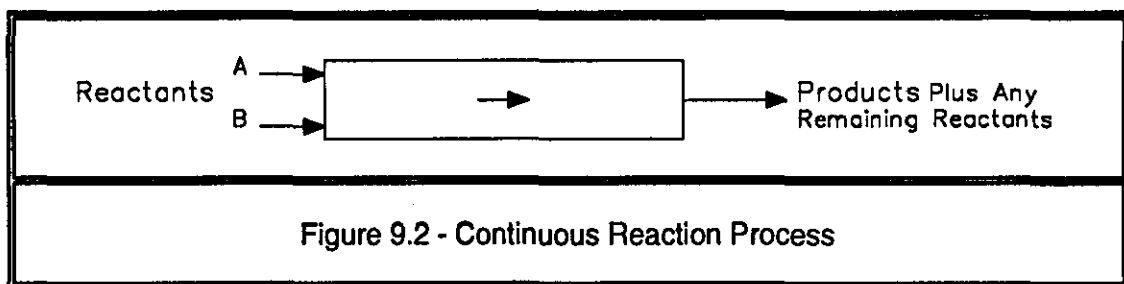
In some reaction processes it may be necessary to keep the concentration of one of the reactants low due to violent or side reactions or simply because of the volume of reactant required (eg. gases). For these a hybrid reaction vessel set-up is used such that one of the reactants is added at the rate at which it is used or at different intervals. This gives

better control than simple batch operation and less complex control than continuous operation. There are three types, a simple semi-batch reaction vessel, a divided-feed reaction vessel and a divided-feed to several reaction vessels in series as shown schematically in Figure 9.3.

Figure 9.3 - Hybrid Reaction Processes
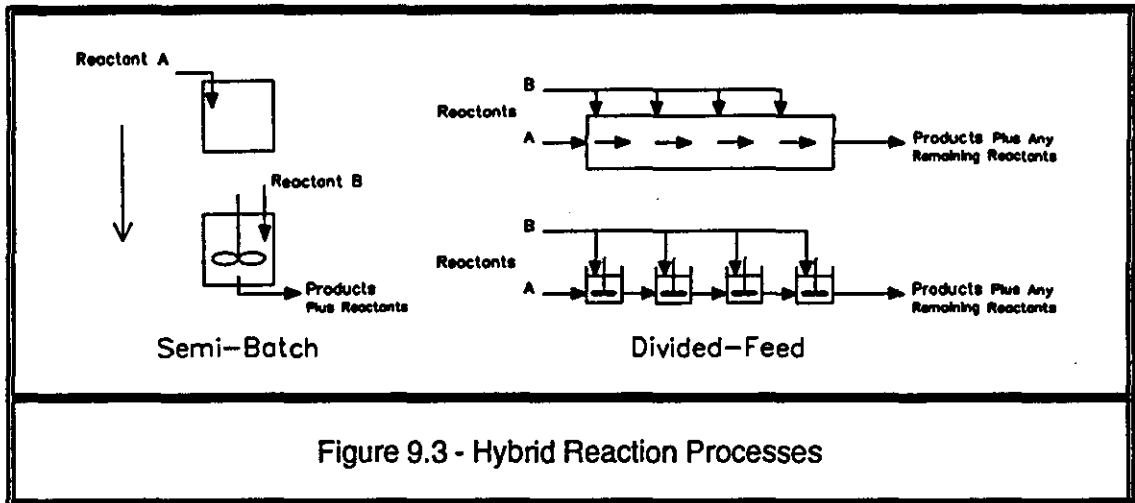
9.2.2 Reaction Vessel Geometry

The actual reaction vessel can be categorised into one of three types. There is the stirred tank or back-mix vessel which has some form of paddle to mix the reactants and products together such that the concentration of any component in the vessel is the same throughout. These may be operated in a batch, continuous, semi-batch or divided-feed process.

A tubular reaction vessel is simply a long tube into which reactants are added at one end and products withdrawn at the other. These are always operated as continuous reaction vessels or as divided-feed reaction vessels.

Packed bed and fluidised bed reaction vessels consist of a vessel with some form of packing which acts as a catalyst or even a reactant. In packed beds the packing remains stationary but with fluidised beds the flow rate of the reactants is sufficient enough to lift the packing and keep it mobile. The packing does not leave the vessel in the product stream. They may be operated as continuous or divided-feed processes.

Figure 9.4 shows the three types of reaction vessel geometry.
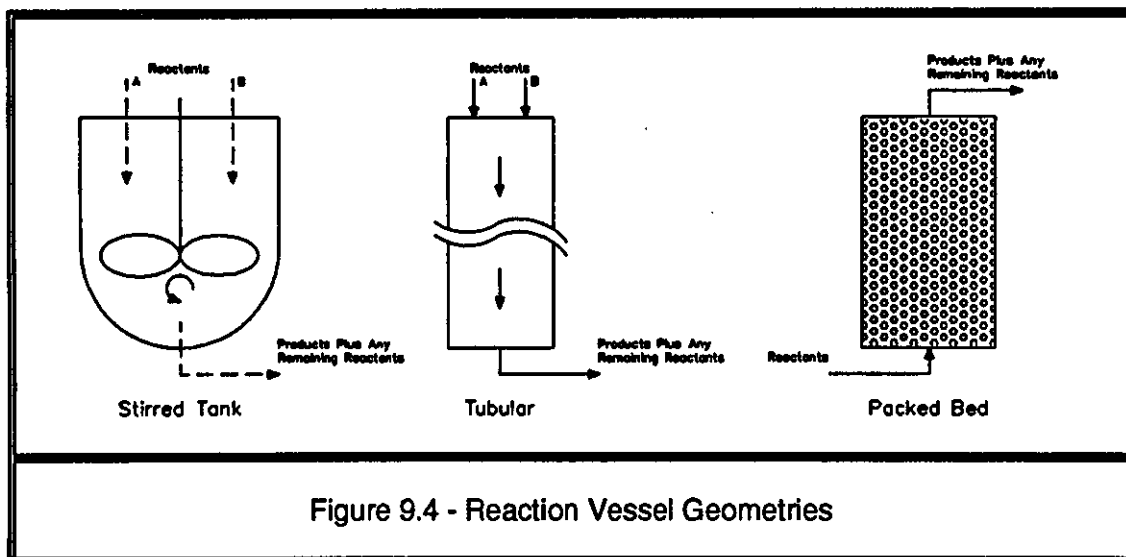
Figure 9.4 - Reaction Vessel Geometries

9.2.3 Phases Present

A homogeneous mixture is one where the reactants, products and any catalyst are present in the same phase. A homogeneous gas phase reaction is always carried out as a continuous process.

A heterogeneous mixture is one where there is more than one phase present.

9.2.4 Heat Transfer Requirements

Most chemical reactions involve the generation of heat (exothermic) or a heat input to keep them going (endothermic). Sometimes the heat removal through the product take-off is sufficient to prevent a runaway reaction or the reactants are heated sufficiently prior to entry to the reaction vessel that no more heat is required. These reactions do not require heat transfer processes to be modelled in the reaction vessel.

There may also be a take-off stream which syphons off reaction fluid, passes it through a heat exchanger and recycles it back to the reaction vessel. Cases like this are external sources and do not require the heat transfer system to be modelled in the reaction vessel.

There are many chemical reactions which do require some additional form of heat transfer system and it is often necessary to provide a heat source or sink to the reaction vessel. These take the form of jackets around the vessel, coils within the vessel or the 'vessel' forms coils within a heat transfer medium. All of these types require the heat transfer process to be modelled within the reaction vessel model.

As with heat exchangers (section 8.2) it is necessary to distinguish between exothermic and endothermic reactions so the appropriate temperature equations can be used.

9.2.5 Common Reaction Vessel Types

Batch operations are usually carried out for liquid phase or liquid-solid reactions in stirred tank type reaction vessels.

Semi-batch operations are also carried out in stirred tank type reaction vessels. They are usually used for contacting liquids with gases or for violent liquid phase reactions.

Continuous operations for gas phase reactions are carried out in tubular reaction vessels. However, by far the most common reaction vessel is the continuous stirred tank reaction vessel (CSTR) used for liquid phase reactions.

9.3 Reaction Types

The number of types of chemical reaction are even more numerous than the vessel types and can involve gases, liquids and solids. Some example equations are

$$A \rightarrow C$$
$$A + B \rightarrow C$$
$$A \rightarrow C + D$$
$$A + B \rightarrow C + D$$
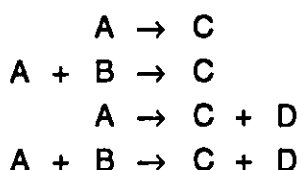
A major factor in any chemical reaction is the temperature. This more than any other parameter affects the reaction rate although secondary factors such as changes in volume, pressure and concentration can be involved. In all reactions an increase in temperature results in a increase in reaction rate.

In gas phase reactions there can be significant changes in volume and pressure. For example, the reaction for the second equation results in a decrease in volume (at constant pressure) and the reaction for the third equation results in an increase in pressure (at constant volume). To encourage the reaction to go to completion quickly a high pressure is used for the second equation.

9.4 Modelling Assumptions And Simplifications

Chemical reactions involving gases often result in a change in volume. Most of these though are carried out as continuous or hybrid processes. This greatly simplifies the modelling because under normal operating conditions any change in volume is a normal state. Consider a reaction where the volume doubles (third equation). If the flow rate into the reaction vessel is lower than normal then the product flow rate will also be lower than normal because there is less reactant to react. The models assume that a volume or pressure change occurs under normal steady state operation and does not represent a fault condition. Hence it is possible to model the reaction vessel without considering these changes.

The stoichiometry of reactions involving gases has other effects besides volume. For example, it is assumed that a higher temperature may cause either a lower or higher pressure, depending on whether the number of gaseous atoms on the right hand side of the equation decreases or increases. This occurs because more reactant than normal has reacted. Also it is assumed that a higher pressure may cause either a higher or lower reaction rate due to a decrease or increase in the amount of gas phase material present.

Reactions that involve catalysts can be modelled in two ways. If the catalyst is a solid it is assumed to be contained within the reaction vessel in the form of a packed/fluidised bed and will not pass out into the product stream under normal operation. The second type is where the catalyst is a liquid (or gas) and enters the reaction vessel in a reactant stream. The catalyst can be modelled using the concentration variable which has the deviations of high (XC1IN HI) or low (XC1IN LO). For the solid catalyst the low deviation may be caused by exhaustion or poisoning and the high deviation by too much catalyst in the reaction vessel. For the liquid the low deviation may be caused by poisoning or generally low concentration in the stream and the high deviation may be caused by an excess in the stream.

The reaction stage in a batch reaction process can be modelled simply as a vessel which has no inlet ports or outlet ports on the reaction side. Control is maintained by adjusting the flow of coolant or heating fluid. The hybrid reaction process must be modelled in a similar way to a continuous process.

The internal characteristics of the reaction vessel are modelled using the vessel port. This port is used to model level (if applicable), temperature, pressure and concentration. Since flow cannot be modelled at the same port as level a separate vessel port is required to model the transfer of material from the reactant side to the product side, ie. the reaction rate.

Reactions which require a heat transfer system built into the reaction vessel can be modelled in the same way whether it is in the form of an external device such as a jacket or furnace, or an internal coil. Basically there is a separate flow stream for heat input or removal which can have its own deviations and affect the temperature within the reaction vessel. It is necessary to determine the high pressure side, if any, such that internal leak faults can be modelled correctly.

9.5 Example Models

Consider the catalytic reaction

$$A + B \rightarrow C + D$$

There are no side reactions and the catalyst, represented by the component letter 'K', is mixed in with reactant A. The reaction is to be carried out continuously but there are two different scenarios.

9.5.1 Stirred Tank Reaction Vessel

As an exothermic liquid phase reaction it may be carried out in a stirred tank reaction vessel. The reactants are pumped into the reaction vessel separately but the products and any unreacted material is removed through a single outlet port. An internal cooling coil is used with liquid coolant at a higher pressure than the reaction vessel and contents. The catalyst may be poisoned by the coolant and all reverse flow effects are ignored. A schematic representation is shown in Figure 9.5.
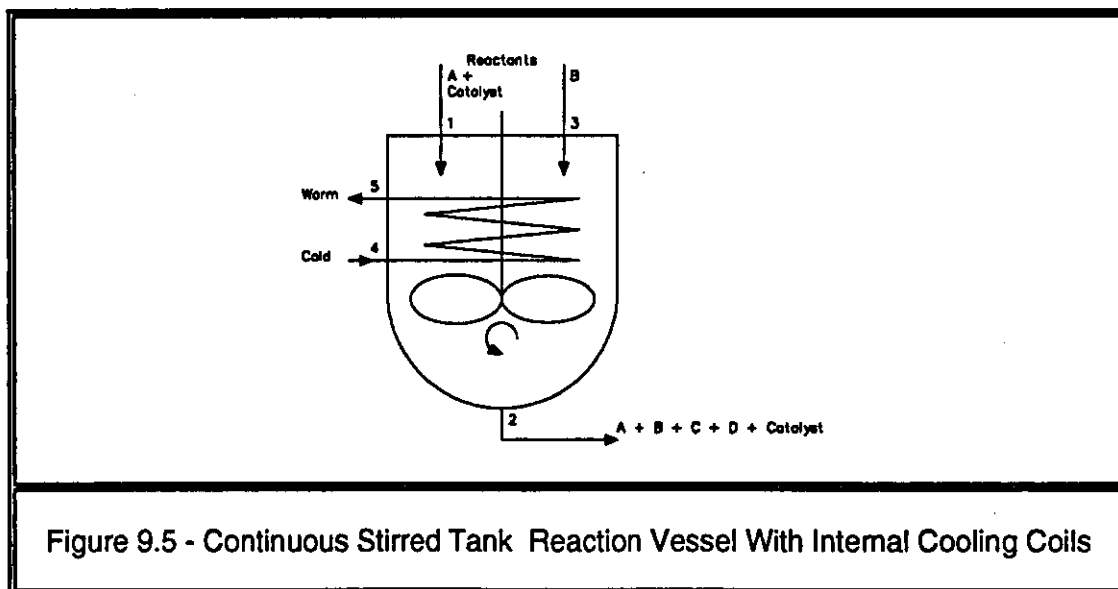
Figure 9.5 - Continuous Stirred Tank Reaction Vessel With Internal Cooling Coils

The numbers on the reaction vessel represent the port numbers. These are

    port 1 - inlet port for reactant A and catalyst

    port 2 - outlet port for catalyst, products and any unreacted material

    port 3 - inlet port for reactant B

    port 4 - inlet port for cooling liquid

    port 5 - outlet port for cooling liquid

    port 6 - vessel port used to model internal variables

    port 7 - vessel port used to model the reaction rate

The flow and pressure propagation equations for the reaction side are

$$G1IN = F(Q1IN)$$
$$R1IN = F(-P6VES)$$
$$G3IN = F(Q3IN)$$
$$R3IN = F(-P6VES)$$
$$Q2OUT = F(G2OUT, P6VES)$$
$$P2OUT = F(P6VES)$$

These show there are no pressure effects on the inlets and there are no direct links between the inlet and outlet ports.

The flow and pressure propagation equations for the coolant side are the same as in the pipe model.

$$G4IN = F(Q4IN, Q5OUT)$$
$$R4IN = F(R5OUT)$$
$$Q5OUT = F(G4IN, G5OUT)$$
$$P5OUT = F(P4IN)$$

The outlet temperature of the reaction side is a function of the internal temperature but the outlet of the coolant side is also affected by the inlet temperature and flow.

$$T2OUT = F(T6VES)$$
$$T5OUT = F(-G4IN, T4IN, T6VES)$$

The internal level and pressure variables are modelled using the following propagation equations

$$L6VES = F(G1IN, G3IN, -Q2OUT)$$
$$P6VES = F(L6VES, T6VES)$$

The internal temperature and composition of the components are modelled using the following propagation equations

```
T6VES= F( T1IN, T3IN, -G4IN, T4IN, Q7VES)
XA6VES= F( XA1IN, -Q7VES)
XB6VES= F( XB3IN, -Q7VES)
XC6VES= F( Q7VES)
XK6VES= F( XK1IN)
```

All are affected by the reaction rate except the catalyst composition. Component 'D' is not modelled because it is the same as component 'C'. The outlet compositions are simple functions of the internal compositions.

The reaction rate is modelled on a separate port. It is a function of the internal temperature, the total component flows of the two reactants and the catalyst composition. Event statements using total component flow are required in addition to the propagation equation.

```
Q7VES= F( T6VES, XK6VES)


V  G1IN ⊗ XA1IN LO : Q7VES LO
V  G3IN ⊗ XB3IN LO : Q7VES LO
V  G1IN ⊗ XA1IN HI : Q7VES HI
V  G3IN ⊗ XB3IN HI : Q7VES HI
```

Some of the event statements that can be used with the model are:

```
F  INT-LK : XK6VES LO
F  MIX-BROK : Q7VES LO
```

to represent the catalyst being poisoned by the coolant leaking into the reaction vessel and the reaction rate being reduced by a broken mixer. Other event statements may be used for high internal temperature caused by poor heat transfer or agitation.

Note :   This is only an outline of some of the propagation equations and event statements required for a continuous stirred tank reaction vessel model and does not represent the entire model.

9.5.2 Tubular Reaction Vessel

As an endothermic gas phase reaction it may be carried out in a tubular reaction vessel. The pumped reactants and catalyst are mixed before entry to the vessel through a single inlet and the products and any unreacted material removed through the single outlet. The reaction tube is jacketed and uses a gaseous heating medium at a higher pressure than the reaction material. The catalyst may be poisoned by the heating medium and all reverse flow effects are ignored.

A schematic representation is shown in Figure 9.6.
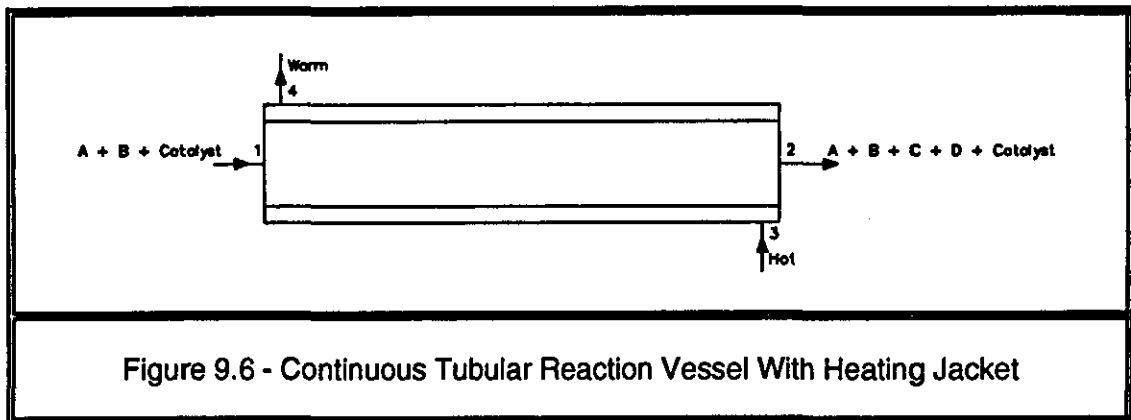
Figure 9.6 - Continuous Tubular Reaction Vessel With Heating Jacket

The numbers on the reaction vessel represent the port numbers. These are

    port 1 - inlet port for reactants and catalyst
    port 2 - outlet port for catalyst, products and any unreacted material
    port 3 - inlet port for heating medium
    port 4 - outlet port for heating medium
    port 5 - vessel port used to model the reaction rate

The flow and pressure propagation equations for the reaction side and jacket side are the same as the pipe model. There are no pressure effects

    G1IN= F(Q1IN, Q2OUT)
    R1IN= F(R2OUT)
    Q2OUT= F(G1IN, G2OUT)
    P2OUT= F(P1IN)
    G3IN= F(Q3IN, Q4OUT)
    R3IN= F(R4OUT)
    Q4OUT= F(G3IN, G4OUT)
    P4OUT= F(P3IN)

The outlet temperatures are functions of the inlet flows and temperatures and the reaction rate. (NB. High inlet reactant temperature does not result in high outlet temperature because another effect is a higher reaction rate which causes cooling.)

T2OUT= F(G3IN, T3IN, -Q5VES)
T4OUT= F(-G1IN, T1IN, G3IN, T3IN, -Q5VES)

V T1IN LO : T2OUT LO

The composition of the components are modelled using the following propagation equations:

XA2OUT= F(XA1IN, -Q5VES)
XB2OUT= F(XB1IN, -Q5VES)
XC2OUT= F(Q5VES)
XK2OUT= F(XK1IN)

All are affected by the reaction rate except the catalyst composition. Component 'D' is not modelled because it is the same as component 'C'.

The reaction rate is a function of the inlet flow of heating medium, the temperatures, the total component flows of the two reactants and the catalyst composition. In this case the total component flows of both components must be high for the reaction rate to be high. Event statements and decision tables are required in addition to the propagation equation.

Q5VES= F(T1IN, G3IN, T3IN, XK1IN)

V G1IN ⊗ XA1IN LO : Q5VES LO
V G1IN ⊗ XB1IN LO : Q5VES LO
V G1IN ⊗ XA1IN HI : A(DUMMY)
V G1IN ⊗ XB1IN HI : B(DUMMY)

I A(DUMMY) I B(DUMMY) T Q5VES HI

Since the reaction rate is the only internal variable the effect of catalyst poisoning due to an internal leak is modelled using this rather than the catalyst composition.

F INT-LK : Q5VES LO

Note :   This is only an outline of some of the propagation equations, event statements
         and decision tables required for a tubular reaction vessel model and does not
         represent the entire model.

9.6 Reaction Vessel Summary Table

OPTIONS	
Reaction vessel mode :	Batch Continuous Hybrid.
Reaction vessel geometry :	Stirred tank Tubular Packed/Fluidised bed.
Reaction mixture phase :	Homogeneous liquid Homogeneous gas Heterogeneous.
Gas phase stoichiometry :	No change Increase Decrease.
Catalyst :	None involved Inside reaction vessel With reactant.
Heat transfer requirements :	None Heating by high pressure source Heating by low pressure source Cooling by high pressure source Cooling by low pressure source.
Additional ports :	Relief (pressure effects) Drain (pressure and level effects).
Reaction port flow types :	Pressure effects No pressure effects.
Heat transfer ports flow type :	Pressure effects No pressure effects.
Reverse flow through ports :	Optional.

9.7 References

1. Coulson J. M., Richardson J. F.; Chemical Engineering, Volume 1, 3rd Edition; Pergamon; 1977; pp 3–84.

2. Coulson J. M., Richardson J. F.; Chemical Engineering, Volume 6; Pergamon; 1983; pp 382–385.

Chapter 10

Rules For Modelling Divider-Header Combinations

10.1 Introduction

The concept of divider-header combinations was introduced to solve a problem encountered during fault tree synthesis. This was caused by the methodology employed to achieve two way fault propagation for flow. Divider-header combinations were first used by Kelly (1) and later updated by Mullhi (2).

Consider the section of plant shown in Figure 10.1 which simply has a dividing and then joining of a process stream.
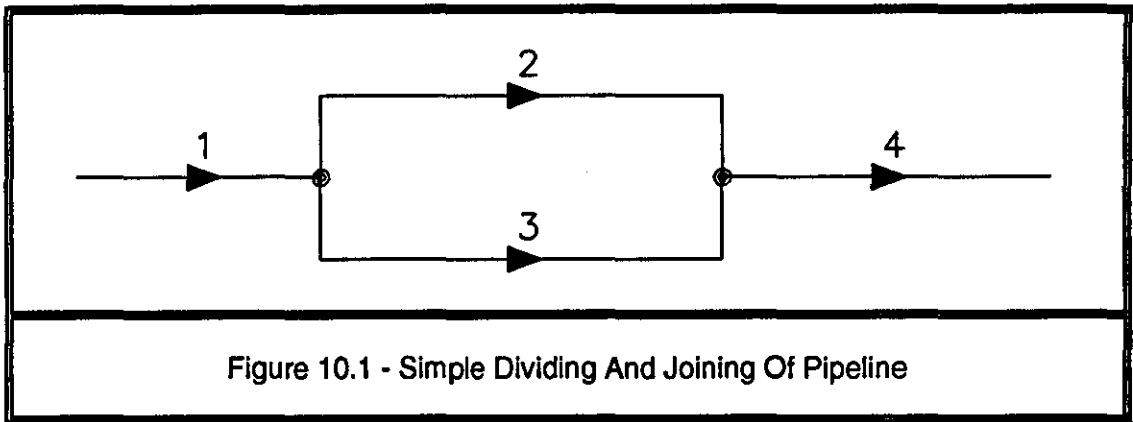
Figure 10.1 - Simple Dividing And Joining Of Pipeline

The partial fault tree for high flow in stream 4 is shown in Figure 10.2.
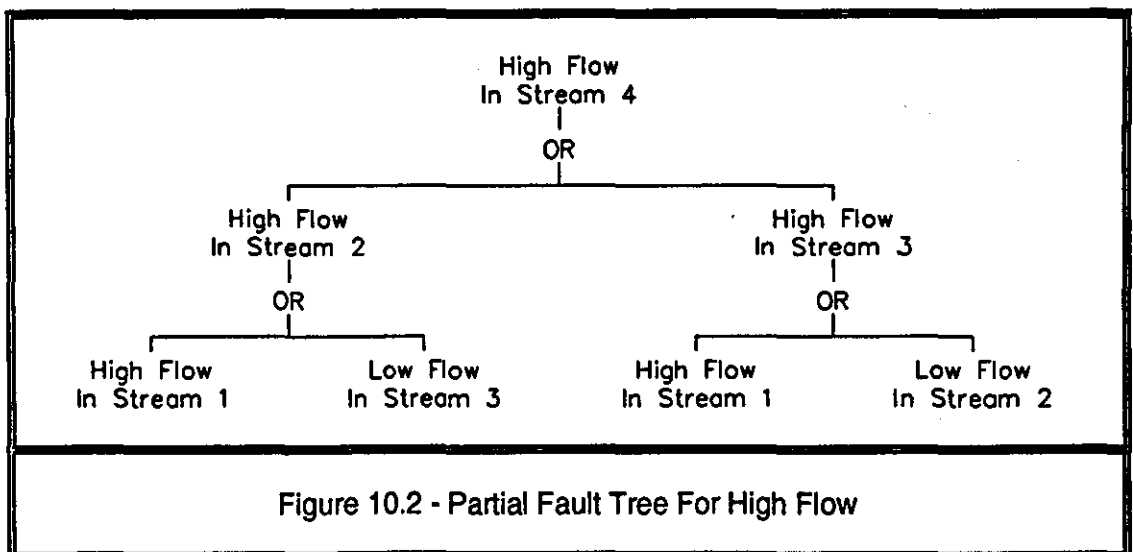
Figure 10.2 - Partial Fault Tree For High Flow

This tree is incorrect because low flow in either of the legs can never give rise to high flow in the outlet stream. The problem arises because synthesis is carried out vertically and consistency checks can only be applied with respect to the branch already developed. The two way propagation results in a loop of information flow which must either be handled by additional consistency checks after synthesis or by defining additional boundary conditions. The former was dropped because the incorrect branches still had to be developed before the checks could be carried out and this was felt unnecessary and time consuming.

10.2 Development Of A Solution

The approach adopted by Kelly was to treat the divider-header combinations as special sub-systems similar to control and trip loops. The user was required to identify the loops during the decomposition stage.

However, this was not as simple as it seems because there are three types of divider-header combination to be considered and each type had its own divider and header models.

a) By-pass with flow, eg. by-pass around a heat exchanger for the excess flow.

a) By-pass without flow, eg. by-pass around a control valve for on-line maintenance.

a) Parallel flow, eg. pump bank with equal flows.

The three types are illustrated in Figure 10.3.

Figure 10.3 - Types Of Divider-Header Combination
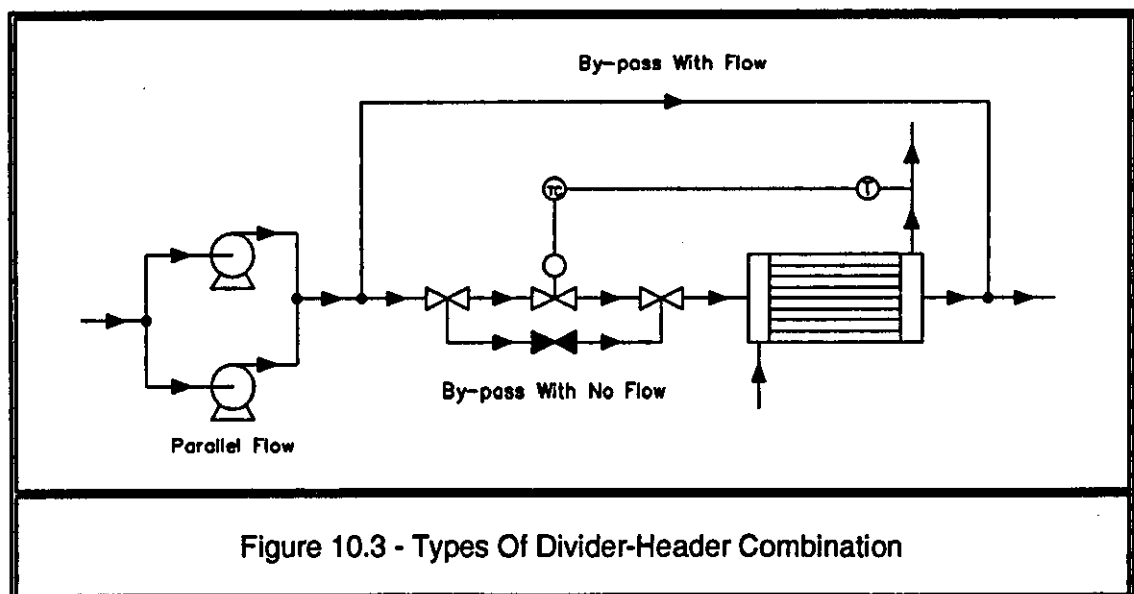
This technique produced the correct fault trees but was restricted to these specific types of divider-header combination and could not deal with overlapping loops. It also had the problem that the user had to identify all the divider-header combinations which in some cases could prove to be very difficult.

10.2.1 Automated Loop Searches

The approach used by Mullhi was to differentiate between the different types of divider-header combination during the synthesis stage. Furthermore, the restrictions on the functionality of the loops were relaxed to cover more types of loop.

The process depends on the ability to detect any loops in the information flow structure. The user has been relieved of the burden of this by an automatic identification procedure used during the configuration input.

All the divider and header units are identified from the connection topology array (section 4.4.3) which contains information on which unit and port number each connection starts from and goes to. Any unit which has two or more entries in the upstream unit column but only one entry in the downstream unit column is marked as a divider unit and any unit which has only one entry in the upstream unit column but two or more entries in the downstream unit column is marked as a header unit.

The next step is to reduce the search area by condensing the topology array until it just contains the divider and header units as nodes and the connections between them as the edges. This effectively removes the *flow through* units which contribute nothing to the loop.

It is then a matter of starting at each divider unit and trying to return back to it. The procedure is exhaustive as every possible route is tried but it leads to duplicated loops and loops which have sub-loops. These are removed by sorting the loops such that they always start and end with an outlet of the same divider and into an ascending order of the number of nodes.

The ends of the loops are identified such that closed flow paths can be recognised in both the upstream and downstream fault propagation routes. This enables any pairs of streams with continuity of flow to be recorded.

All this information is passed on to the synthesis algorithm so that the appropriate boundary conditions can be set up.

10.2.2 The Divider And Header Models

The Kelly approach to divider-header combinations used three different models for dividers and headers for each type of combination. This requires the user to not only identify the type of combination but also the correct divider and header model to use from the unit model library.

The automated search method introduced by Mullhi removes the need to find the loops and utilises single models for the divider and for the header. These models simply model flow as being the normal state in both legs.

10.2.3 Flow Capacities

Having reduced the number of models to choose from, the synthesis algorithm still requires the different types of divider-header combinations to be separated. This is carried out by specifying the flow capacities of each outlet leg of the divider or each inlet leg of the header during the configuration data input. This serves to identify non-flow legs and excess capacity in parallel systems. The user is prompted automatically for each divider or header and has to supply a value of 0, 50 or 100 to represent 0%, 50% or 100% of the required throughput, respectively.

10.3 Improvements To Models

Consider the configuration in Figure 10.4 which shows part of a larger plant system. The divider-header combination forms a by-pass with no flow through the closed hand valve.
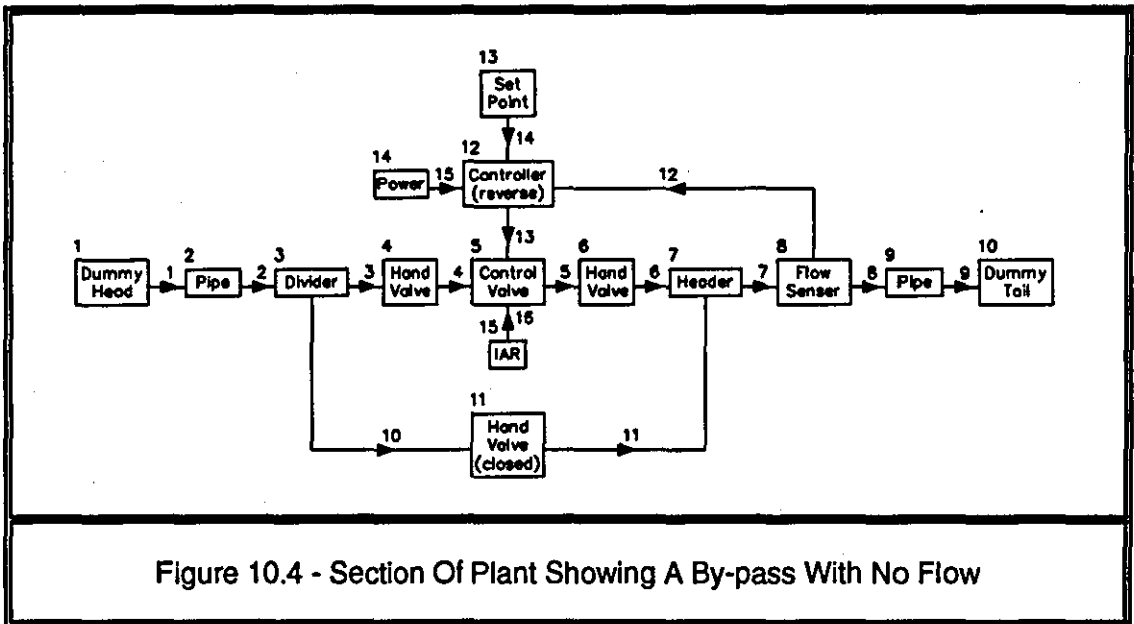
Figure 10.4 - Section Of Plant Showing A By-pass With No Flow

Figure 10.5 shows part of the fault tree for high flow through this plant section.
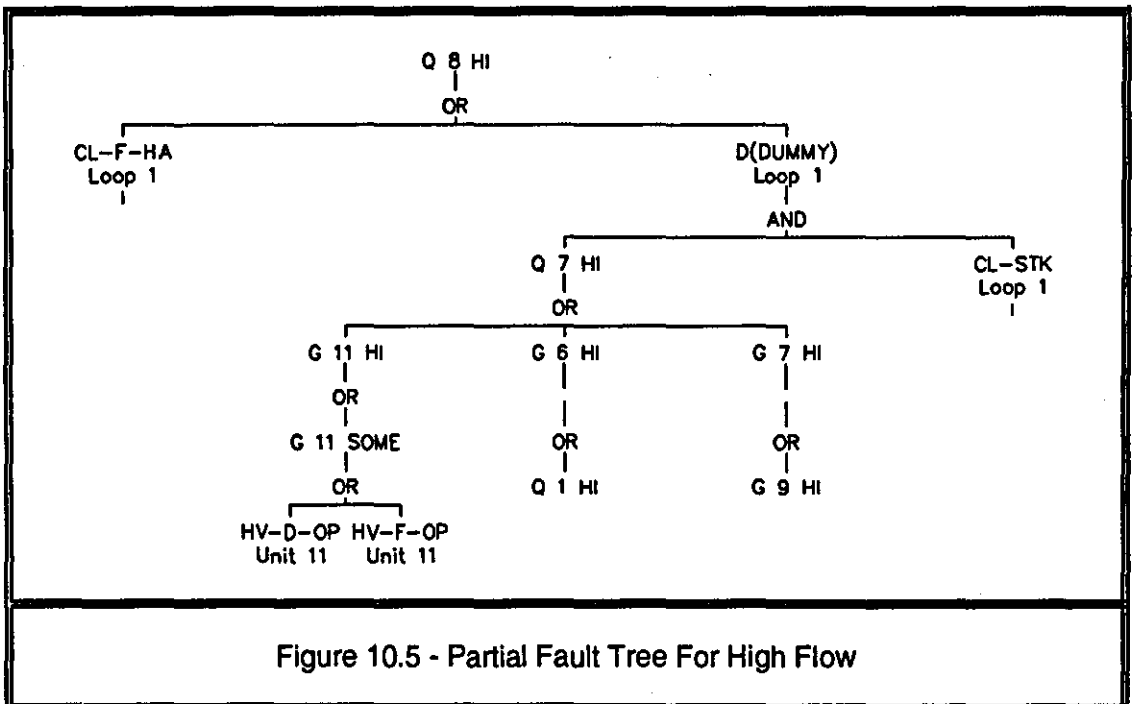
Figure 10.5 - Partial Fault Tree For High Flow

The control loop failure branches (CL-F-HA and CL-STK) take the normal format and have been omitted for clarity.

The high flow fault is traced to the outlet of the header. The causes of this are high flow down either inlet leg of the header or high pressure gradient out of the header. The high flow deviation in the non-flow leg of the header is correctly changed to some flow since 'SOME' represents any type of deviation in the normal direction of flow in a non-flow leg. The causes of some flow are given as only the hand valve being open.

There are two problems here, the first is that for there to be some flow out of the hand valve there must be some flow into it, which is not given, and the second is that this some flow deviation is not continued beyond the hand valve or divider unit.

A similar problem was encountered when tracing high flow into the divider unit rather than high flow out of the header unit. The causes of these problems were found to be in the divider and header unit models and the closed valve unit model.

10.3.1 Divider And Header Model Changes

The original methodology for handling divider-header combinations used three different pairs of divider and header models. These were adapted to model only one specific type of combination. This was inefficient and consequently the methodology was changed to make use of just one pair of divider and header models.

However, these models were not designed specifically for this approach but were simply a combination of the two flow models used previously. It was anticipated that the new methodology would be able to deal with the non-flow legs. This has been shown not to be the case.

Although, for example, a high flow deviation proceeding along a non-flow leg is changed to 'some flow' once the deviation reaches the other end of the divider-header combination the synthesis algorithm cannot find any causes of some flow. Consequently it is deleted as an impossible event.

It can be seen that the divider and header models require additional information in them to deal with the 'SOME' deviation. This will not alter the results for the flow legs because some flow is a normal state and will be deleted by the synthesis algorithm in these cases.

Changes to the divider model include: some flow in either outlet leg being caused by high inlet flow or reverse flow in the other outlet leg. Equivalent combinations are also used for pressure and relief.

The revised divider model is given in Appendix E.1.

Changes to the header model include: some flow in either inlet leg being caused by high outlet flow or reverse flow in the other inlet leg and high flow in one inlet leg may cause none or reverse flow in the other. Equivalent combinations are also used for pressure and relief.

The revised header model is given in Appendix E.2.

10.3.2 Closed Valve Model Changes

The original methodology used in this type of unit, which contains any unit which normally has no flow, was directed specifically at the non-flow leg divider-header combination. The methodology was different in that the 'HIGH' deviation meant 'SOME' and the 'LOW' deviation meant 'REVERSE'. This is very confusing and does not work with the new single divider and header model methodology. Therefore, the closed valve type unit models have had to be revised.

The models have been altered such that only the 'SOME' and 'REVERSE' deviations for flow and pressure are used. All deviations are ANDed with the valve being open in a decision table. Some examples are:

```
F  HV-F-OP  V  G1IN SOME  T  G2OUT SOME
F  HV-F-OP  V  Q2OUT SOME  T  Q1IN SOME
```

The model in this format only partially solves the problem noted in section 10.3. As can be seen in Figure 10.5 the cause of 'some flow' is the valve being open. However, there also has to be 'some flow' into the valve. Using the decision tables for G1IN and Q2OUT, above, allows the AND gate to be incorporated in the fault tree in one direction but not the other. For example, when tracing high flow out of the header unit the AND gate with the valve faults is used but when tracing high flow into the divider unit it is not. The reason for this is the way two-way fault propagation of flow is achieved. All possible combinations were tested but each resulted in only one direction with the AND gate.

Use of two pairs of decision tables for flow incorporating all the flow variables resulted in another problem.

```
F  HV-F-OP  V  G1IN SOME  T  G2OUT SOME
F  HV-F-OP  V  G2OUT SOME  T  G1IN SOME
F  HV-F-OP  V  Q1IN SOME  T  Q2OUT SOME
F  HV-F-OP  V  Q2OUT SOME  T  Q1IN SOME
```

The AND gate using these decision tables appeared twice in the fault tree. This is undesirable because it increases the cutset order by one which may result in loss of detail.

The modelling problem was solved by using event statements to link the flow variables for two-way fault propagation. The format is

V Q1IN SOME: G1IN SOME
V G2OUT SOME: Q2OUT SOME

F HV-F-OP V G1IN SOME T G2OUT SOME
F HV-F-OP V Q2OUT SOME T Q1IN SOME

This places a single AND gate in the fault tree for both directions. However, it also places an OR gate with the valve open faults below it for one of the directions. This can be seen in Figure 10.6 which gives the partial fault tree for high flow when the new models are used.
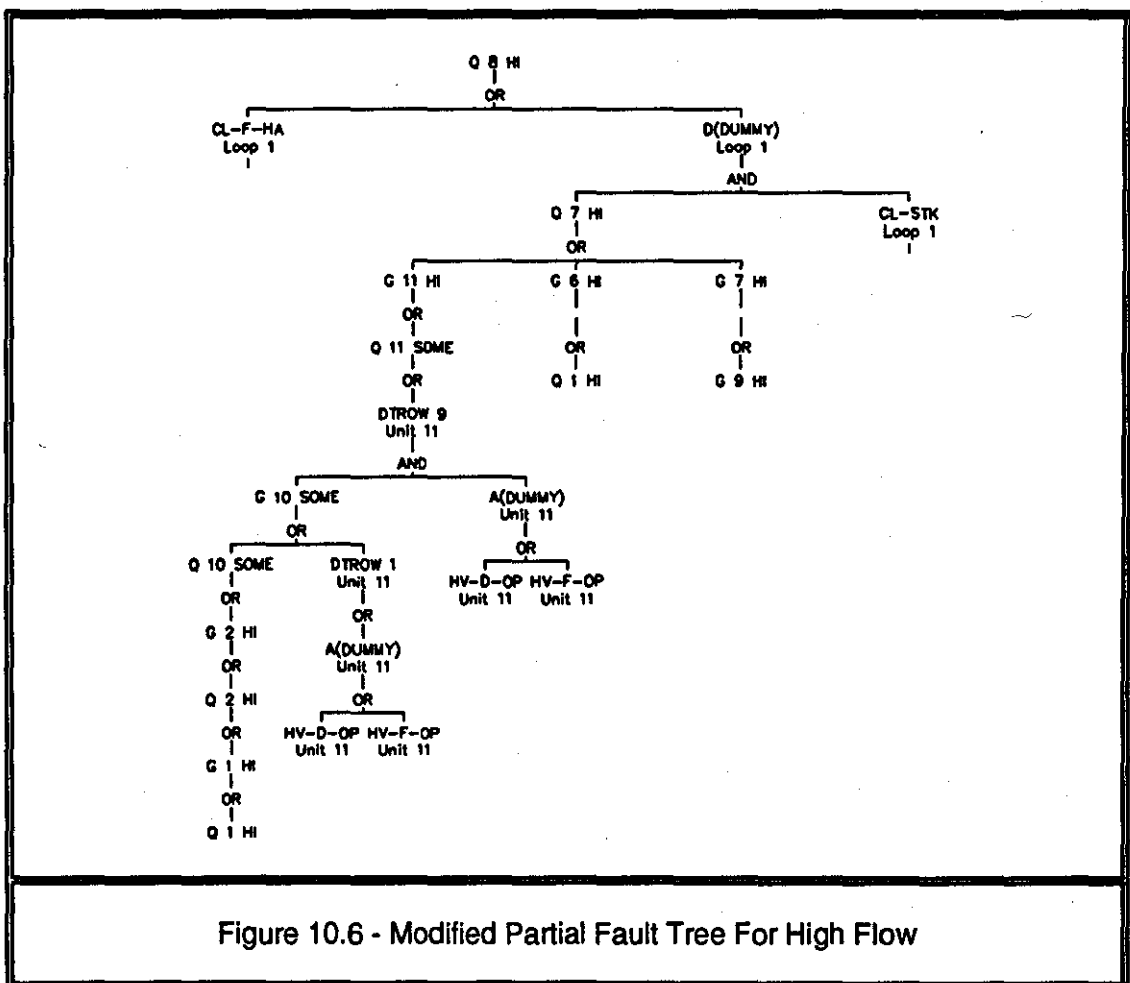
Figure 10.6 - Modified Partial Fault Tree For High Flow

The problem is not one of modelling in the valve but one of the synthesis algorithms consistency checking routines. The results of the synthesis process have been examined closely and during synthesis the AND gate appears twice as before. However, in the direction traced in Figure 10.6 the SOME deviation (not shown) under the 'DTROW 1 Unit 11' is immediately deleted as occurring further up the branch (series consistency : boundary conditions) changing the AND gate to a single cause OR gate.

In the other direction the boundary conditions series check does not work but during the parallel consistency checks the SOME deviation under the second decision table is noted as occurring further up the branch and is marked for deletion. The mark consists of the label 'impossible' which is processed later. As expected if one branch of an AND gate is impossible then the whole gate is impossible and is deleted. This conveniently removes the second decision table resulting is just the single occurrence of the valve open faults.

This investigation has partially solved the problem with closed valve models but has highlighted a discrepancy in the consistency checks which needs to be examined further.

The duplication of the valve open faults in the fault tree results in an 'incorrect' tree. However, this does not affect the final cutsets because the analysis program used recognises these duplicated events. Therefore, the additional branch in the tree can be considered as more of a nuisance rather than a problem because it does not have any influence on the structure of the rest of the fault tree or its analysis.

The new closed valve model is given in Appendix E.3. It should be noted that there are no propagation equations because under fault free conditions variable deviations cannot propagate through the unit.

10.4 Revision To Loop Searches

The purpose of divider-header combinations was to maintain a consistency between events leaving a divider-header combination and causes propagating into them. For example low flow propagating out of a header cannot be caused by high flow propagating into the divider (a valid cause of some flow in a non-flow leg).

The automatic loop search facility identifies every possible divider-header combination and information flow path loop. However, this also labels signal paths as divider-header combinations which, since they do not propagate the flow variable, is incorrect. The divider and header identification process has been modified such that the ports on the unit must be either of the inlet port type or the outlet port type.

The loops identified by this process incorporate every unit between the divider and the header regardless of the type. However, consider the three loops shown in Figure 10.7.

Figure 10.7 - Divider-Header Combinations Involving Vessels
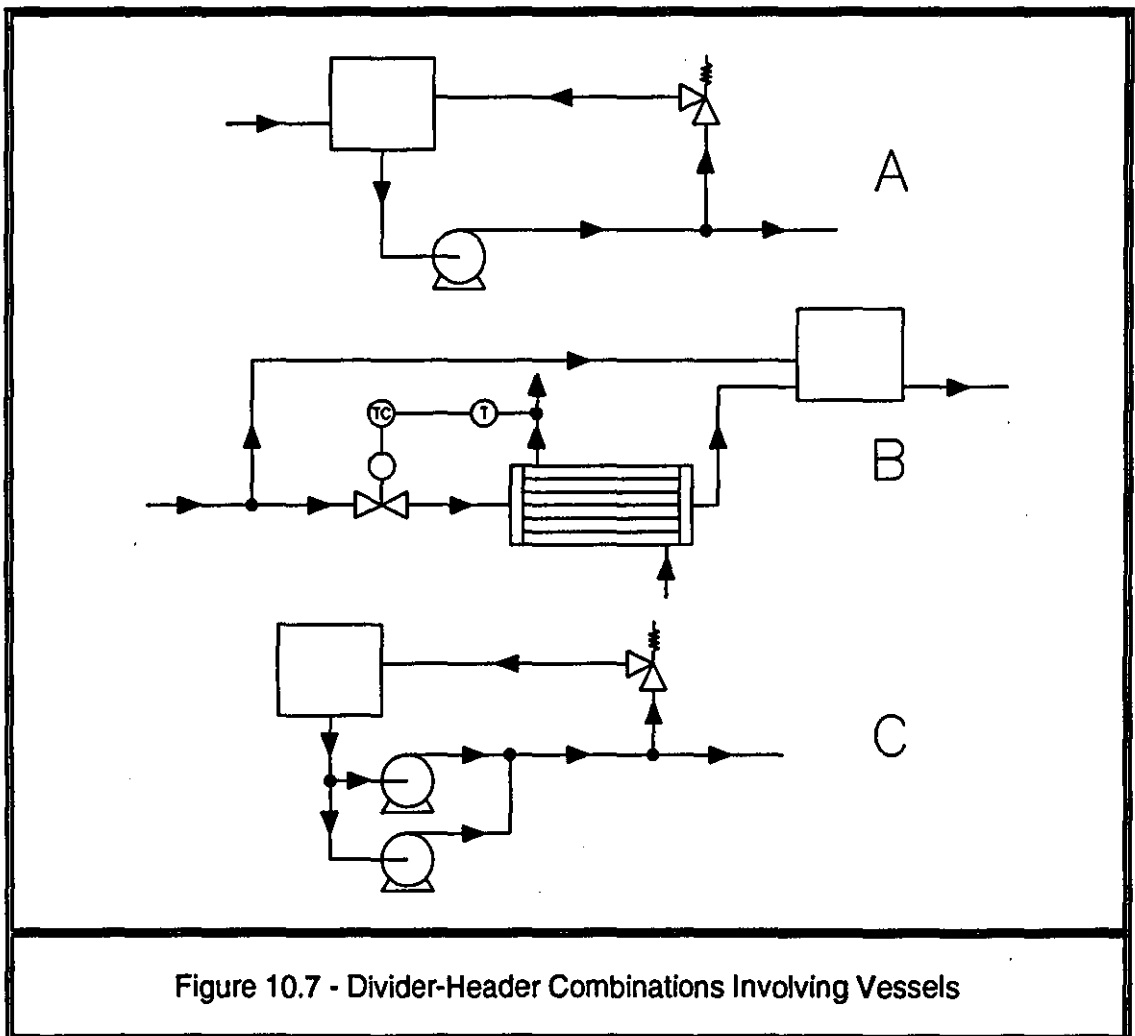
Configuration A shows a pump protection system consisting of a relief valve on a kick-back leg to the vessel which operates should the pressure rise above a certain level.

Configuration B shows a by-pass around a heat exchanger.

Configuration C shows a pump protection system for two pumps in parallel.

For configuration A the loop search identifies not only the divider on the kick-back leg but it also identifies the vessel as a header type unit. This is reasonable because the vessel does have two inlet ports and one outlet port. The process thus concludes that the vessel and divider unit form a divider-header combination or information loop.

As was mentioned earlier, the divider-header combination concept was introduced to prevent inconsistencies in flow deviations at the inlet and outlet of the loop. However, this is a special case because the vessel unit has the ability to accumulate mass. A cause of low flow into the vessel is high level which may be caused by some flow through the kick-back leg. A cause of the relief valve working is high pressure caused by no flow out of the divider unit. This gives no flow out of the loop as a cause of low flow into it. This represents a change in the flow deviation and the boundary conditions imposed for the divider-header combination force it to be deleted. The accumulation effect, though, means the cause is valid. Similar logic is applied for configuration B.

The accumulation also affects the deviations when a vessel forms one of the units contained within the divider-header combination loop as in configuration C. Although there are three loops in this example only two cause a problem. The first loop is the one around the pumps. The second and third loops are similar in that they go from one pump, through the kick-back leg, through the vessel and back to the same pump. These latter two loops contain the vessel as one of the interconnecting units. This causes problems because low flow out of the vessel may be caused by low level which in turn may be caused by no inlet flow. This results in a change in deviation and consequently opposite ends of a loop may have different flow deviations.

For this reason the loop identification process has been modified to remove loops which contain any unit where an accumulation of mass can occur.

10.5 Revision To Flow Capacity Specification

As part of the process of improving the user interface of the programs the options given for the capacities have been changed. Originally the values of 0, 50 or 100 were used but some users, especially with three-way divider-header combinations find this confusing because in theory the parallel flow system has $33\frac{1}{3}$% of the capacity in each leg.

To avoid this the number system has been removed and in its place text is used. There are no differences in the results but it is less confusing for the user. The new options are:

none	non-flow leg
shared/some	shared capacity or some flow leg
full	full capacity flow leg

The text option chosen is converted to the numerical capacity by the program.

10.6 Summary

Thorough tests have been carried out on all combinations of two-way divider-header combinations. These are by-pass with no flow, by-pass with flow and parallel flow systems. All deviations into and out of such divider-header combinations have been studied and the synthesis algorithm deals with them correctly (except the duplication of events in a non-flow leg which is dealt with by the cutset analysis programs).

All deviations into and out of multiple-way divider-header combinations with parallel flow are also dealt with satisfactorily.

Preliminary tests have been carried out on the effects of having three pumps in parallel, one being on standby, but although the tree was synthesised with the low flow template it was not easily interpreted and some of the branches had errors in them. Further investigation is needed on this.

10.7 References

1. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1986; pp 8-1–8-20.

2. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 5/1–5/36.

Chapter 11

Modelling Control Loops And Trip Loops

11.1 Introduction

The modelling of individual control and trip loops has been covered in earlier Chapters and by Kelly (1, 2) and Mullhi (3, 4). This chapter looks at modelling systems with multiple control loops and combined control and trip loops. The modelling of multiple trip loops has not caused any problems.

This project has looked at two types of control and protection system. One is a system which has multiple control loops which are closely linked together in such a way as to leave the system over-determined and the other is the combined control and trip system.

11.2 Over-Determined Control Loops

The problem of over-determined control loops has not been addressed by any of the previous workers. The object was to ascertain whether any problems might arise during fault tree synthesis of such a system.

11.2.1 First Example Configuration

Consider the control system shown in Figure 11.1.
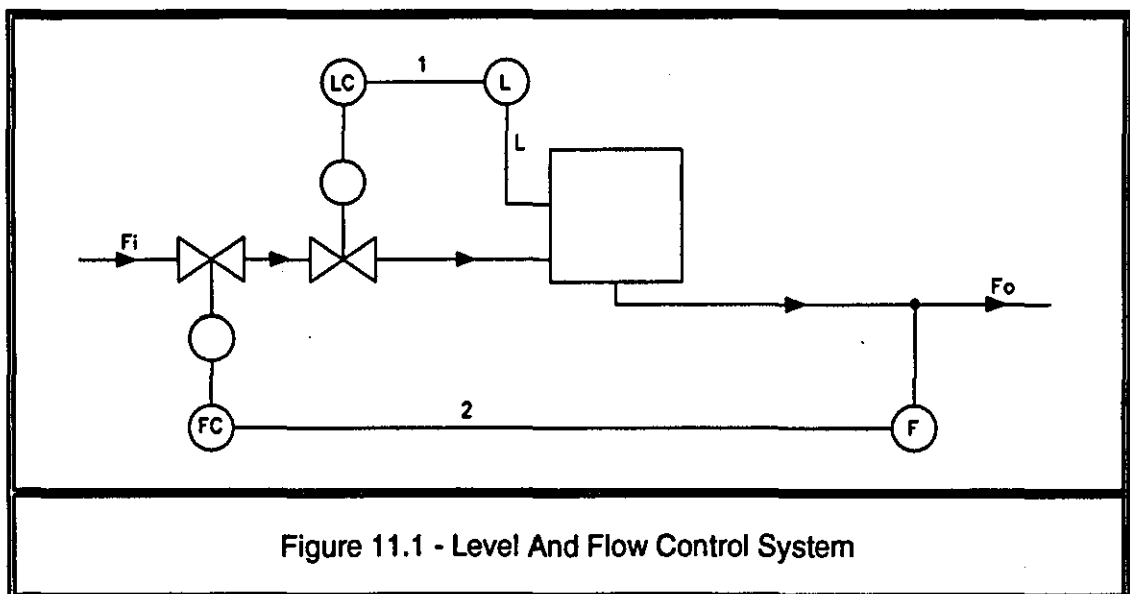
Figure 11.1 - Level And Flow Control System

Loop 1 is a level control loop which regulates the level in the vessel by manipulating the inlet flow rate. A second control system is added (Loop 2) which senses the outlet flow rate and regulates the inlet flow rate accordingly.

The configuration diagram is shown in Figure 11.2.
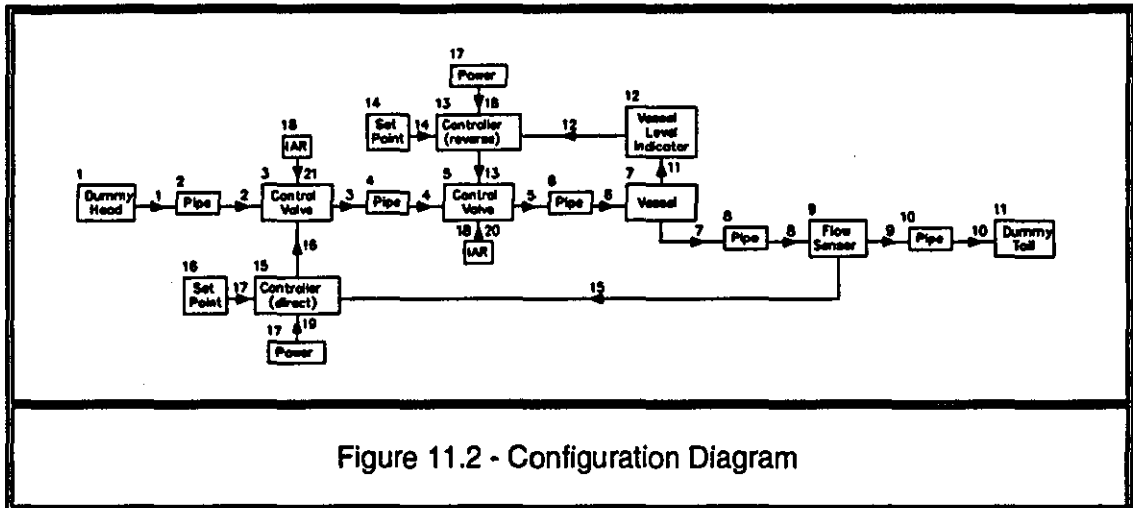
Figure 11.2 - Configuration Diagram

11.2.1.1 System Analysis

If only the flow and associated variables are considered then there are three variables in the system, the flow in Fi, the flow out Fo, and the level L. However, there are four relationships for these in the unit models concerned and the control loops.

$$Fi = F(-L)$$
$$Fi = F(Fo)$$
$$L = F(Fi, -Fo)$$
$$Fo = F(L)$$

This makes the system over-determined because the number of equations is greater than the number of variables.

Similarly by using the directed graph or digraph approach shown in Figure 11.3 it can be seen that the system is over-determined because the level and flow out will be competing with each other.
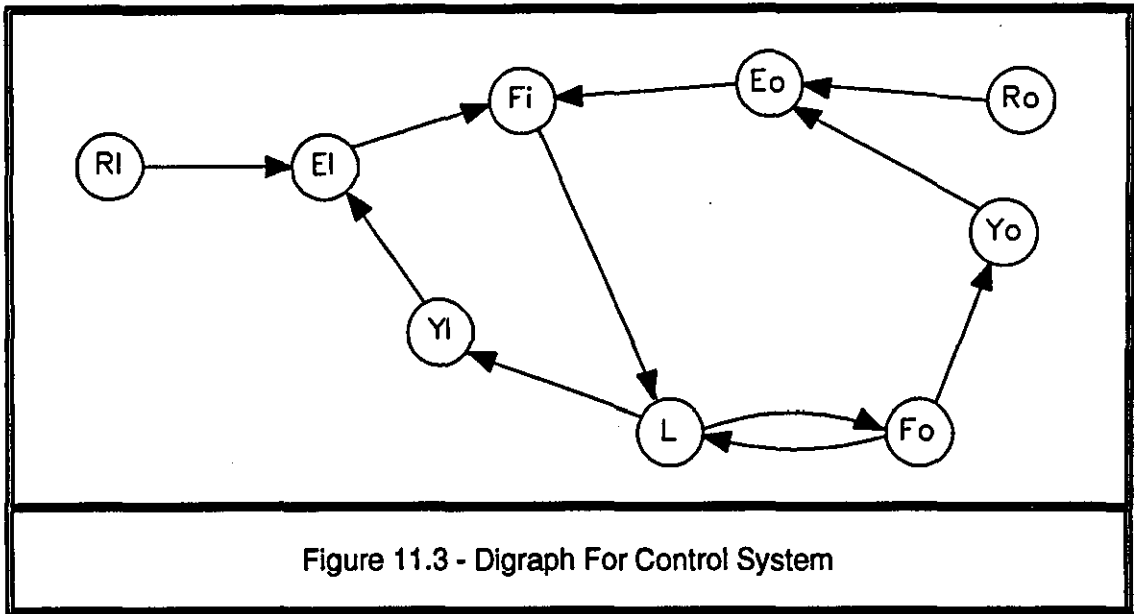
Figure 11.3 - Digraph For Control System

The terms used are:

F - flow
L - level
Y - measurement value
R - set point value
E - error in measurement

i - for inlet
o - for outlet
l - for level

However, this system will work provided that the set points for the level and flow out are compatible. The loops can also compensate for faults in the other provided the fault does not lead to overload.

11.2.1.2 System Fault Trees

Two top events have been chosen for the example, the first is a low level in the vessel and the second is a high flow into the vessel. The respective fault trees are shown in Figures 11.4 and 11.5.
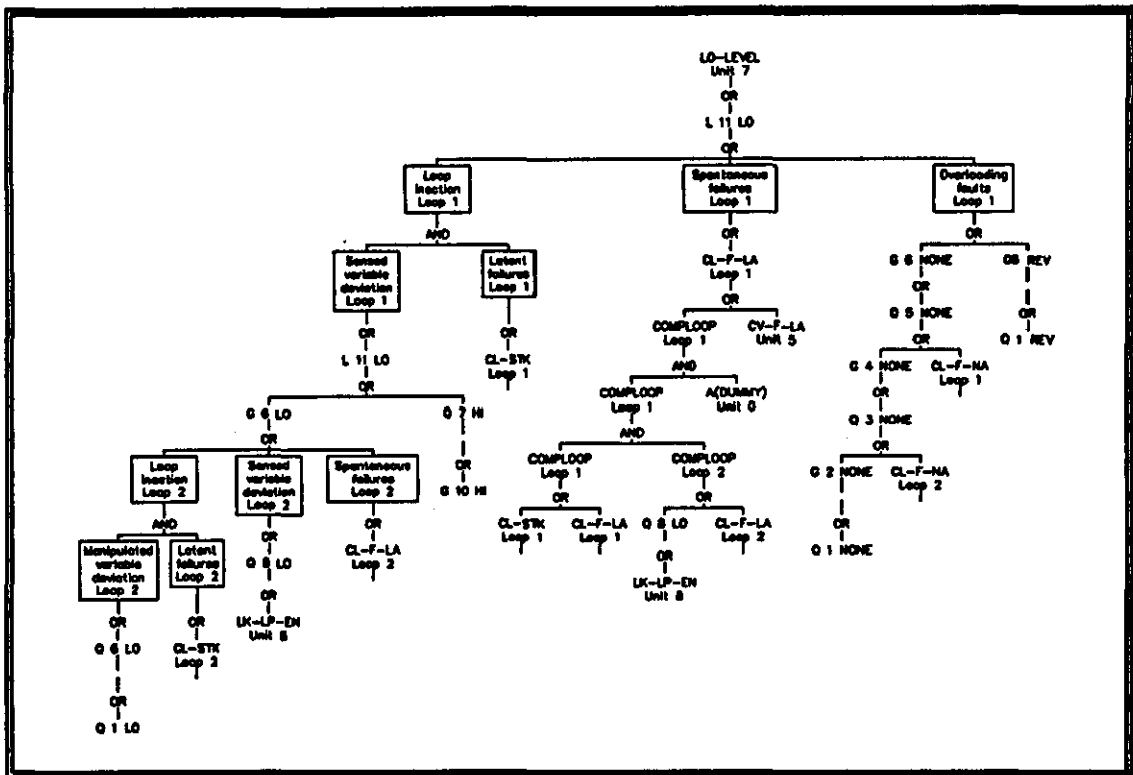
Figure 11.4 - Partial Fault Tree For Low Level
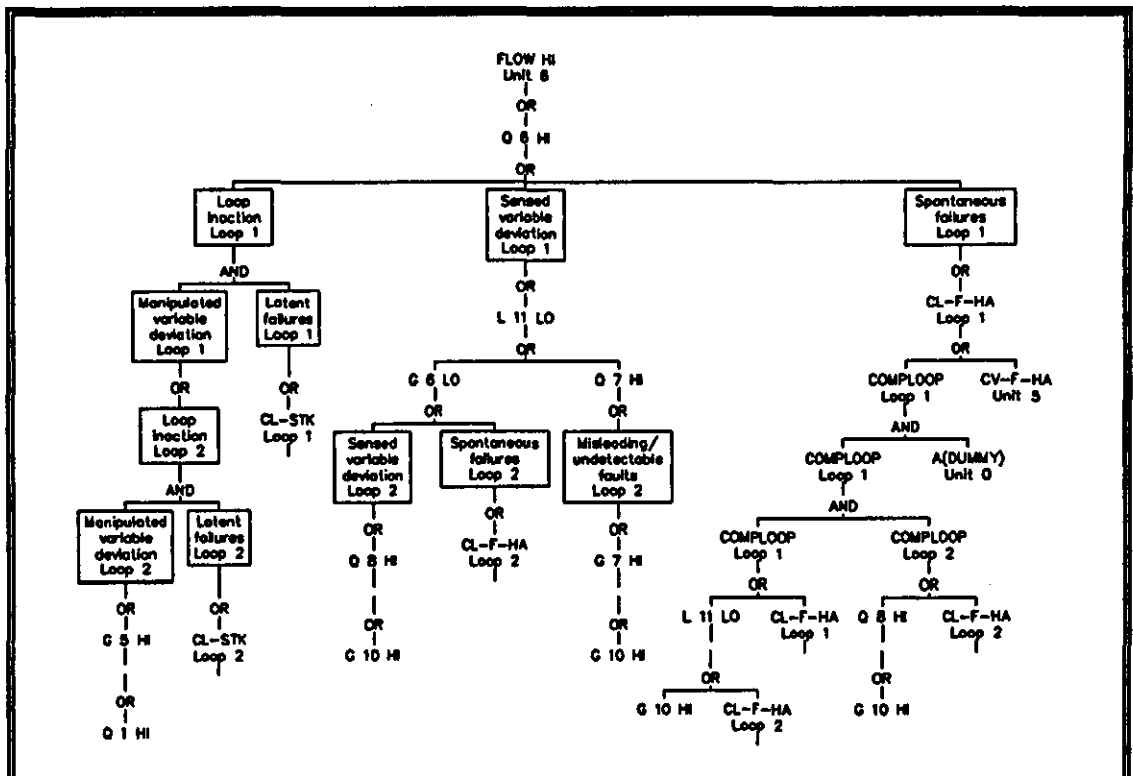
Figure 11.5 - Partial Fault Tree For High Inlet Flow

11-4

Both these fault trees show the use of the compensating control loop feature. This is used when one or more control loops are configured in such a way that if one fails then another can compensate for the fault. In Figure 11.4 the control loop stuck and control loop fails low aperture faults of each loop are ANDed together. The event 'A(DUMMY) Unit 0' is required to prevent the combinations forming lower order cutsets than the combination of latent failure loop 1, latent failure loop 2 and manipulated variable deviation loop 2.

The cutsets for the causes of low level are:

Q 1 REV
Q 1 NONE
CL-F-NA Loop 1
CL-F-NA Loop 2
CV-F-LA Unit 5
CL-STK Loop 1   and G 10 HI
CL-STK Loop 1   and CL-F-LA Loop 2
CL-STK Loop 1   and LK-LP-EN Unit 8
CL-F-LA Loop 1   and CL-F-LA Loop 2
CL-F-LA Loop 1   and LK-LP-EN Unit 8
CL-STK Loop 1   and CL-STK Loop 2   and   Q 1 LO

These show the four over-loading faults to be single order cutsets. The top event will occur if the inlet flow is none or reversed or either control loop fails with a closed valve. Also the level control valve failing with low aperture is a cause. If the level control loop sticks (loop 1) this must be accompanied by high flow out of the system or loop 2 failing with low aperture. Also a leak to a low pressure environment causing low flow in the sensor but high flow out of the vessel and loop 1 failing with low aperture or stuck is a cause. Both the loops failing with low aperture or both sticking and low flow into the system will cause the top event.

The cutsets for high inlet flow to the vessel are:

G 10 HI
CL-F-HA Loop 2
CV-F-HA Unit 5
CL-F-HA Loop 1   and CL-F-HA Loop 2
CL-STK Loop 1   and CL-STK Loop 2   and   Q 1 HI

These show the causes of high inlet flow to the vessel to be high outlet flow or either control valve failing with high aperture. Both loops failing with high aperture or both loops sticking with high inlet flow to the system are also causes of the top event.

11.2.2 Second Example Configuration

Now consider the system given in Figure 11.6. The system is very similar to that in Figure 11.1 except that the flow control valve has been placed on the outlet of the vessel. This severs the influence that the flow controller has on the level.

Figure 11.6 - Second Level And Flow Control System

The configuration diagram is shown in Figure 11.7.
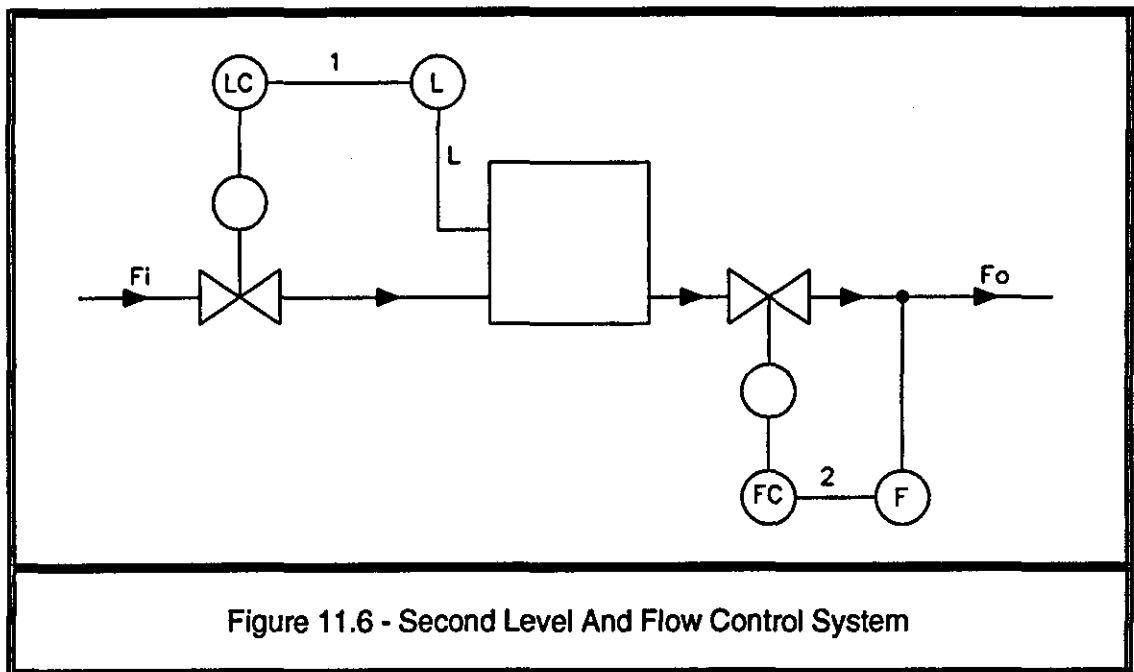
Figure 11.7 - Configuration Diagram
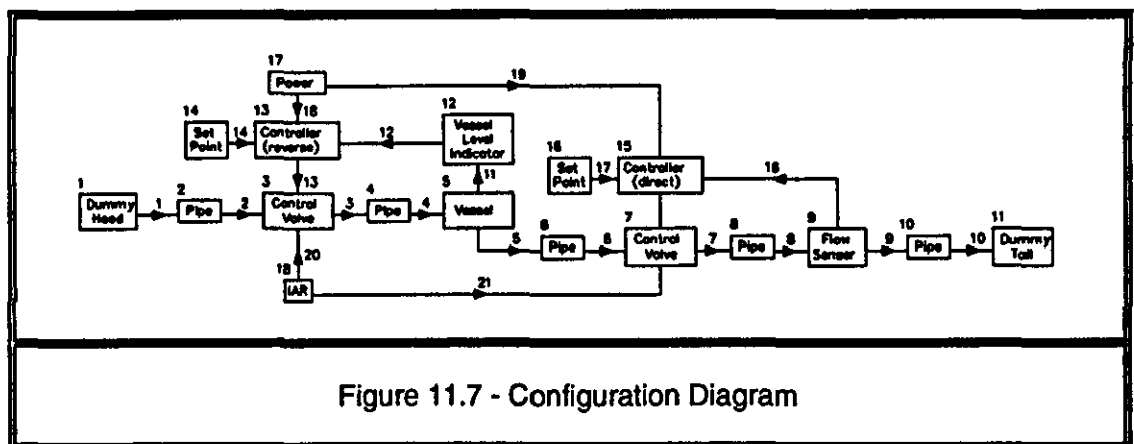
11.2.2.1 System Analysis

If the same models are used as before and only the flow and associated variables are considered then there are three variables in the system, the flow in Fi, the flow out Fo, and the level L. However, this time there are only three relationships for these in the unit models concerned and the control loops.

$$Fi = F(-L)$$
$$L = F(Fi, -Fo)$$
$$Fo = F(L)$$

Therefore the system is not over-determined.

However, the digraph given in Figure 11.8 shows the link between the level and flow out which indicates that the two set points for the control loops must be compatible.
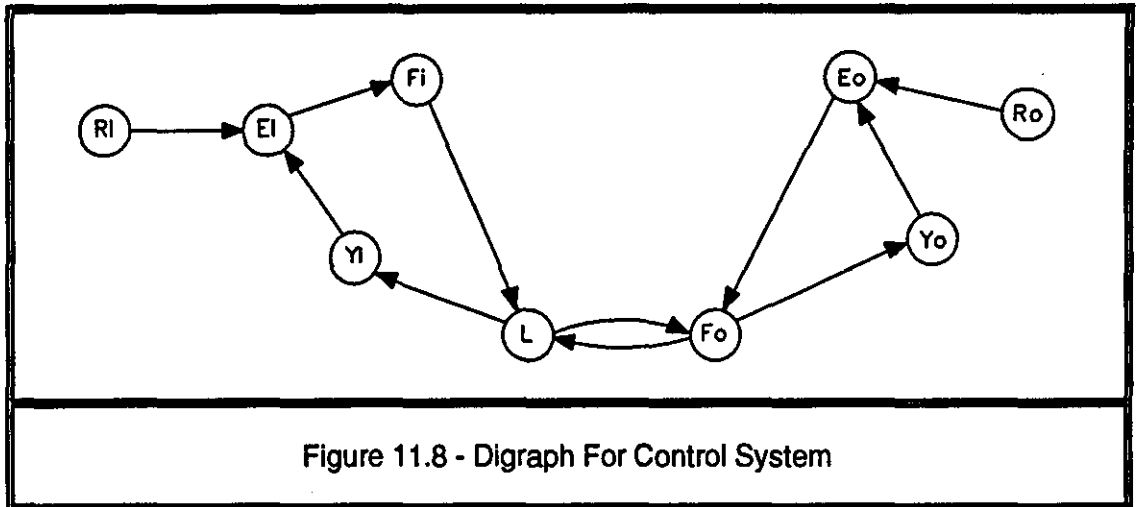
Figure 11.8 - Digraph For Control System

The same nomenclature for Figure 11.3 is used.

This time the two loops cannot compensate for faults in each other.

11.2.2.2 System Fault Trees

The same two top events have been chosen for this example, the first is a low level in the vessel and the second is a high flow into the vessel. The respective fault trees are shown in Figures 11.9 and 11.10.
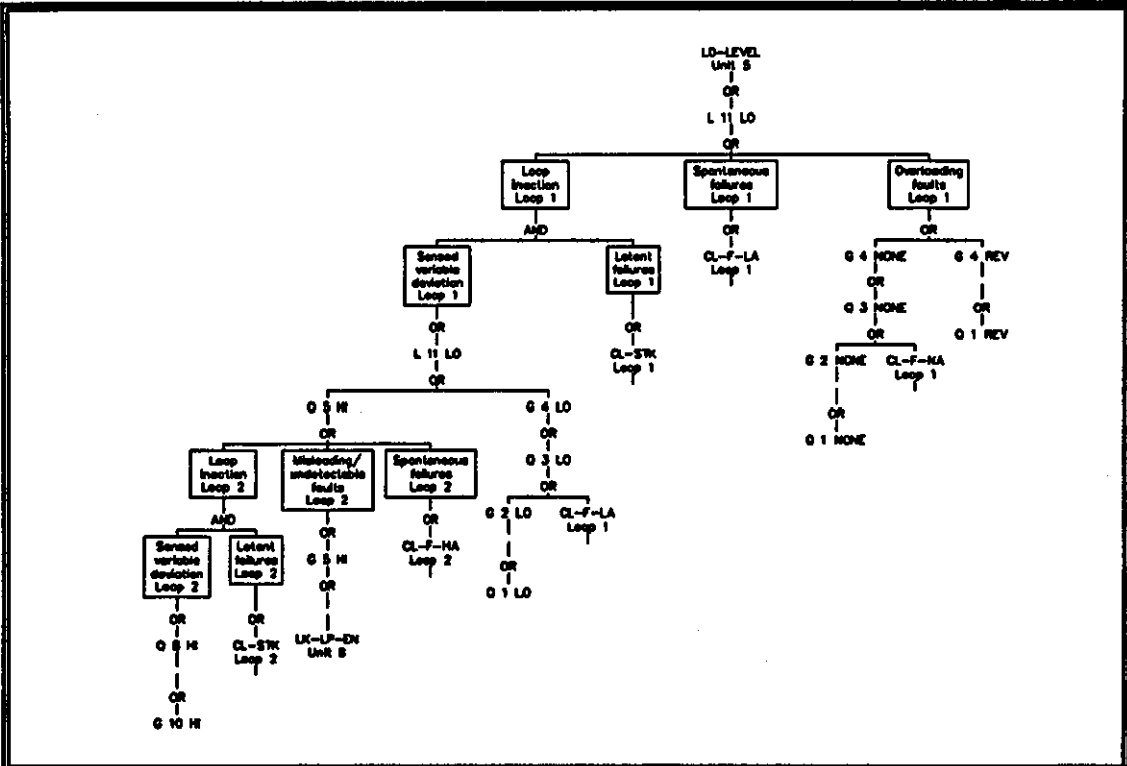
Figure 11.9 - Partial Fault Tree For Low Level
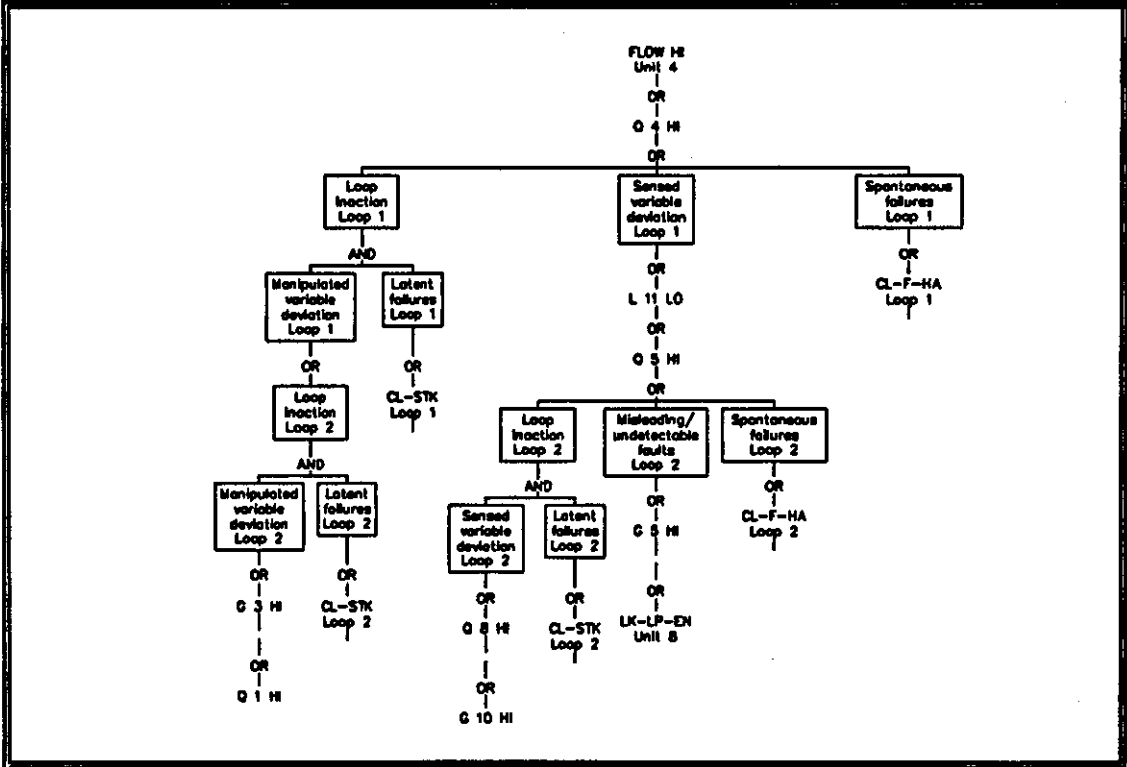
Figure 11.10 - Partial Fault Tree For High Inlet Flow

11-8

The cutsets for the causes of low level are:

Q 1 REV
Q 1 NONE
CL-F-NA Loop 1
CL-F-LA Loop 1
CL-STK Loop 1   and   Q 1 LO
CL-STK Loop 1   and   LK-LP-EN Unit 8
CL-STK Loop 1   and   CL-F-HA Loop 2
CL-STK Loop 1   and   CL-STK Loop 2   and   G 10 HI

These cutsets are very similar to the ones obtained for the configuration in Figure 11.1. The differences can be attributed to the positioning of the flow control valve from the inlet to the vessel to the outlet of the vessel.

The cutsets for high inlet flow to the vessel are:

CL-F-HA Loop 1
CL-F-HA Loop 2
LK-LP-EN Unit 8
CL-STK Loop 2   and   G 10 HI
CL-STK Loop 1   and   CL-STK Loop 2   and   Q 1 HI

Again the cutsets are very similar.

11.2.3 Conclusions

These two examples were chosen specifically to demonstrate the ability of the modelling and synthesis methodology to deal with over-determined control loop systems should the need arise. The examples are very similar in construction and operation except for one factor. The first example, with the two control valves on the inlet stream to the vessel, is over-determined whereas the second, with the control valves on opposite sides of the vessel, is not.

The fault trees produced and the resultant cutsets are very similar so it can be concluded that the methodology is able to deal with configurations involving over-determined control loop systems.

11.3 Combined Control And Trip Loop Systems

The methodology for modelling the protective systems in any configuration has been developed by considering one control loop or one trip loop operating on one variable. This means that in any configuration all the control loops and trip systems must be independent of each other.

This approach is limited in its use especially with more and more complicated process plant control systems being designed. The methodology for dealing with multiple control loops has been developed by Mullhi (3). There are currently no foreseeable problems in modelling multiple trip systems as combinations of the individual trips.

A problem encountered by Mullhi was in modelling combined control and trip loops. These consist of single valve units which act as control valves under normal operating conditions but should an undesired event occur a trip switch operates to slam shut the valves. The rapid closing of the valves is achieved by venting the pneumatic control signal.

Consider the section of plant shown in Figure 11.11.
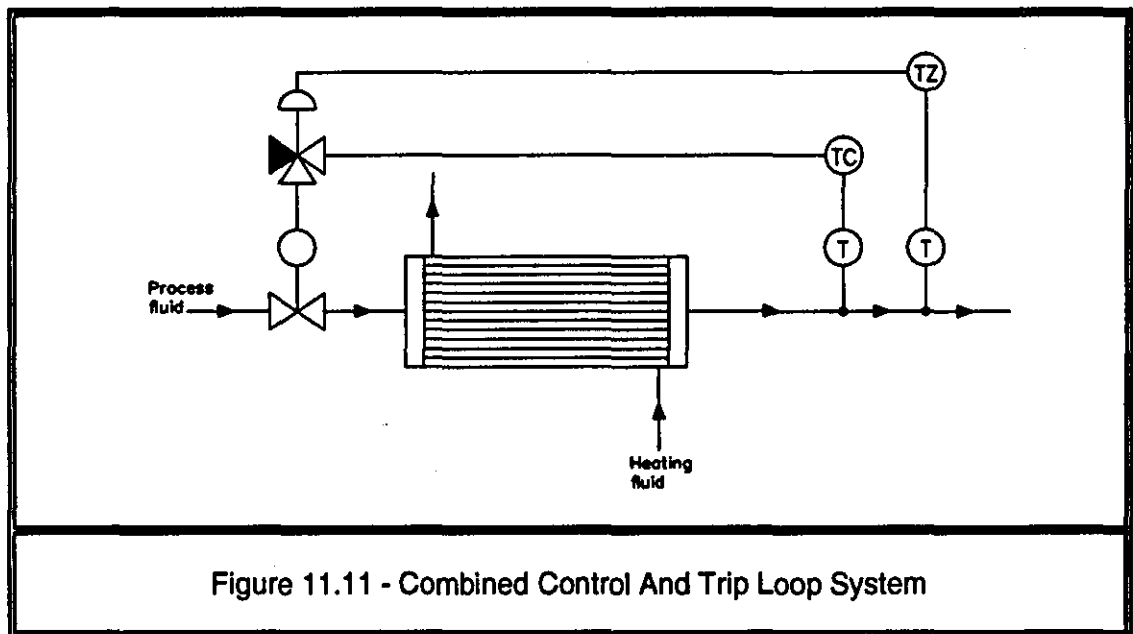
Figure 11.11 - Combined Control And Trip Loop System

A process fluid is supplied from storage and is to be heated before proceeding further down the pipeline. The heat exchanger provides a constant load so the temperature is controlled by manipulating the flow rate of the process fluid. The control/trip valve is pneumatic and the control signal passes through a vent valve. Should the exit temperature be below the minimum temperature then a trip switch on a second temperature sensor operates the vent valve. This vents the control signal and since the control/trip valve is of the air-to-open type the control/trip valve closes rapidly.

The configuration diagram is shown in Figure 11.12.

Figure 11.12 - Configuration Diagram
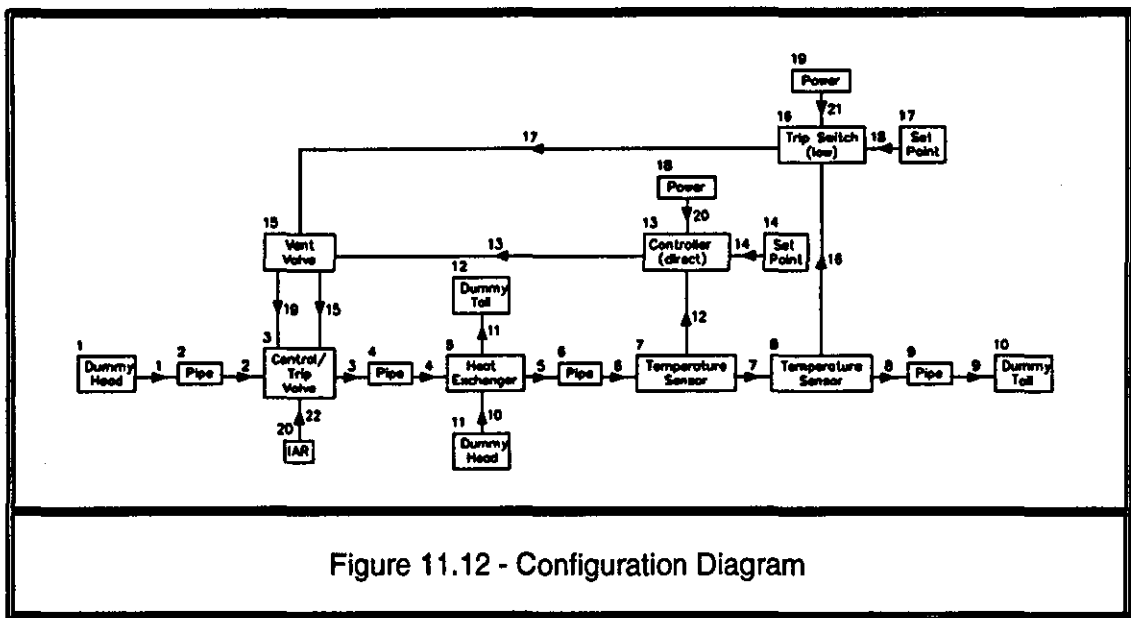
The top event to be examined is low temperature propagating into the pipeline from the heat exchanger.

11.3.1 Modelling And Terminology

The immediate problem with this system is that although the trip switch operates on the vent valve the result of its action is directed to the control/trip valve to shut off the flow. Therefore, the trip valve unit must be specified as the control/trip valve and not the vent valve. It is not possible to specify the vent valve because the methodology requires the valve to be on a process flow stream. There are two signal lines from the vent valve to the control/trip valve to enable the trip loop and the control loop to be modelled correctly.

The control/trip valve must contain the failure modes for both the control loop and the trip loop. The control loop failures pertinent to the top event are control loop stuck CL-STK and control loop fails with high aperture CL-F-HA. The trip loop failure is trip loop functional failure TL-FN-F.

Within the unit model for the control/trip valve are the following event statements for the control loop failures

F CTV-F-HA : CL-F-HA
F CTV-STK : CL-STK

11-11

representing control/trip valve fails with high aperture and control/trip valve stuck respectively.

The trip loop failures are any position where the valve is unable to close. These are control/trip valve fails with high aperture CTV-F-HA, control/trip valve fails with low aperture CTV-F-LA and control/trip valve stuck CTV-STK. The following event statements are used:

   F CTV-F-HA : TL-FN-F
   F CTV-F-LA : TL-FN-F
   F CTV-STK : TL-FN-F

The synthesis algorithm uses these definitions to develop the control loop failures and trip loop failures.

In order for low temperature to propagate out of the system both the control loop and the trip loop must fail. The synthesis of the fault tree is such that the control loop template is added first and then the trip loop functional failure branch is ANDed to it.

11.3.2 Mutually Exclusive Failure States

For the low temperature top event the partial fault tree shown in Figure 11.13 is obtained.
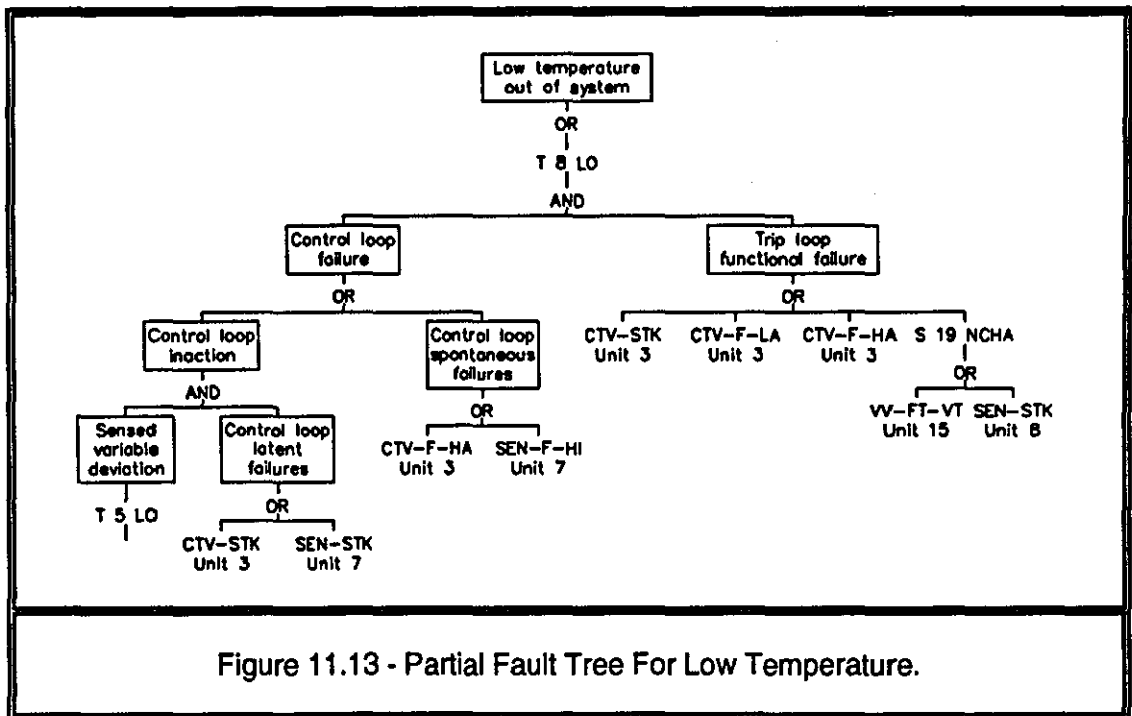
Figure 11.13 - Partial Fault Tree For Low Temperature.

This shows the control loop failures ANDed with the trip loop failures. However, the consequence of this is different failure states of the control/trip valve being ANDed together.

Mullhi regarded this as an incorrect fault tree and he developed a solution to deal with this. The solution was not applied to the main fault tree but to the fault tree developed for use by the analysis program. This involved manipulating the data and restructuring the fault tree such that the trip loop functional failure branch was ANDed with each control loop failure branch. The trip loop functional failure branch was then modified to contain only the failure state of the valve which was the same as that in the control loop branch. This is explained more fully in his thesis.

However, there were two draw-backs to this approach. The first was that the structural rearrangement was not carried out on the main fault tree so the plotted fault tree still remained as before. The second was that the process had not been developed fully and was not complete.

The problem has been re-examined by investigating the operation and results of the analysis program. The fault tree in Figure 11.13 yields the following cutsets:

1	CTV-STK Unit 3	and	SEN-F-HI Unit 7		
2	CTV-STK Unit 3	and	CTV-F-HA Unit 3		
3	CTV-STK Unit 3	and	CTV-STK Unit 3	and	T 5 LO
4	CTV-STK Unit 3	and	SEN-STK Unit 7	and	T 5 LO
5	CTV-F-LA Unit 3	and	SEN-F-HI Unit 7		
6	CTV-F-LA Unit 3	and	CTV-F-HA Unit 3		
7	CTV-F-LA Unit 3	and	CTV-STK Unit 3	and	T 5 LO
8	CTV-F-LA Unit 3	and	SEN-STK Unit 7	and	T 5 LO
9	CTV-F-HA Unit 3	and	SEN-F-HI Unit 7		
10	CTV-F-HA Unit 3	and	CTV-F-HA Unit 3		
11	CTV-F-HA Unit 3	and	CTV-STK Unit 3	and	T 5 LO
12	CTV-F-HA Unit 3	and	SEN-STK Unit 7	and	T 5 LO
13	VV-FT-VT UNIT 15	and	SEN-F-HI Unit 7		
14	VV-FT-VT UNIT 15	and	CTV-F-HA Unit 3		
15	VV-FT-VT UNIT 15	and	CTV-STK Unit 3	and	T 5 LO
16	VV-FT-VT UNIT 15	and	SEN-STK Unit 7	and	T 5 LO
17	SEN-STK UNIT 8	and	SEN-F-HI Unit 7		
18	SEN-STK UNIT 8	and	CTV-F-HA Unit 3		
19	SEN-STK UNIT 8	and	CTV-STK Unit 3	and	T 5 LO
20	SEN-STK UNIT 8	and	SEN-STK Unit 7	and	T 5 LO

The first event is a trip loop failure and the second is a control loop failure. The third event is the initiating variable deviation.

These cutsets form the basic raw cutset data obtained from the fault tree. It is necessary to delete the duplicated events from the cutsets and then delete any non-minimal cutsets. The list becomes:

1	CTV-STK Unit 3	and	SEN-F-HI Unit 7		
3	CTV-STK Unit 3	and	T 5 LO		
5	CTV-F-LA Unit 3	and	SEN-F-HI Unit 7		
8	CTV-F-LA Unit 3	and	SEN-STK Unit 7	and	T 5 LO
10	CTV-F-HA Unit 3				
13	VV-FT-VT UNIT 15	and	SEN-F-HI Unit 7		
16	VV-FT-VT UNIT 15	and	SEN-STK Unit 7	and	T 5 LO
17	SEN-STK UNIT 8	and	SEN-F-HI Unit 7		
20	SEN-STK UNIT 8	and	SEN-STK Unit 7	and	T 5 LO

As can be seen by the outcome of this process, the cutsets no longer contain mutually exclusive failure states. This happens because when the two cutsets with identical events ANDed together are reduced to contain only one of these events the cutset order is also reduced by one. This results in a new minimum cutset which supersedes the others causing the cutsets with mutually exclusive events to be deleted as non-minimal.

11.3.3 Conclusions

It can be concluded from this that for the current example the fault tree synthesis program has the ability to deal with the mutually exclusive failure states but further work is required to establish the generality of this.

However, this highlights that there is perhaps a problem in dealing with combined control and trip loops that needs further investigation. The problem may stem from the control/trip valve model. It appears that the methodology for defining the failure states of the valve for the trip loop must be made compatible with the definitions in the control loop failures.

11.4 Summary

The methodology for modelling control and trip loop systems has been tested on many types of configuration. This includes simple control loops and trip loops to complex over-lapping control systems. To date there are no foreseeable problems.

However, combined control and trip loop systems have caused problems which need to be further investigated. It may require a new methodology for modelling the control/trip valve unit.

11.5 References

1. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1986; pp 6-1–6-43.

2. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology; 1986; pp 7-1–7-54.

3. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 6/1–6/21.

4. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 7/1–7/17.

Chapter 12

Some Fault Tree Synthesis And Analysis Problems

12.1 Introduction

This chapter looks at some of the problems encountered during fault tree synthesis and analysis besides those already mentioned in previous chapters. The problems arise from the experiences of users during this project.

There is the stability of the fault tree to changes in the models used and the adaptation of the fault tree to mimic manually drawn trees. The increased size of the plants modelled and the complexity of the units has resulted in increases in the size of data arrays used in both the unit models and the configuration program. It has also been necessary to carry out further processing on the cutsets generated by the analysis program to remove unnecessary and misleading events. Work is under way in developing a new method of presenting the fault trees to the user by investigation of fault tree drawing packages.

12.2 Fault Tree Stability

The stability of the fault tree refers to the robustness of the tree to changes in the models used. During a small project using the Faultfinder package to develop fault trees for a chemical plant an ever changing fault tree was encountered. The plant modelled was a storage facility and the undesired event being studied was a vessel rupture. The top event model TANK-RUP was created for the system and is shown in Figure 12.1.

```
                    TANK—RUP
                       |
                      OR
        ┌──────────────┴──────────────┐
    SUP—FAIL                       DTROW 1
                                      |
                                     AND
                            ┌─────┬───┴───┬─────┐
                          R5OUT  R6OUT  R7OUT  P8VES
                          NONE   NONE   NONE    HI
```
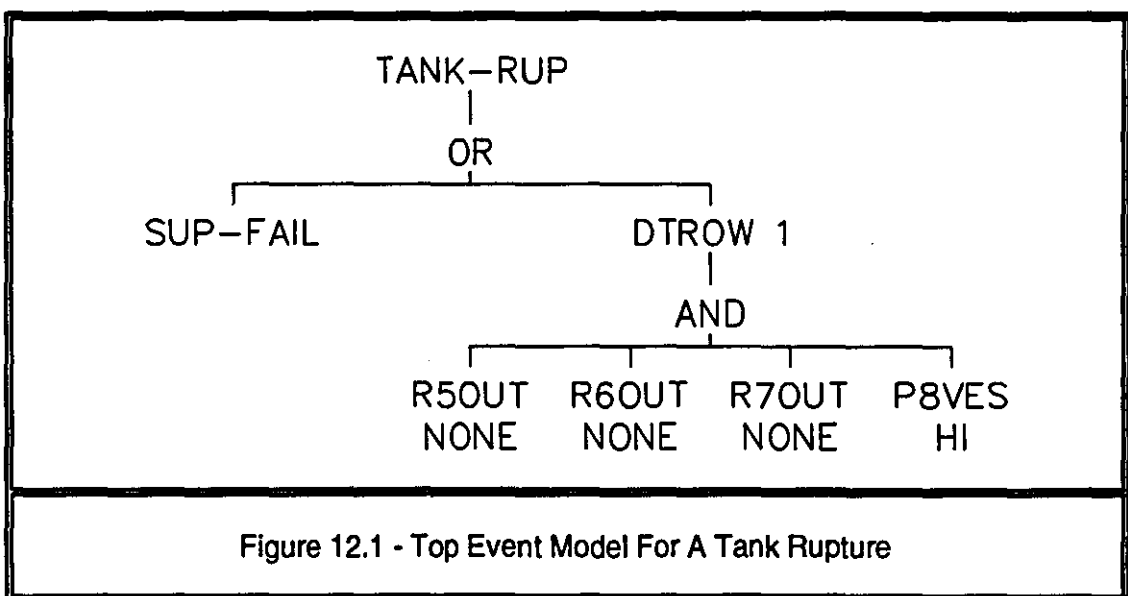
Figure 12.1 - Top Event Model For A Tank Rupture

The short duration of a project can mean that the user has to develop models for the plant very quickly and without fully understanding the methodology. It should be noted that this took place before the vessel model generation routine was developed. A plant configuration and some simple models were produced but when these were altered slightly the structure of the tree was changed. This was particularly apparent when composition and pressure variables were used in the same model.

The programs were run and the fault tree produced. Examination of the fault tree revealed that the R5OUT and R7OUT branches had not been developed. The cause of this was the internal consistency checks which identified some events in the branches as 'certain' events resulting in the entire branch being deleted as certain. However, it was noted that the vessel model being used was not the one intended which was remedied by changing the configuration library number and re-running.

This solved part of the problem but a fault tree could not be obtained because the program entered a continuous loop during synthesis. The vessel model was examined for faults and more event statements were added and a propagation equation altered to prevent the looping. This time the fault tree was synthesised with all the branches fully developed.

This has highlighted a common area where inexperienced users have encountered problems with

    a) incorrect fault trees and
    b) unstable fault trees.

The cause of this stems from the users trying to

    a) select a unit model and top event model such that a mismatch is obtained and
    b) use incomplete unit models.

The fault tree produced depends on the data supplied to the program. If, for example, the top event does not match up to the unit to which it is attached then spurious results can occur. It is also important to ensure that any models developed are complete and correct. At first it may not seem necessary to include all variables in a model when only one is of interest but the synthesis package may trace faults to other variable deviations in the model. The implementation of the vessel model generation routine is a powerful aid for this and helps to maintain a user-friendly facility.

12.3 Risk Assessment Fault Trees

The Faultfinder approach to fault tree synthesis is to trace variable deviations from their effects to their sources, usually basic faults. However, risk assessment encompasses not only variable deviation causes but mechanical faults as well. Therefore, fault trees produced manually for risk assessment include chains of these mechanical faults linked together.

An example of such a fault tree is shown in Figure 12.2 which has been adapted from one developed by Cremer and Warner Ltd (1) for a chlorine storage vessel.

Figure 12.2 - Partial Fault Tree For Risk Assessment
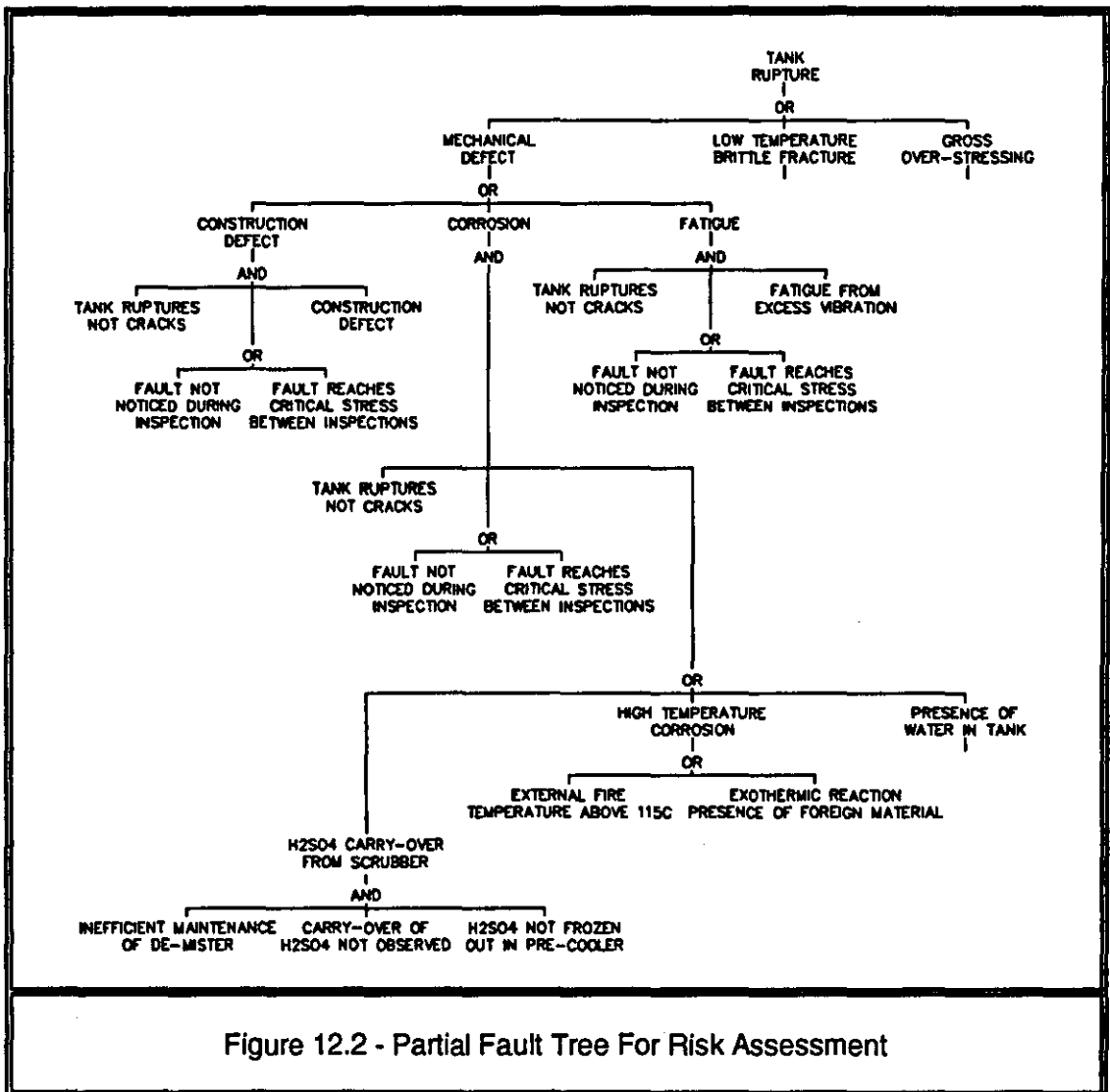
Only the lower branches for the mechanical defects fault are shown. The mechanical defects are split into three types of causes, construction defects, corrosion and fatigue. These all have sub-trees made up of other faults.

While the Faultfinder methodology allows variable deviations to be linked together, basic faults, which include mechanical faults, only appear at the end of the fault tree branches.

It is desirable to be able to link basic events together to form a failure branch. An example chain of basic events is

> tank rupture, caused by
> support failure, caused by
> over-stressed support, caused by
> design error.

Attempts were made to include the chain as part of the unit model using the intermediate event status. However, this failed because the synthesis program ignored these events such that they never appeared in the generated fault tree. The reason for this is that the top event initialises which variable deviations to trace. The chain of basic faults are causes of a particular event and not a variable deviation, therefore, they are 'over-looked'.

The majority of the mechanical fault branches appear near the top of the tree and are generally associated with the immediate causes of the top event. Therefore, it is reasonable to incorporate the failure branches into the top event.

The top event model library contains a complete set of context independent models which can be applied to pipe type models or vessel type models. These models are used to initiate fault tracing of a single variable or a related pair of variables.

It may be possible for a 'package' of faults, such as the mechanical faults, to be contained in the model library and used when required to connect to an existing top event to produce the risk assessment type fault tree. The faults contained in the 'package' must be self-contained. These faults can have other faults as causes but must exclude any faults caused by variable deviations, such as corrosion caused by an impurity.

However, this approach has been discarded in favour of encapsulating the entire top event in one model. This is also necessary if the top event initiates more than one type of variable deviation trace. In the current example the variables are high concentration and high pressure.

The linking together of the basic faults in the top event model is achieved using the event statements and decision tables and the intermediate event status. However, in the present form the program requires the mechanical faults to be entered in a particular order. The faults are required in an ascending order from the bottom row of the tree. If the order is altered, the fault tree will not contain the basic faults.

12-4

As an illustration, the fault tree in Figure 12.2 can be represented on the Faultfinder package in the form of Figure 12.3.
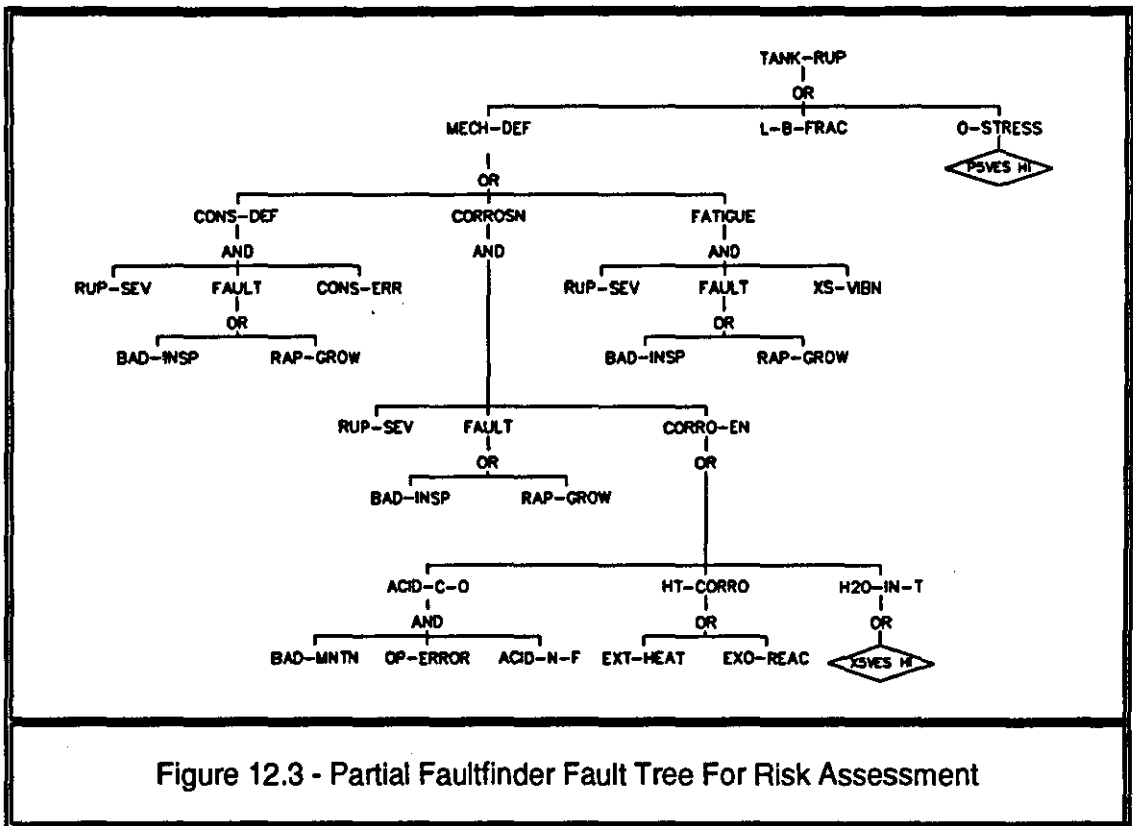
Figure 12.3 - Partial Faultfinder Fault Tree For Risk Assessment

The events enclosed in diamonds are those which initiate the Faultfinder package to start the synthesis process.

The following event statements and decision tables are required:

F BAD-MNTN F OP-ERROR F ACID-N-F T ACID-C-O (bad maintenance)

(operator error)

(acid not frozen out)

F EXT-HEAT : HT-CORRO   (external heat)

F EXO-REAC : HT-CORRO   (exothermic reaction)

V X5VES HI : H2O-IN-T   (high concentration)

I ACID-C-O : CORRO-EN   (acid carry-over)

I HT-CORRO : CORRO-EN   (high temperature
                                                      corrosion)

I H2O-IN-T : CORRO-EN   (water in tank)

F BAD-INSP : FAULT   (bad inspection)

F RAP-GROW : FAULT   (rapid growth of crack)

```
F  RUP-SEV  I  FAULT  F  CONS-ERR  T  CONS-DEF      (ruptures severely)
                                                    (fault)
                                                    (construction error)
F  RUP-SEV  I  FAULT  I  CORRO-EN  T  CORROSN       (corrosive environment)
F  RUP-SEV  I  FAULT  F  XS-VIBN  T  FATIGUE        (excessive vibration)
I  CONS-DEF : MECH-DEF                              (construction defect)
I  CORROSN : MECH-DEF                               (corrosion)
I  FATIGUE : MECH-DEF                               (fatigue)
V  P5VES HI : O-STRESS                              (high pressure)
I  MECH-DEF : TANK-RUP                              (mechanical defect)
I  L-B-FRAC : TANK-RUP                              (low temperature brittle
                                                       fracture)
I  O-STRESS : TANK-RUP                              (over-stressing)
```

Faultfinder limits the basic fault names to eight characters in length so abbreviations have to be used. The text in brackets translates these.

The 'FAULT' and 'CORROsive-ENvironment' events have been added as intermediate events because in the original fault tree two logic gates were linked together. Faultfinder requires a named event between all logic gates.

A simpler example of this type of top event is the ruptured tank, RUP-TANK top event in the model library. The partial fault tree for this top event is given in Figure 12.4.
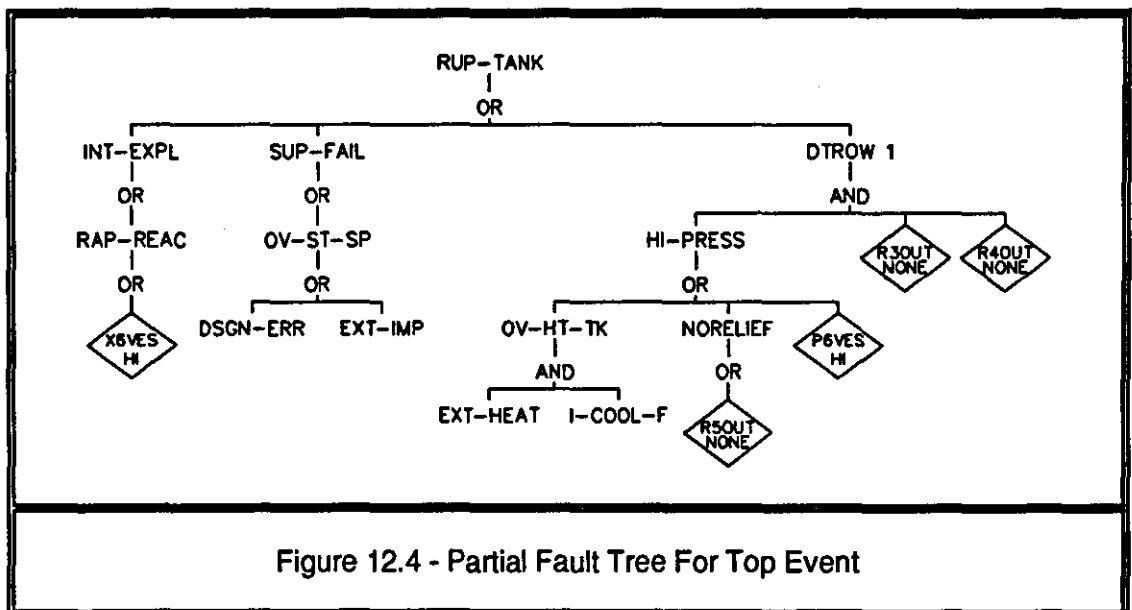
Figure 12.4 - Partial Fault Tree For Top Event

The event statements and decision tables are:

    V X6VES HI : RAP-REAC   (high concentration)
    I RAP-REAC : INT-EXPL   (rapid reaction)
    F DSGN- ERR : OV-ST-SP   (design error)
    F EXT-IMP : OV-ST-SP   (external impact)
    I OV-ST-SP : SUP-FAIL   (over-stressed support)
    F EXT-HEAT AND F I-COOL-F : OV-HT-TK   (external fire,
                                                               internal cooling failed)

    V R5OUT NONE : NORELIEF   (no relief)
    I OV-HT-TK : HI-PRESS   (over-heated vessel)
    I NORELIEF : HI-PRESS   (no relief)
    V P6VES HI : HI-PRESS   (high pressure)
    I INT-EXPL : RUP-TANK   (internal explosion)
    I SUP-FAIL : RUP-TANK   (support failure)
    I HI-PRESS V R3OUT NONE V R4OUT NONE T RUP-TANK

This is for a reaction vessel with two inlet ports and two outlet ports. Port 5 is the relief port and port 6 is the vessel port for the internal variables.

The model states that a high concentration in the vessel will cause a rapid reaction, which will cause an internal explosion causing the vessel to rupture. A design error or an external impact will cause an over-stressed support, which will cause the support to fail, also causing the vessel to rupture. If there is an external fire and the internal cooling fails the vessel will become over-heated causing a high pressure. Another cause of high pressure is no relief through the relief port or the variable deviation high internal pressure. High pressure coupled with no relief through either of the outlet ports will cause the vessel to rupture.

Before synthesis is started the first few rows of the fault tree will be made up of the minitrees describing the top event. There are five variable deviations to be traced for this top event: no relief at either outlet port, no relief at the relief port, high internal pressure and high internal concentration. The initial causes of all these events will be found in the reaction vessel model to which the top event is connected.

This process now allows computer synthesised fault trees to be compared with manually drawn fault trees.

12.4 Dimensionality Aspects

The development of the fault tree synthesis package has been carried out using real industrial examples. As the capability of the facility has improved, the examples have become larger containing many units and connections. The size of the data arrays used in all the programs making up the synthesis package was initially set many years ago and was determined by the size of the computer. With the advances in computing technology this is no longer a restriction.

However, uncontrolled increases in data storage result in a greatly reduced processing speed. In some cases the data array sizes have been reduced. In general the following changes have been carried out

Unit model generation: Increase in overall size of modelling arrays.

Top event model generation: Decrease in overall size of modelling arrays.

Secondary event modelling: Decrease in number of different models possible.

Configuration size: Increase in number of units to 400
                     Increase in number of connections to 500
                     Increase in number of control and trip loops to 30 each
                     Decrease in number of divider-header combinations to 100
                     Decrease in number of secondary failures.

The synthesis program was initially set up to handle up to 9000 individual events and so far not even half of these have been used for any of the test configurations. Therefore an increase in the synthesis ability has not been necessary.

However, as expected, an increase in configuration size and complexity has resulted in larger fault trees. It has been found necessary to increase the number of transmissive, basic and diamond events stored by the synthesis program for the cutset analysis program. These have been increased to the same size as the fault tree contained within the synthesis program.

With increases in the size of the fault tree produced the time required for synthesis also increases. However, the processing time only depends on the size of the initial, raw, synthesised fault tree. A large configuration may involve a large fault tree and a correspondingly high processing time but after rationalisation the final fault tree may be relatively small. Equally, the same configuration but with a different top event may result

in another small fault tree which is synthesised very quickly. The processing time on a DEC Microvax II can be roughly estimated to 1 minute for every 500 minitrees added to the initial fault tree. This is also dependent on the speed of the computer.

Larger and more complex fault trees also result in higher processing times for the fault tree analysis program. Experience has shown a rapid increase in analysis time with increasing fault tree size. It is not possible to estimate the processing time from the fault tree but experience has shown the time to range from a few seconds up to 4 minutes for a fault tree yielding 2500 cutsets.

The time for processing the cutsets has been estimated to be in the form of a geometric expression involving the number of cutsets contained in the initial results. For the DEC Microvax II computer used in this project the expression used to estimate the time, in seconds, is:

$$\left(\frac{n^2}{3000}+\frac{n}{100}\right)*\frac{7}{4}+30$$

where 'n' is the number of cutsets in the fault tree. This is also dependent on the work load. This gives process times of 1/2 minute for 100 cutsets, 3 minutes for 500 cutsets and just over 10 minutes for 1000 cutsets.

However, the analysis and cutset processing times can be decreased independently of the synthesised fault tree by the user by reducing the maximum order of the cutsets required. The analysis program has a built in mechanism for this and has been altered to read in the maximum order value with the fault tree input. The synthesis package has been changed to provide the maximum order value and the means to set this appears as one of the options with the top event specification in the configuration data input program MASTER.

The package is currently stored on a small mainframe type computer (DEC Microvax II) with a high memory and hard disk capacity. This has posed no restrictions to the operation of the programs. With the advances in personal computer (PC) technology it is hoped to be able to produce a compact PC version. The present size of the package is four megabytes for storage. This includes all the modelling, synthesis, drawing and analysis programs, the unit model and top event model libraries and a complete worked example. An additional one megabyte of storage space is recommended for the users own examples and models but this can be reduced by deleting old synthesised trees but keeping the configuration files. The largest program requires a memory capacity of 250 kbytes so it is clearly possible to develop a PC version of the Faultfinder package.

12.5 Fault Tree Presentation

The basic method for producing a fault tree from the package is on a line printer in the form of text characters. The synthesis program provides a drawing file which contains all the transmissive, basic and diamond event names along with co-ordinates for each based on a 250 by 130 grid. The connectivity of the events is contained in a separate table in the file.

A fault tree drawing program reads in this data file and constructs a text file containing the tree. Each event name is made up of a 12 character length string split over 3 lines (ie. 4 characters per line). The actual field length is 5 characters. Below each event is a gate type occupying 3 lines and a connectivity line, made up from asterisks (*) occupying 3 more lines. The connectivity lines may be stretched to enable the tree to be drawn in the available width.

The file can be plotted on a 132 character line printer. An example of the type of fault tree output is shown in Figure 12.5.

As can be seen this is not very clear but it is a very quick and easy method of obtaining a fault tree.

A program was written to generate a plotted fault tree from the data using the GINO graphics routines. The program worked successfully but the facility was machine and software dependent and both are no longer available. Although the output was easy to interpret the synthesis of larger fault trees resulted in long lengths of paper similar to that obtained from the text print-out.

Ideally the fault tree would be more presentable if it could be produced on a regular sheet of paper. The major cause of the long length of paper required to plot the fault tree is the single event intermediates, ie. the events which only have one cause. These are characteristic of the systematic tracing of variable deviations from effect to source encompassing every unit and connection in between as used in the computer generation of fault trees. However, the Faultfinder package has the ability to suppress these intermediate events, thereby reducing the length of the tree.

Another method is to use the fault tree generated for the fault tree analysis program. The tree only contains the events immediately under multi-input logic gates, ie. the single event intermediates are removed. This approach is being developed with the help of British Gas and a program called FTPLOT. This program uses the input file for FTAP, the fault tree analysis program, to plot a fault tree in an 'A size' format. The input file has had to

be modified to contain the text for each event. This data is placed at the end of the file and does not affect the operation of the analysis program. It is hoped that the plotting program can be used to produce fault trees which are easy to comprehend.
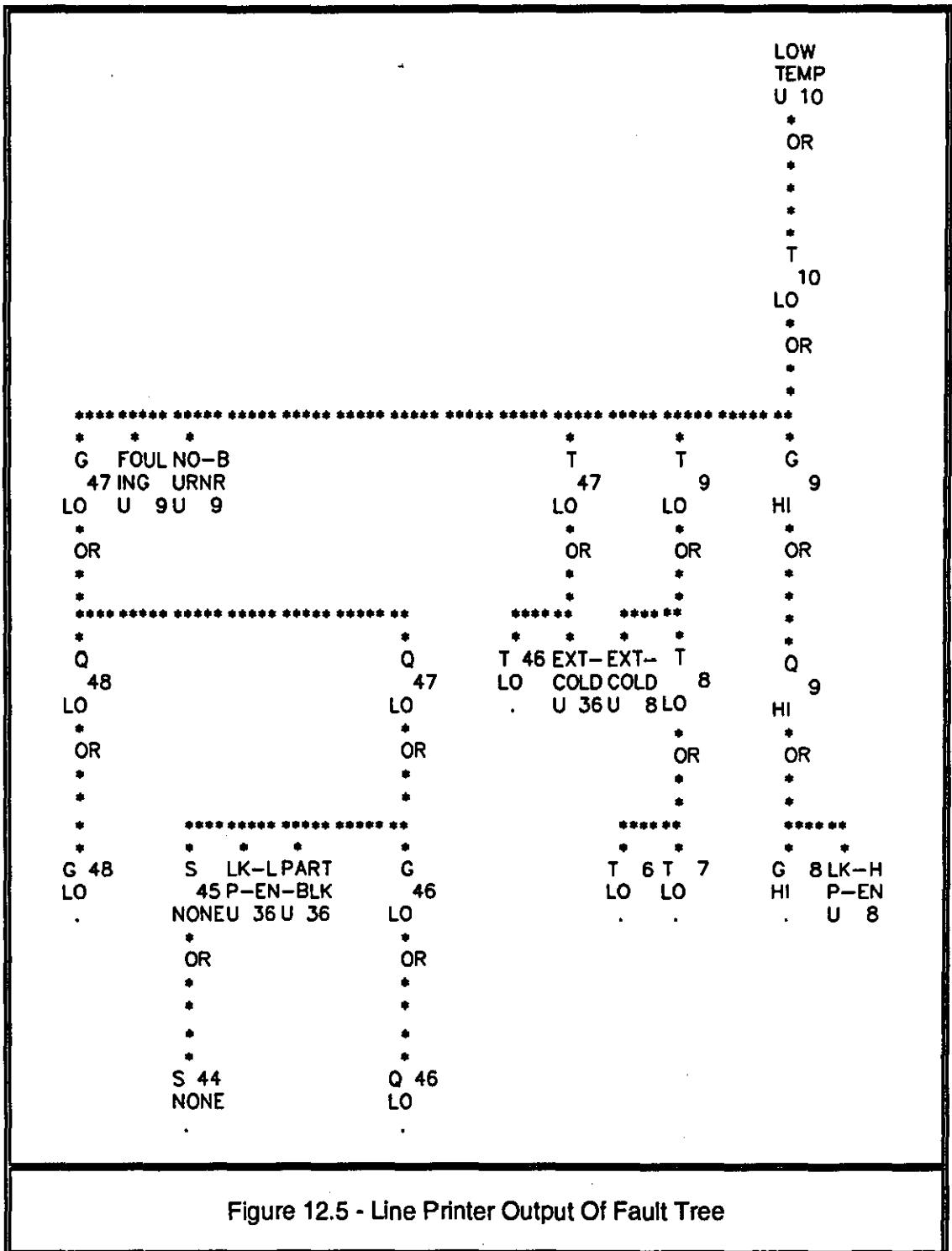
Figure 12.5 - Line Printer Output Of Fault Tree

12.6 Removal Of The 'Normally Working' Events

The synthesis methodology has the ability to model common mode aspects of utility supply failures. For example, a batch of control loops may be connected to the same power supply which when disrupted results in the loss of all the loops.

However, there are instances where it is necessary to state explicitly that the utility must be working for events further down the branch to have an effect. This was first recognised by Trenchard (2) in his alarm diagnosis work. The approach used was to introduce the 'normally working' events into the unit models.

This was later expanded by Mullhi (3), who incorporated the 'working normally' events into the fault tree at the synthesis stage. Consider a simplified version of the plant layout used by Mullhi shown in Figure 12.6.
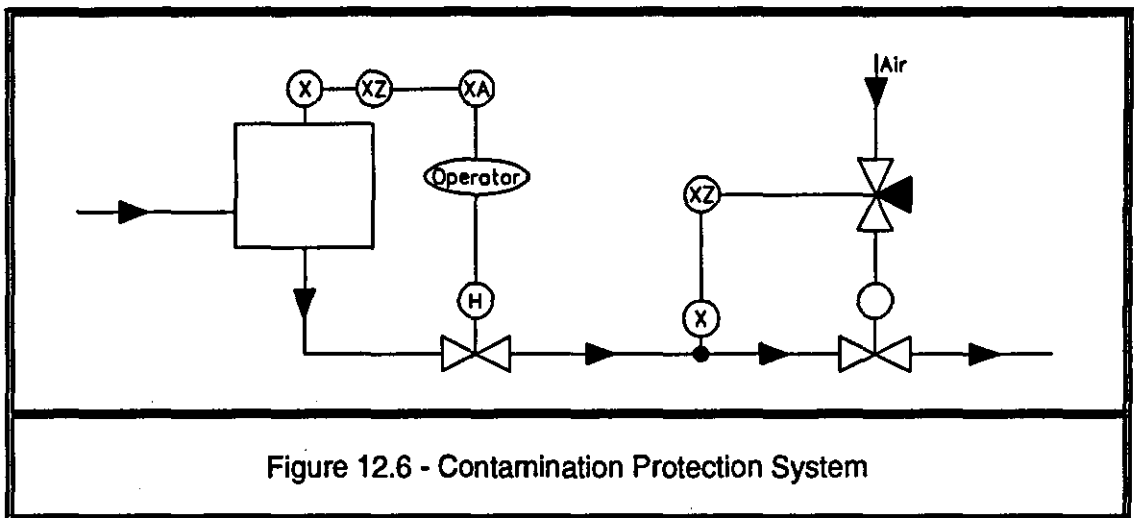
Figure 12.6 - Contamination Protection System

The plant consists of an effluent holding vessel which discharges into a river. To prevent contamination of the river two protective devices are used. An alarm connected to a concentration sensor in the vessel alerts the operator to shut the hand valve on the exit from the vessel. There is a back-up trip system on the exit pipe which operates on a high concentration signal from a sensor. The second trip operates a solenoid valve which vents the air from the trip valve slamming it shut.

The configuration diagram is shown in Figure 12.7.
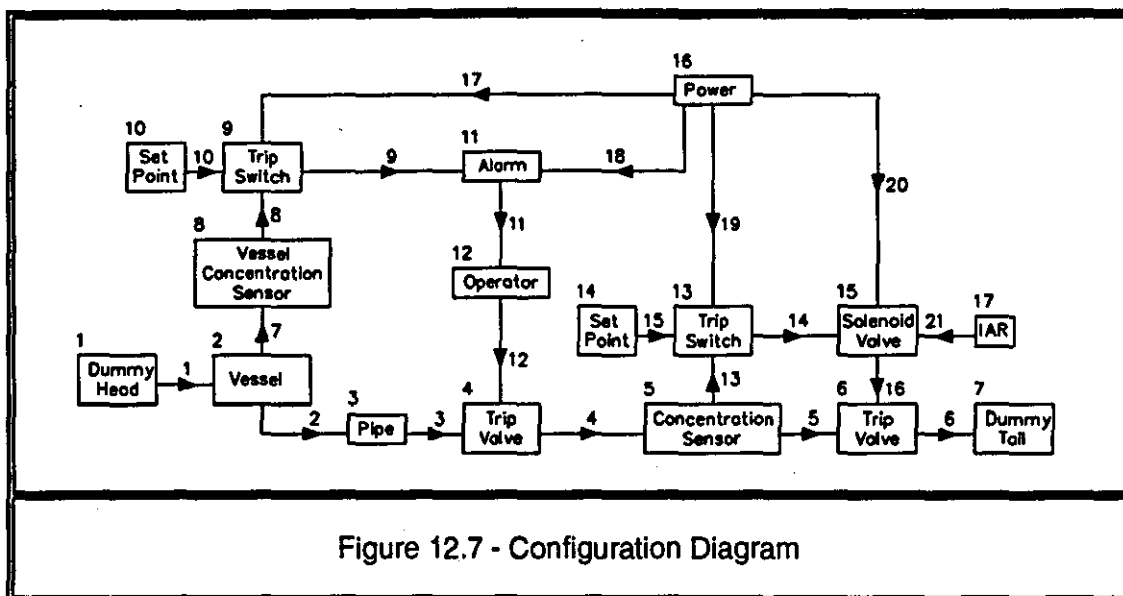
Figure 12.7 - Configuration Diagram

It should be noted that in order for the alarm system to be modelled as a trip loop a trip switch is required between the alarm and the sensor. This is because the alarm is modelled as a simple device which operates should it receive a signal from the trip switch. This allows the failures of both 'trips' to be modelled as trip loop functional failure TL-FN-F branches. For the purpose of this example both the concentration sensors are without a separate power supply.

The protection is a fail safe mechanism because should the air supply fail the trip valve will shut and if the power supply should fail then the solenoid valve will de-energise venting the air, thus shutting the trip valve.

There are two 'trip' mechanisms, trip 1 is the alarm and operator system and trip 2 is the automatic system. A cause of trip 1 failing is the power supply failing, because the alarm and the trip switch will both be inoperative. However, the loss of power will not cause a toxic release because the second trip system will close off the flow unless either of its valves are stuck. In order for the other failures of the second trip system to be incorporated into the fault tree correctly they must be ANDed with a working utility (power) supply.

The partial fault tree is shown in Figure 12.8.
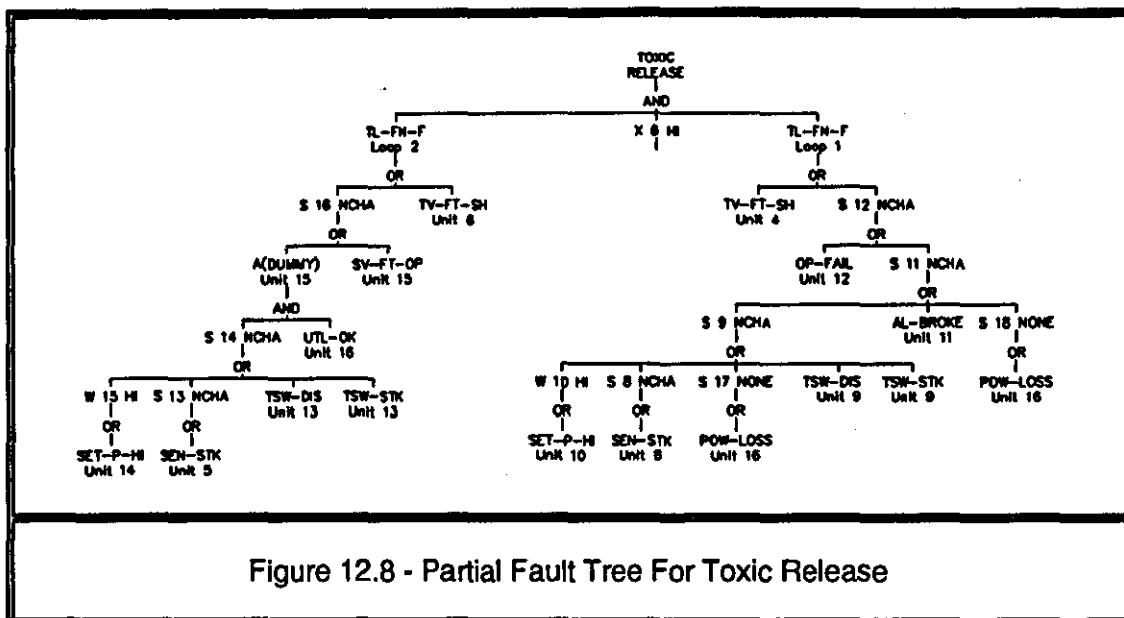
Figure 12.8 - Partial Fault Tree For Toxic Release

The fault tree can be split into three sections. The middle section deals with the causes of the high concentration and has not been developed since it does not form part of this discussion. The right and left branches deal with the functional failures of trip loops 1 and 2 respectively.

For trip loop 1 there are two points where the loss of power may cause the trip to fail. The first (ie. the highest occurrence in the branch) is a cause of the alarm failing and the second is a cause of the trip switch failing.

For trip loop 2 a cause of no change in the signal between the trip switch and the solenoid valve (S 14 NCHA) is a loss of power to the trip switch. This would give a similar sub-tree structure as that for trip loop 1. However, it has been noted that a power loss cannot cause the trip to fail. Therefore, inclusion of power loss as a cause of failure of trip 2 is incorrect. In fact the power supply must be working for the failures of the trip switch, sensor and set point units to have an effect. Hence the event S 14 NCHA must be ANDed with UTIL-OK to represent a working power supply.

The inclusion prevents the analysis program listing, for example, power loss AND trip switch stuck unit 13 as a cutset. Instead the cutset becomes power loss AND trip switch stuck AND utility working. The post-cutset generation program written by Mullhi then removes the cutset because it has two mutually exclusive failure states of the same unit.

However, a problem has resulted from the use of this procedure. Since the approach is systematic it very often results in many occurrences of the UTIL-OK event in the fault tree. The problem with this is the extra AND gates in the fault tree which greatly increase the

orders of the cutsets. Experience has shown this to be very misleading with large order cutsets being ignored when in fact once all the UTIL-OK events have been removed they reveal a serious third order cutset.

It is desirable to remove these 'working normally' events from any fault tree analysis output so that a concise list of fault events can be produced. The probability of such events is very close to one and can be assumed to be unity. If these events remained in the cutsets the overall probability of a top event will be reduced, so by removing them the actual computed value will be higher.

The removal of the 'working normally' events is achieved by using the post-cutset generation program. This program was written by Mullhi to remove the mutually exclusive failure states and to check the consistency of the cutsets. After all the deletions have been carried out the program checks all the remaining cutsets for minimality and re-packs them into an ascending cutset order.

The algorithm written to remove the 'working normally' events is inserted before the minimality checks but after the removal of any mutually exclusive failure states. The program examines each cutset and removes any event with '-OK' in it.

12.7 Summary

This chapter has highlighted some of the problems in synthesis and analysis encountered during this project. The work is aimed at making the package more adaptable to both the user and the types of synthesis required.

12.8 References

1. Risk Analysis Of Six Potentially Hazardous Industrial Objects In The Rijnmond Area, A Pilot Study : A Report To The Rijnmond Public Authority; D. Reidel Publishing Company; 1982; pp 491.

2. Trenchard A. J.; Process Plant Alarm Diagnosis Using Synthesised Fault Tree Knowledge; PhD Thesis, Loughborough University Of Technology; 1990.

3. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 8/8–8/14.

PART 3

Chapter 13

Worked Examples

13.1 Introduction

This chapter covers some of the more substantial examples looked at over the duration of the project. The work covered in this thesis deals mainly with the modelling aspects of computer-aided fault tree synthesis. Therefore, although fault trees are presented, the actual intricacies of their generation are omitted and only a brief description of the synthesis is given. However, the difficulties in modelling the various systems are covered in more detail.

13.2 Nitric Acid Cooling System

This example was first introduced by Lapp and Powers (1) and has since been widely covered in the field of fault tree synthesis. Two previous workers on this project, Kelly and Mullhi, have looked at this example in considerable detail. The system is shown in Figure 13.1.
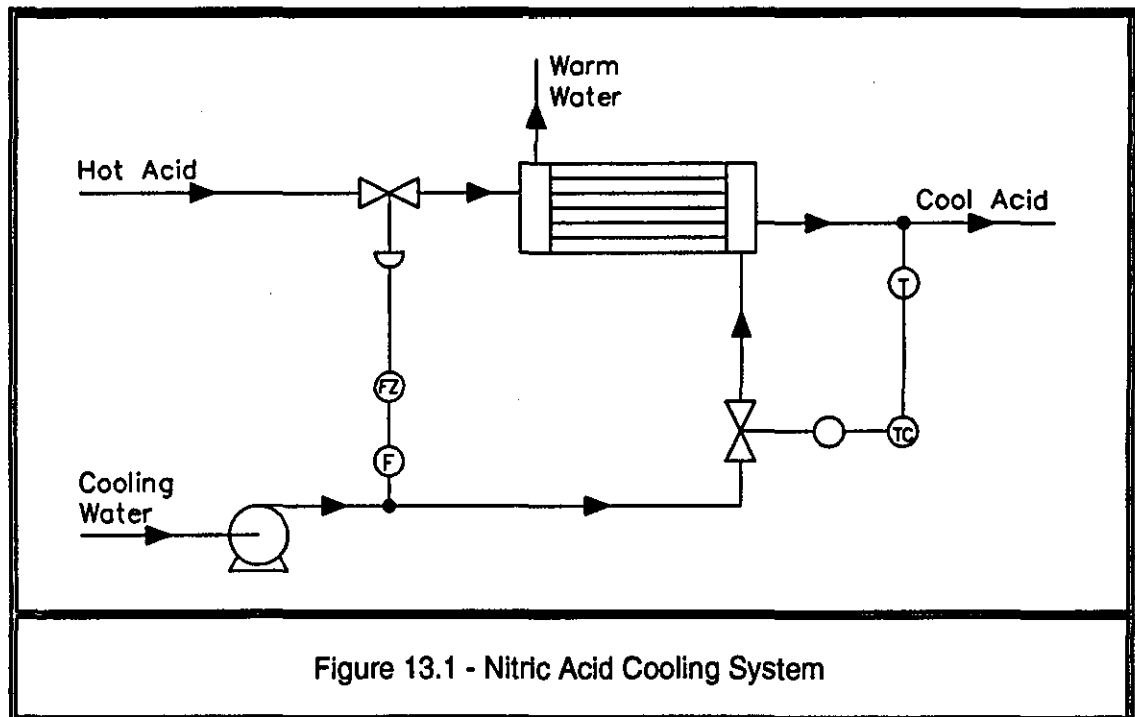
Figure 13.1 - Nitric Acid Cooling System

The hot nitric acid comes from the manufacturing process and must be cooled sufficiently before proceeding to other areas of the plant. A shell and tube heat exchanger with a cooling water supply is used for this purpose. A control loop connected to a temperature sensor on the acid outlet from the heat exchanger manipulates the flow of cooling water into the heat exchanger. Should the cooling water supply fail completely a trip system operates to shut off the flow of hot acid.

13.2.1 Configuration And Modelling

The configuration diagram is show in Figure 13.2.
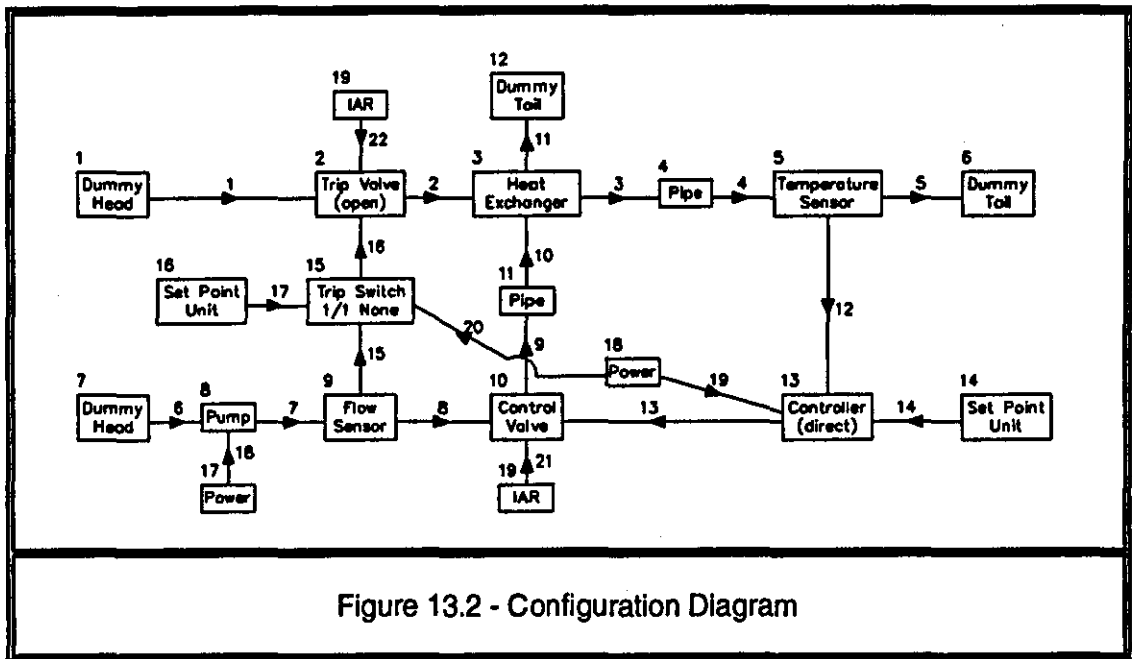
Figure 13.2 - Configuration Diagram

All models used in this configuration are present in the unit model library. The heat exchanger model, although present in the unit library, could easily be generated using the heat exchanger model generation routine and a detailed account of the model characteristics can be found in Chapter 8.

Common mode utility failures are modelled using single supply units. From Figure 13.2 it can be seen that the power supply (unit 18) and the instrument air supply (unit 19, shown as two units for clarity) both have connections to the control loop and the trip system. It should be noted that a different power supply is used for the pump.

The information required to define the control loop and the trip system has been covered in detail in Chapter 4, as has a more detailed description for determining the format of the configuration diagram. Therefore, only the definition of the control loop and trip system is given here.

CONTROL LOOP

SENSED VARIABLE: T
VARIABLE SENSED IN UNIT: 5
CONTROL VALVE UNIT NUMBER: 10
OTHER UNITS IN CONTROL LOOP: 13 14 18 19
VARIABLE T REGULATED IN CONNECTIONS: 3   4   5
FLOW MANIPULATED IN CONNECTIONS: 6   7   8   9 10 11
LOOP IS NOT OF FEEDFORWARD TYPE

OPEN VALVE TRIP SYSTEM

TRIP VALVE UNIT NUMBER: 2
OTHER UNITS IN TRIP SYSTEM: 9 15 16 18 19
TRIP IS OF THE FEEDFORWARD TYPE

13.2.2 Fault Tree Synthesis

A very detailed description of the process for synthesising a fault tree for high temperature in the acid outlet stream for this example is given by Mullhi (2). The greater part of the synthesis principles and rules has not been altered during this project so a step by step description will not be given.

The top event chosen for the example is high flow of coolant out of the pipe from the control valve to the heat exchanger. This was chosen because it uses the manipulated variable control loop template as opposed to the regulated variable template used by Mullhi.

The top event is HIGHFLOW Unit 11 which translates to Q 10 HI. The synthesis algorithm immediately recognises this to be in the manipulated stream of the control loop and applies the appropriate control loop template. This produces the partial fault tree shown in Figure 13.3.

Figure 13.3 - Partial Fault Tree Showing Control Loop Template
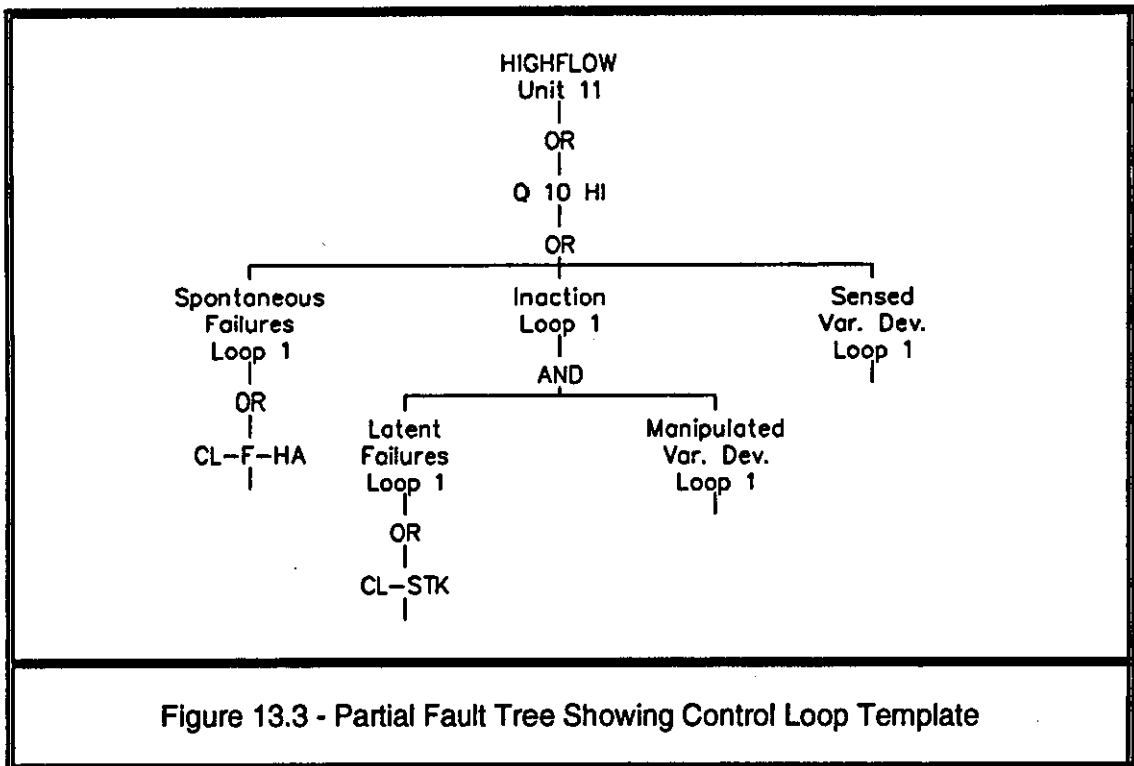
The fault tree branch for the spontaneous failure of the control loop and the branch for the latent failure of the control loop are shown in Figure 13.4.

a)   b)

Figure 13.4 - Control Loop Failure Branches
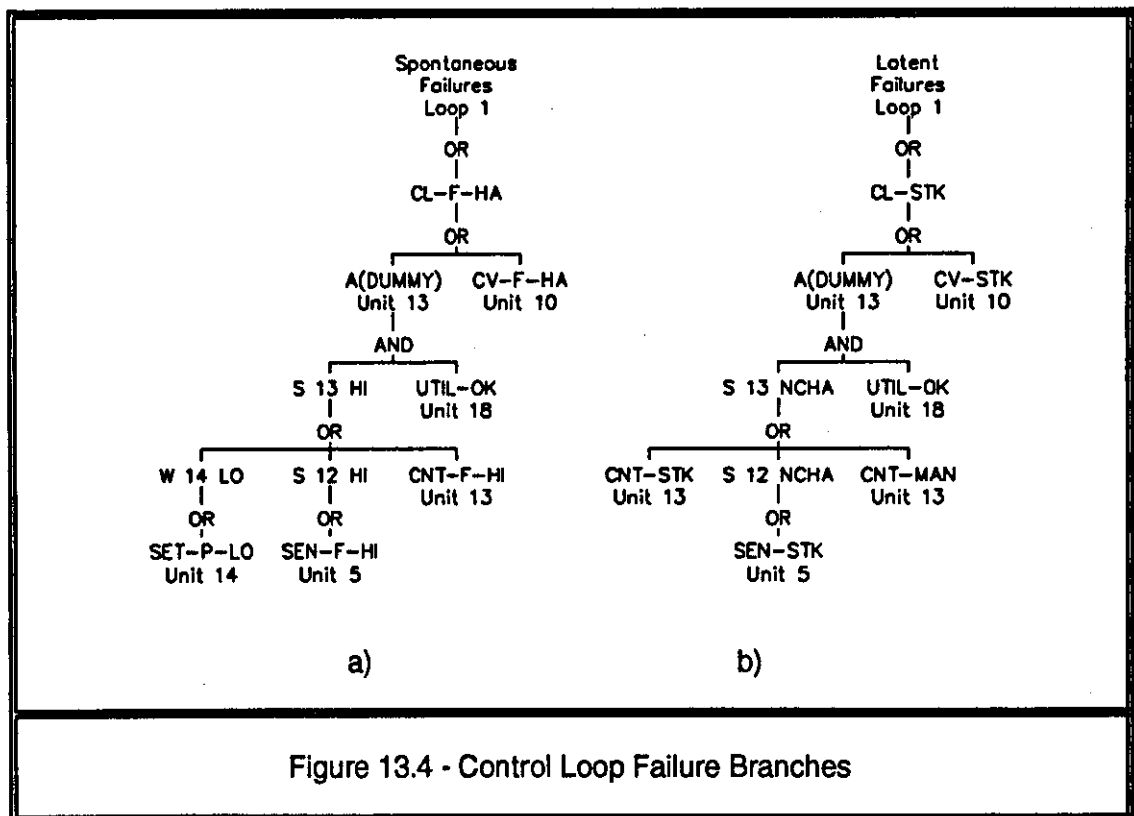
The spontaneous failure branch traces all the faults which may cause the control loop to fail high. It starts off with the control valve failing with a high aperture CV-F-HA, or a high signal to the valve S 13 HI. However, in order for a high signal to reach the valve the utility supply to the controller must be functioning. Therefore, S 13 HI is ANDed with UTIL-OK Unit 18. A high signal from the controller S 13 HI may be caused by the controller failing high CNT-F-HI, a high signal from the sensor S 12 HI, or a low signal from the set point unit W 14 LO. The latter may be caused by a low set point SET-P-LO. The high signal from the sensor may be caused by the sensor failing high SEN-F-HI. The spontaneous failure branch is shown in Figure 13.4a.

The synthesis for the latent failure branch of the control loop is very similar to the spontaneous failure branch and is shown in Figure 13.4b.

The fault tree branches for the manipulated variable deviation and the sensed variable deviation are shown in Figure 13.5.

Figure 13.5 - Manipulated And Sensed Variable Deviation Branches
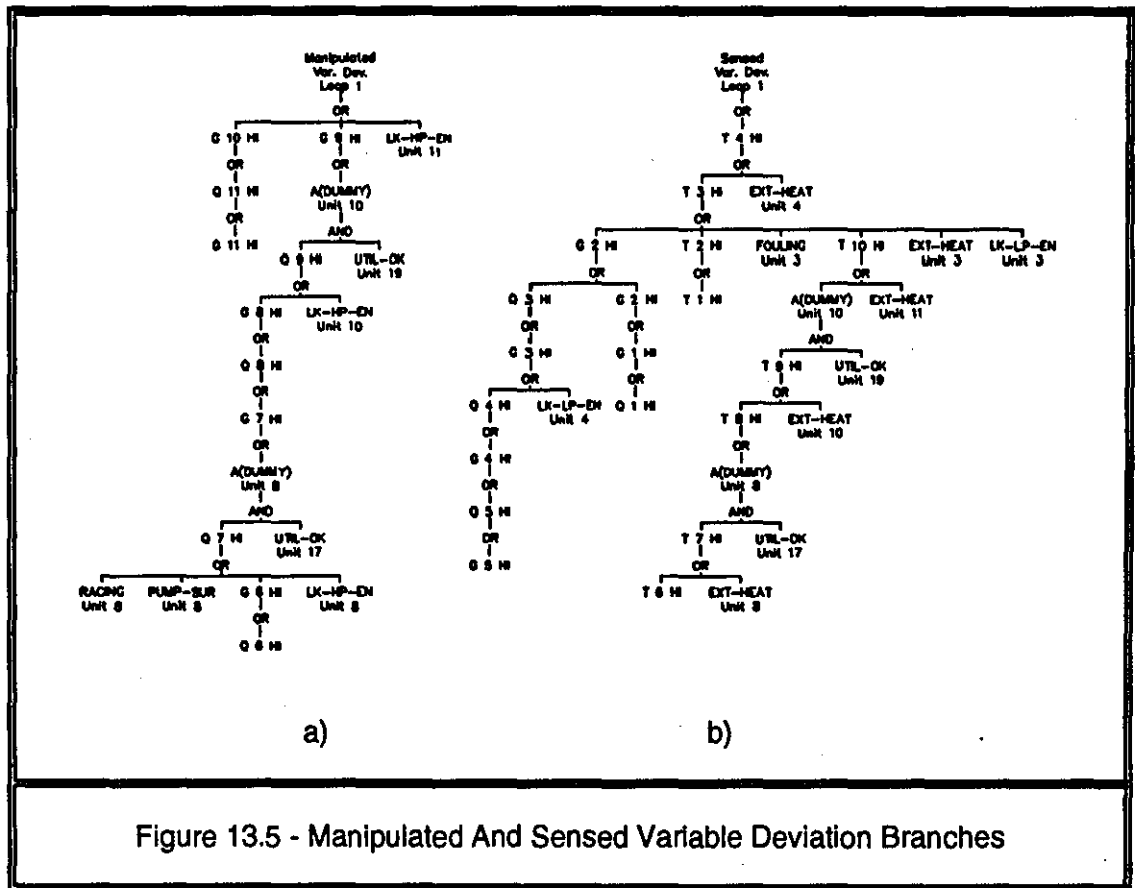
The continuation branch of the manipulated variable deviation is derived from the pipe model minitrees. The causes of Q 10 HI are G 10 HI or G 9 HI or a leak from a high pressure environment LK-HP-EN. The causes of G 10 HI are traced through the heat exchanger to the dummy tail unit resulting in the diamond event G 11 HI. The causes of

G 9 HI are traced back through the control valve. Since the valve requires an air supply to open Q 9 HI is ANDed with UTIL-OK Unit 19. The trace is continued through the flow sensor to the pump. A high flow out of the pump requires the pump to be working so the flow is ANDed with UTIL-OK Unit 17. Causes of high flow out of the pump are high flow into the pump G 6 HI, the pump racing RACING, a pump surge PUMP-SUR or a leak from a high pressure environment LK-HP-EN. The causes of high flow into the pump are traced to the dummy head unit resulting in the diamond event Q 6 HI. This branch of the fault tree is shown in Figure 13.5a.

The sensed variable deviation is a high temperature in the pipe on the heat exchanger outlet T 4 HI. This branch is included because a deviation in the sensed variable accompanied by the correct action of the control loop will result in a deviation of the manipulated variable. The causes of this are high temperature out of the heat exchanger T 3 HI, or an external heat source EXT-HEAT. There are several causes for high temperature in the heat exchanger outlet. The basic faults are fouling, an external heat source EXT-HEAT, or a leak to a low pressure environment LK-LP-EN in the cooling side. Another cause is a high acid temperature into the heat exchanger T 2 HI, which is traced to the dummy head giving the diamond event T 1 HI, and a high coolant temperature T 10 HI, which is traced to the dummy head unit giving the diamond event T 6 HI. The final cause of high temperature from the heat exchanger is high flow of acid G 2 HI. This is traced downstream to the diamond event G 5 HI and upstream to the diamond event Q 1 HI. The sensed variable deviation branch is shown in Figure 13.5b.

This completes the synthesis of the fault tree for the high flow top event. The trip system does not appear in the fault tree because neither of the failure modes of the trip result in high flow.

13.2.3 Conclusions

The Lapp and Powers nitric acid cooling system has been used as a basic illustration of the decomposition, modelling and fault tree synthesis ability of the Faultfinder package. Together with the top event used by Mullhi (2) they give an example of a regulated variable deviation and a manipulated variable deviation.

This example has shown the structural features imposed by the control loop template on the fault tree giving it a formal framework which is easy to interpret.

13.3 Reaction Vessel Charging System

The reaction vessel charging system is a hypothetical section of plant developed for use in alarm diagnosis work. The format used here is a simplification of the system used by Trenchard (3). He used the Faultfinder package to generate fault trees for use in an alarm diagnosis package. The system is shown in Figure 13.6 where the extra line appearing on the process flow path is a representation of the electrical tracing used.

Figure 13.6 - Reaction Vessel Charging System
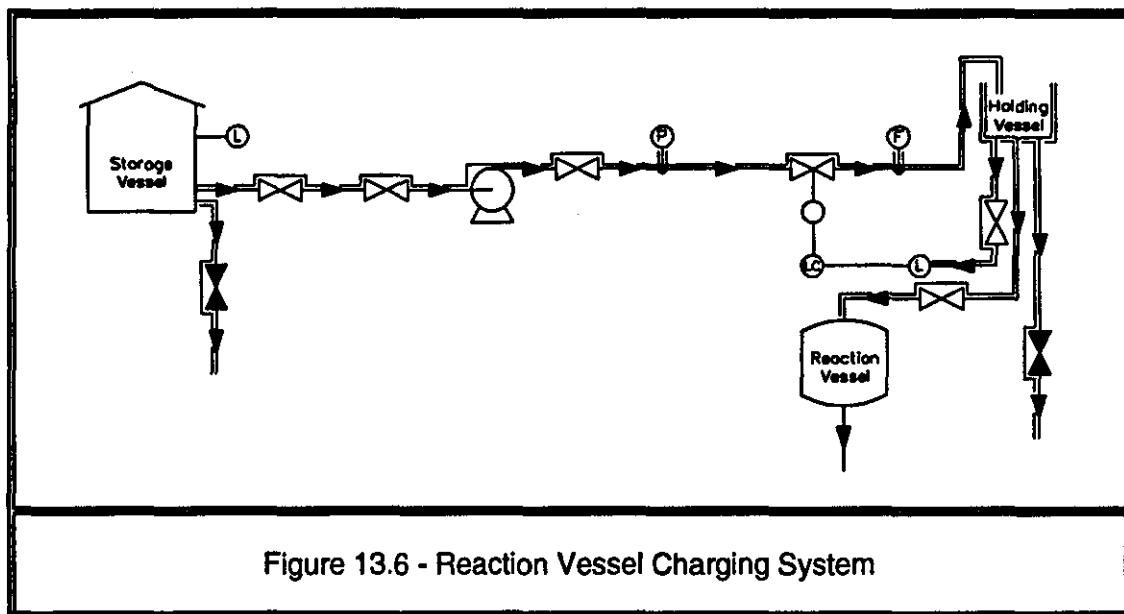
The function of the section of plant is to provide the reaction vessel with a constant flow rate of ethanoic acid. The acid is pumped from the storage facility to an elevated holding vessel. The level in the vessel is maintained by a controller and control valve on the inlet. The acid flows under gravity to the reaction vessel and hence the flow rate is controlled by the level in the holding vessel.

The process may seem straightforward but the choice of ethanoic acid with its freezing point of 16.6°C poses problems in that the piping and instruments require an electrical tracing system to prevent blockages due to freezing. This is shown as an additional line along the process flow lines in Figure 13.6.

13.3.1 Configuration And Modelling

The configuration diagram is shown in Figure 13.7.

The difficulty with this configuration is how to model the pipe tracer malfunctions. A new variable E is introduced to model the flow of electrical current from the tracer unit to the pipeline.
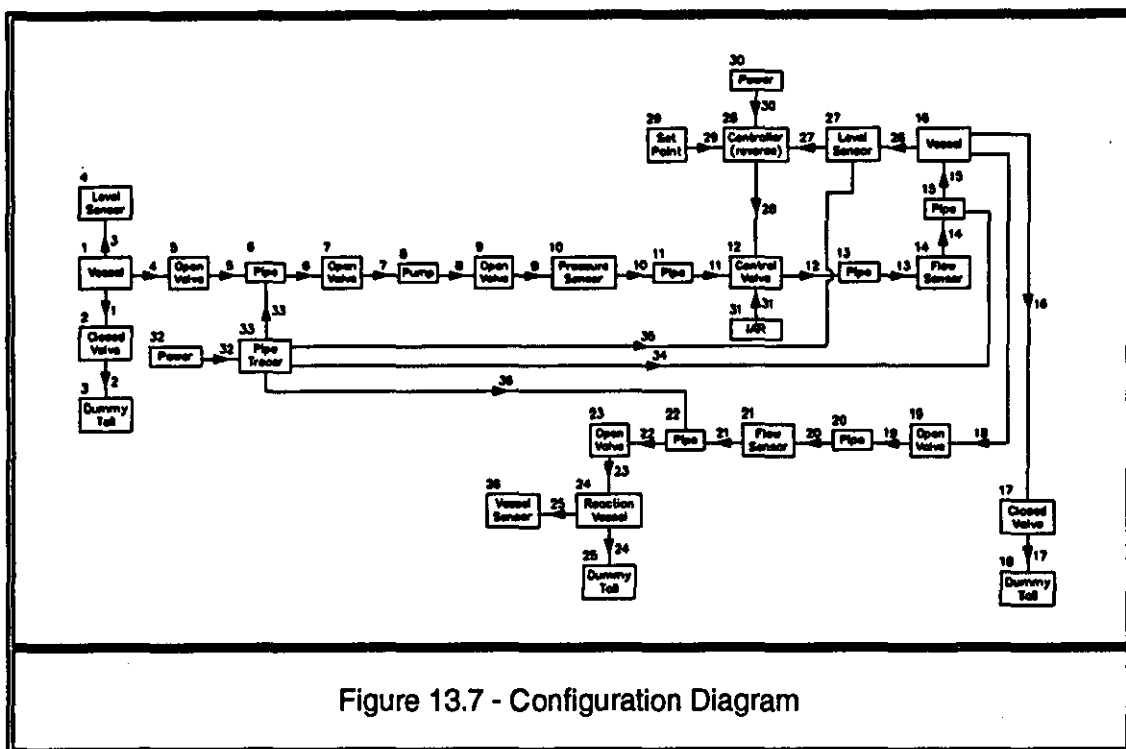
Figure 13.7 - Configuration Diagram

The effects of the process fluid freezing are partial blockage or, more severely, complete blockage. Some of the models in the pipeline contain these events as basic faults so it is possible to model the causes as secondary failures.

Secondary failures were introduced to allow variable deviations or basic faults in unit models to be further developed for a particular situation without the need to change the unit model. For example, a leak may be caused by an impurity or low flow may be caused by a low temperature. Two types of secondary failure are used for this: a Type I failure represents an additional cause of a variable deviation and a Type II failure represents an extension to a basic fault. Thus, an example of a Type I failure is a low flow resulting from FREEZING caused by low temperature and an example of a Type II failure is a leak to a low pressure environment resulting from CORROSION caused by an impurity.

Two secondary failures are used to represent differing degrees of severity. They are both defined as Type II failures because they model the causes of basic faults. The first, called FREEZE1, identifies some freezing as an additional cause for partial blockage PART-BLK. This is caused by low electrical current. The second failure is called FREEZE2 and represents the fluid being completely frozen. This is caused by a complete loss of electrical current. The following event statements are used:

V E3IN LO : FREEZE1
V E3IN NONE : FREEZE2

Rather than exhaustively modelling the effects of freezing in every unit the secondary failures are only defined as affecting some of the models. These are the pipe models used at the start of the tracing, at the entrance to the holding vessel and in the line to the reaction vessel and the level sensor unit. This avoids excessive occurrences of the faults. The secondary failure definitions for the configuration are:

```
NAME: FREEZE1
SUSCEPTIBLE FAULTS: PART-BLK
UNITS:  6  15  22  27


NAME: FREEZE2
SUSCEPTIBLE FAULTS: COMP-BLK
UNITS:  6  15  22  27
```

The overall effect of this is that a blockage may result from freezing caused by insufficient current to the electrical tracer.

Although the pipe model in the unit library only has two ports, an inlet and an outlet, it can be seen in the block diagram shown in Figure 13.7 that the units connected to the pipe tracer have three connection points. This does not cause any problems because the secondary failure models provide the link between the blockage faults in the pipe model and the connection to the pipe tracer.

Should it be necessary to model the effects on temperature, a variable deviation, in the pipe of the tracer unit, a Type I secondary failure can be used to relate pipe temperature to flow of electric current.

The level sensor is highlighted as a special model since it is susceptible to blockages, even more so than the pipe since flow does not occur in this unit. Therefore, the model contains the two blockage faults. From Figure 13.6 it can be seen that the level sensor consists of a hand valve and a pneumatic level sensing device. In the Faultfinder methodology it is not possible to connect a hand valve to a vessel port so the level sensor and the hand valve are modelled as one unit. The resultant model contains three additional causes of no change in output signal to the simple level sensor model. The new level sensor model contains the following propagation equation and event statements:

```
S2SIG= F( L1VES)


F  SEN-F-HI : S2SIG HI
F  SEN-F-LO : S2SIG LO
F  SEN-STK : S2SIG NCHA
```

F PART-BLK : S2SIG NCHA
F COMP-BLK : S2SIG NCHA
O HV-D-SH : S2SIG NCHA
V S4UTL NONE : S2SIG NONE

The pipe tracer unit is used to introduce specific faults for this unit and to convert a utility supply to an electrical current. Event statements are used to convert a low utility supply to a low electrical current and the causes of no electrical current are the pipe tracer heater failing or no utility supply. The event statements are:

V S1UTL LO : E2OUT LO, E3OUT LO, E4OUT LO, E5OUT LO
V S1UTL NONE : A(DUMMY)
F HEATFAIL : A(DUMMY)
I A(DUMMY) : E2OUT NONE, E3OUT NONE, E4OUT NONE, E5OUT NONE

All the vessels in the configuration have been developed using the vessel model templates or the generation routine. The storage facility is modelled as a large capacity unit with just two outlet ports and no inlet ports. This model already exists in the unit library but can be obtained from the open vessel model template by deleting all ports except for one outlet flow port to a pump and one outlet non-flow port to gravity. Low level or no level may result from some flow through the closed valve or a leak in the unit. The holding vessel is straightforward with no unusual features. It has a pumped inlet and a gravity outlet which are affected by the level in the vessel. The second outlet port is a non-flow port. Any reaction effects in the reaction vessel are ignored since these are assumed to affect only the temperatures and outlet flows but not the flow into it. This unit is modelled as a simple vessel.

The definition of the control loop for the level is straightforward but it should be noted that the tracer units are also included, as failure of these results in failure of the control system. The definition is

CONTROL LOOP

SENSED VARIABLE: L
VARIABLE SENSED IN UNIT: 27
CONTROL VALVE UNIT NUMBER: 12
OTHER UNITS IN CONTROL LOOP: 28 29 30 31 32 33
VARIABLE L REGULATED IN CONNECTIONS: 26
FLOW MANIPULATED IN CONNECTIONS: 4 5 6 7 8 9 10 11 12 13 14 15
LOOP IS NOT OF FEEDFORWARD TYPE

13.3.2 Fault Tree Synthesis

The top event chosen to illustrate this example is low flow of ethanoic acid to the reaction vessel. The unit is number 20, the pipe model. Figure 13.8 shows the causes of low flow both upstream and downstream of the pipe and the point where the control loop template for the regulated variable is placed.
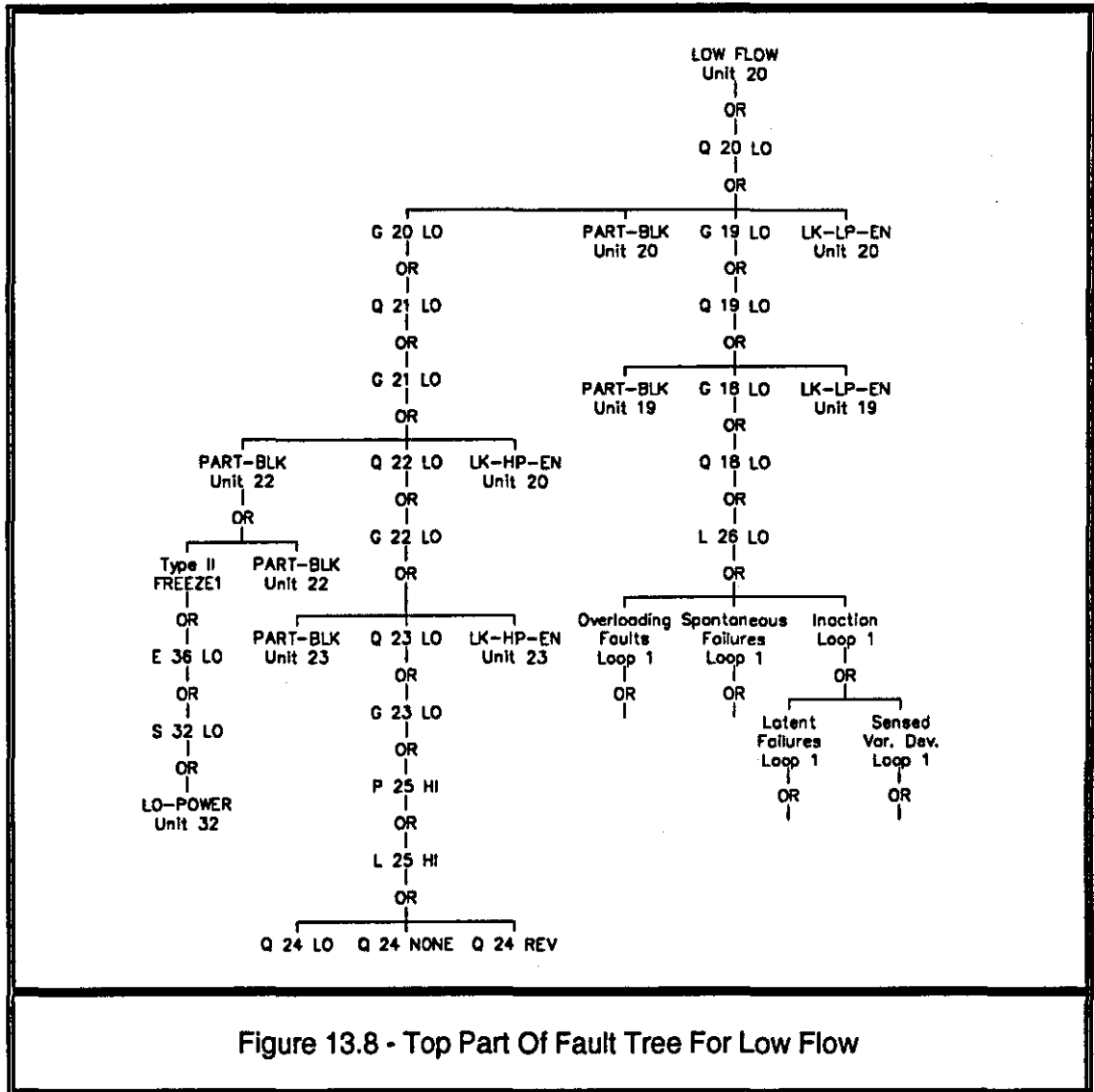
Figure 13.8 - Top Part Of Fault Tree For Low Flow

The G 20 LO branch traces faults downstream towards the reaction vessel and shows the addition of the FREEZE1 secondary failure event at the fault PART-BLK Unit 22. This fault is traced to low electric current from the tracer E 36 LO caused by a low utility supply LO-POWER Unit 32.

The G 19 LO branch traces faults upstream to the holding vessel where a deviation in the level causes the control loop template to be used.

Figure 13.9 shows the control loop overload branch.
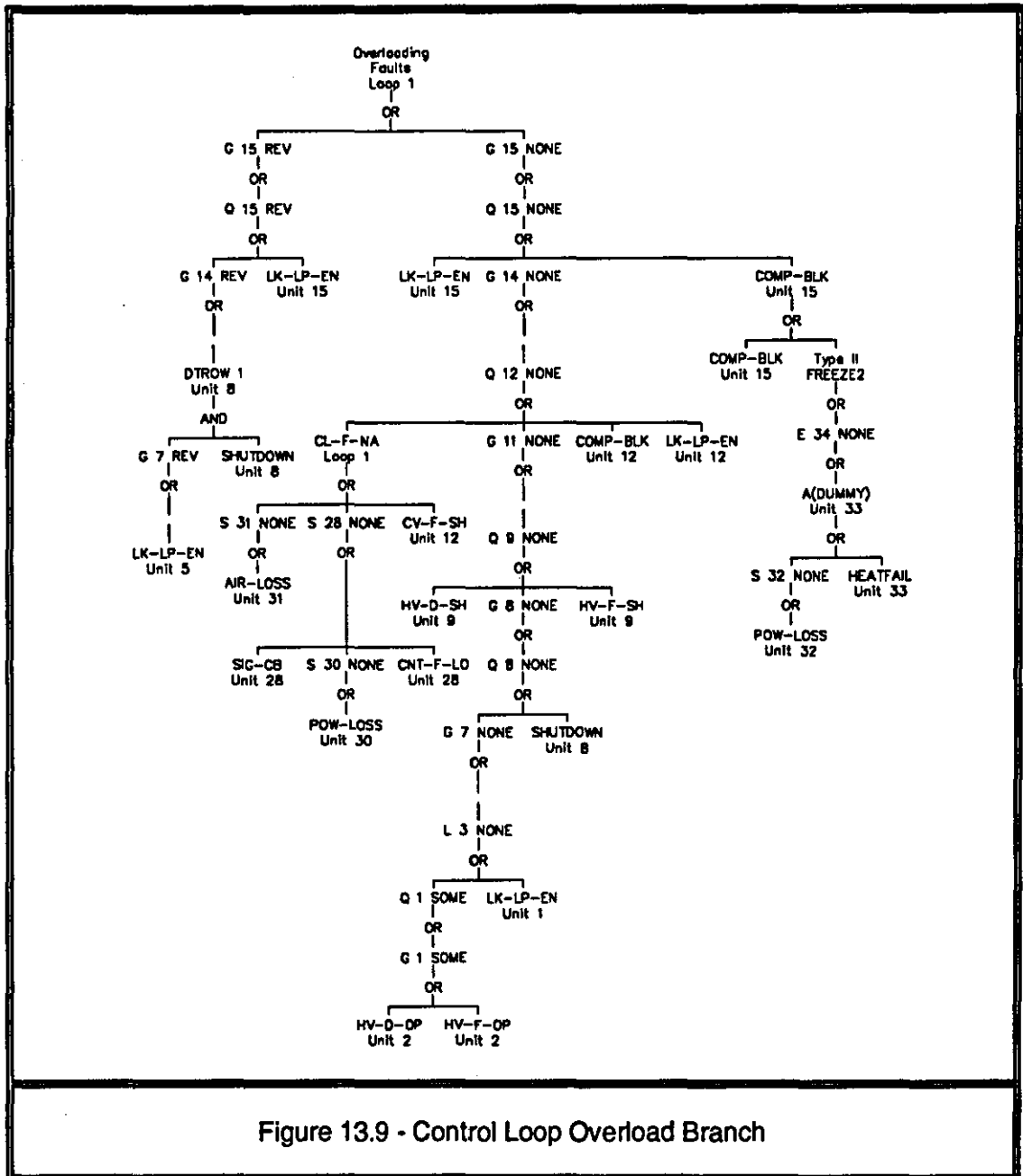
Figure 13.9 - Control Loop Overload Branch

The causes of control loop overload are no flow or reverse flow in the manipulated stream. The reverse flow branch is quite straightforward and not all intermediate events and variable deviations are shown (this is indicated by the extended broken line after an OR gate). The non-flow branch has a FREEZE2 secondary failure at the complete blockage COMP-BLK fault on unit 15 and the control loop failing with no aperture CL-F-NA at the control valve. This branch is traced to the hand valve being open on the other storage vessel outlet connection.

13-12

The spontaneous failure and latent failure branches of the control loop are shown in Figure 13.10.
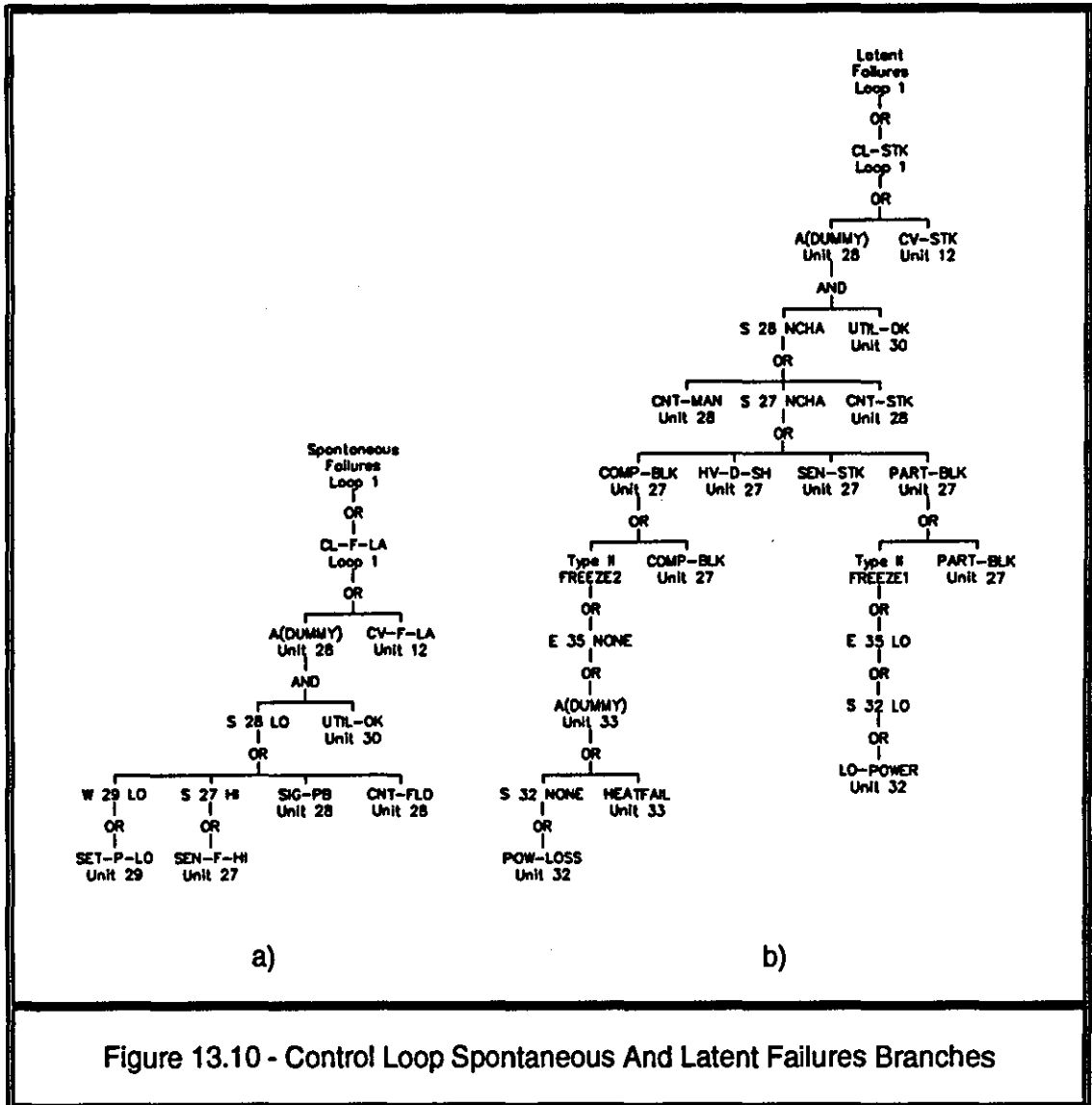
Figure 13.10 - Control Loop Spontaneous And Latent Failures Branches

The spontaneous failures branch of the control loop is shown in Figure 13.10a and poses no problems.

The control loop latent failures branch shown in Figure 13.10b is slightly unusual due to the level sensor model used. The branch has the hand valve closed fault HV-D-SH Unit 27 to represent closure of the valve between the connection to the vessel and the sensor. Also the sensor is susceptible to both the secondary failure effects.

The sensed variable deviation branch shown in Figure 13.11 is straightforward and again has the secondary failure branch on the partial blockage fault PART-BLK Unit 15.
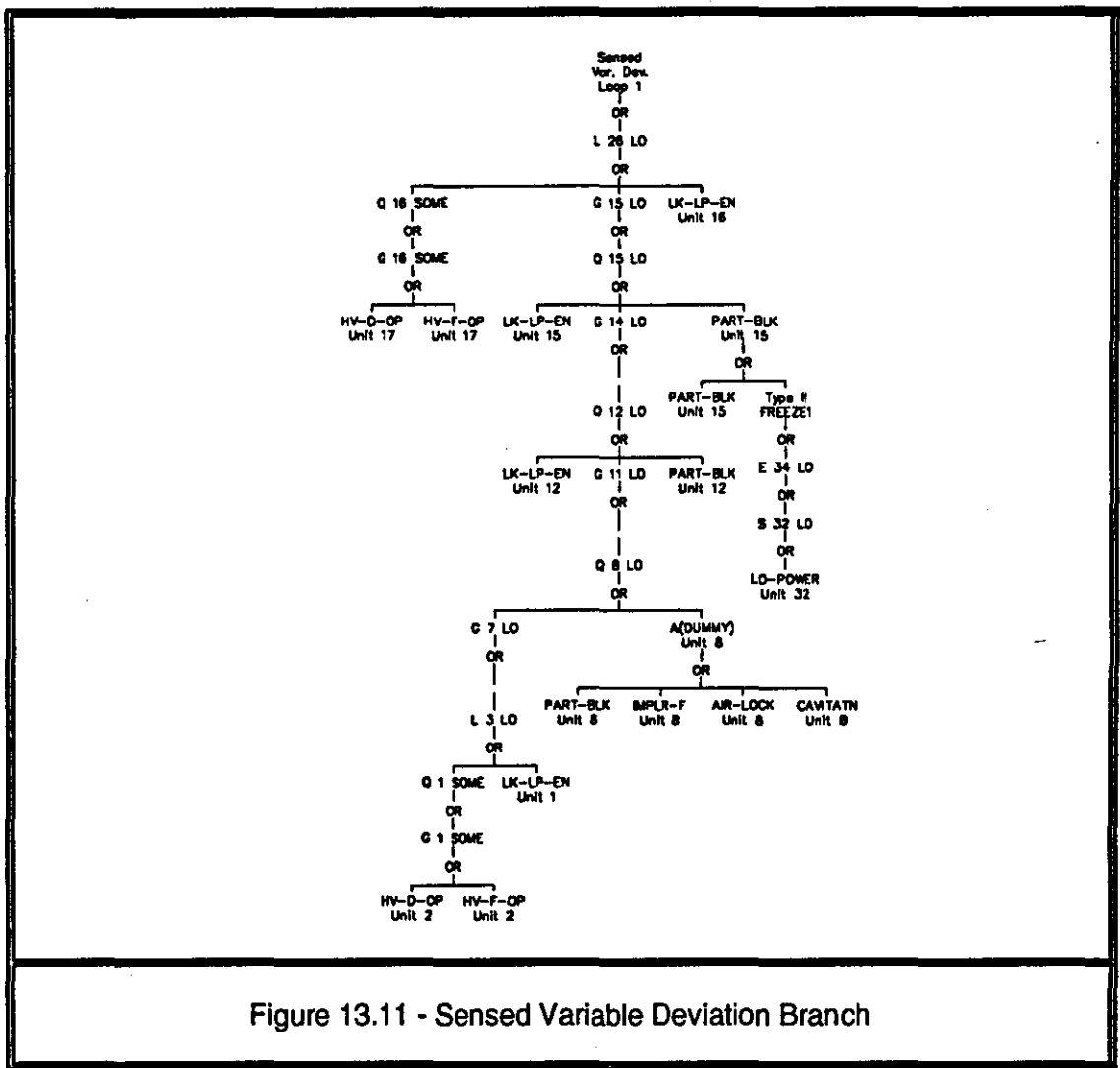
Figure 13.11 - Sensed Variable Deviation Branch

This completes the fault tree for low flow into the reaction vessel.

13.3.3 Conclusions

The reaction vessel charging system has illustrated the ability of the Faultfinder package to deal with unusual plant systems such as the distribution of an electrical heating system throughout the whole section. The modelling has resulted in the introduction of a new variable for the electrical current.

The system has also demonstrated the use of the secondary failures facility which was created during the first version of the package but which, before the present work, had hardly been used. It has highlighted the need for the type of secondary failure name to be revised because the original interpretation no longer applies. This only reflects the changes in the application of the failures but not the methodology used. This is covered in more detail in Section 5.8.

13.4 Reaction Vessel Cooling System

The ethylene oxide reaction vessel cooling system was used by Piccinini and Levy (4, 5) for an operability analysis. It has been used by Khan (6) as part of a study for computer-aided design. The system diagram is shown in Figure 13.12.
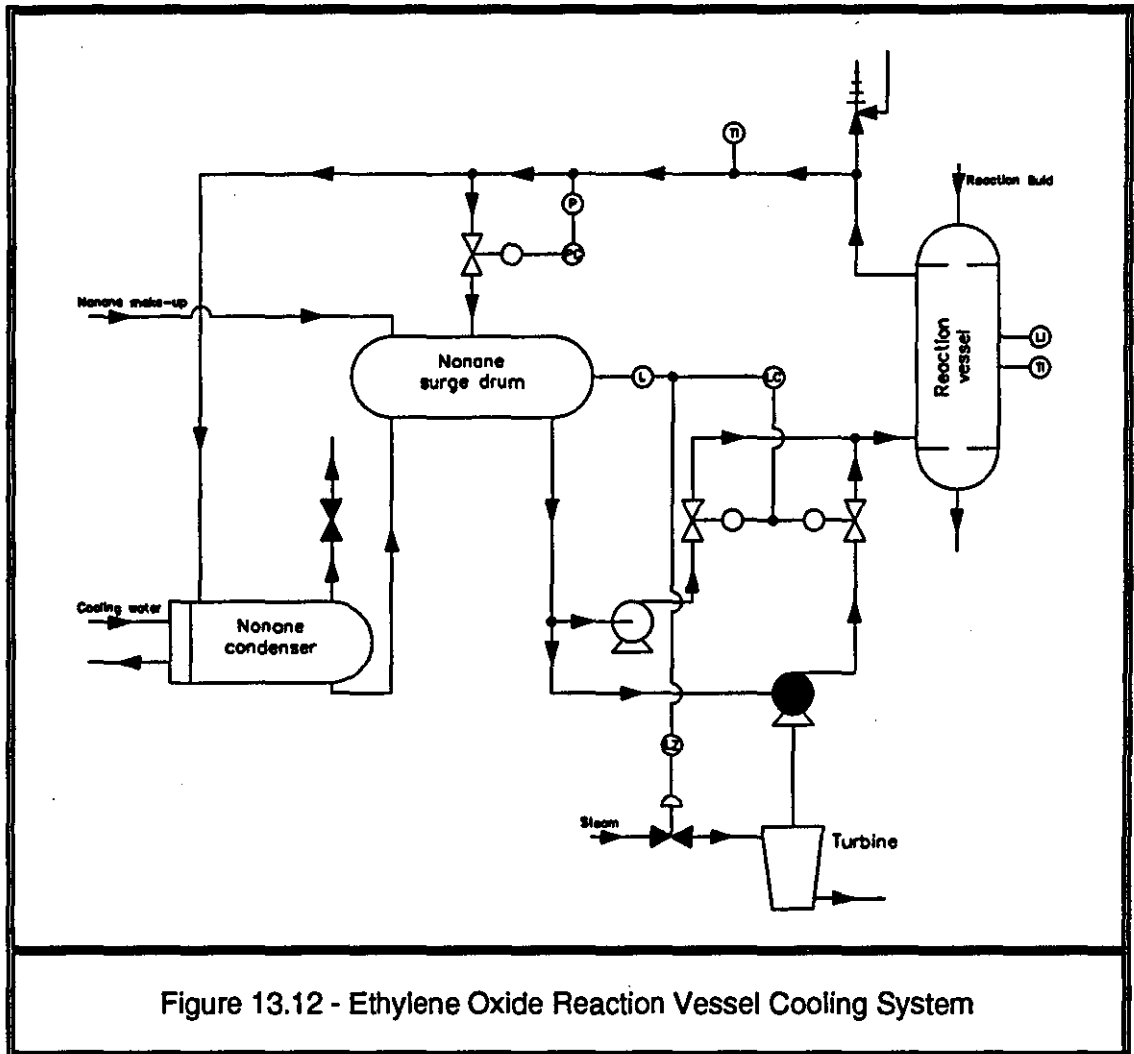
Figure 13.12 - Ethylene Oxide Reaction Vessel Cooling System

A detailed account of the ethylene and oxygen reaction process and the nonane cooling system is given by the authors (5). The basic layout of the plant is as follows. The vertical reaction vessel and surge drum are six metres above the ground-mounted condenser and pumps. The coolant loop is closed hence the level controller on the surge drum maintains a constant level in the reaction vessel.

The temperature of the coolant is kept constant by controlling the evaporation pressure. Under normal operating conditions the pressure in the surge drum plus the hydrostatic head of the liquid in the drum exerts sufficient pressure on the liquid outlet from the condenser to submerge some of the cooling tubes. This exposes enough tubes to

condense the required amount of vapour and allows for undercooling. If the amount of vapour in the reaction vessel outlet stream increases as a result of higher temperature in the reaction vessel, the pressure control valve aperture decreases. This reduces the pressure in the surge drum, causing more liquid to flow from the condenser and consequently uncovering more condensing tubes allowing the extra vapour to be condensed.

Should the pump break down or should there be a sudden evaporation in the reaction vessel, the surge drum level will increase to a point where the high level trip will operate. This opens the closed trip valve on the steam line to the turbine generator for the second pump (filled in black in Figure 13.12 to represent a non-operating status).

13.4.1 Configuration And Modelling

The configuration diagram is shown in Figure 13.13.
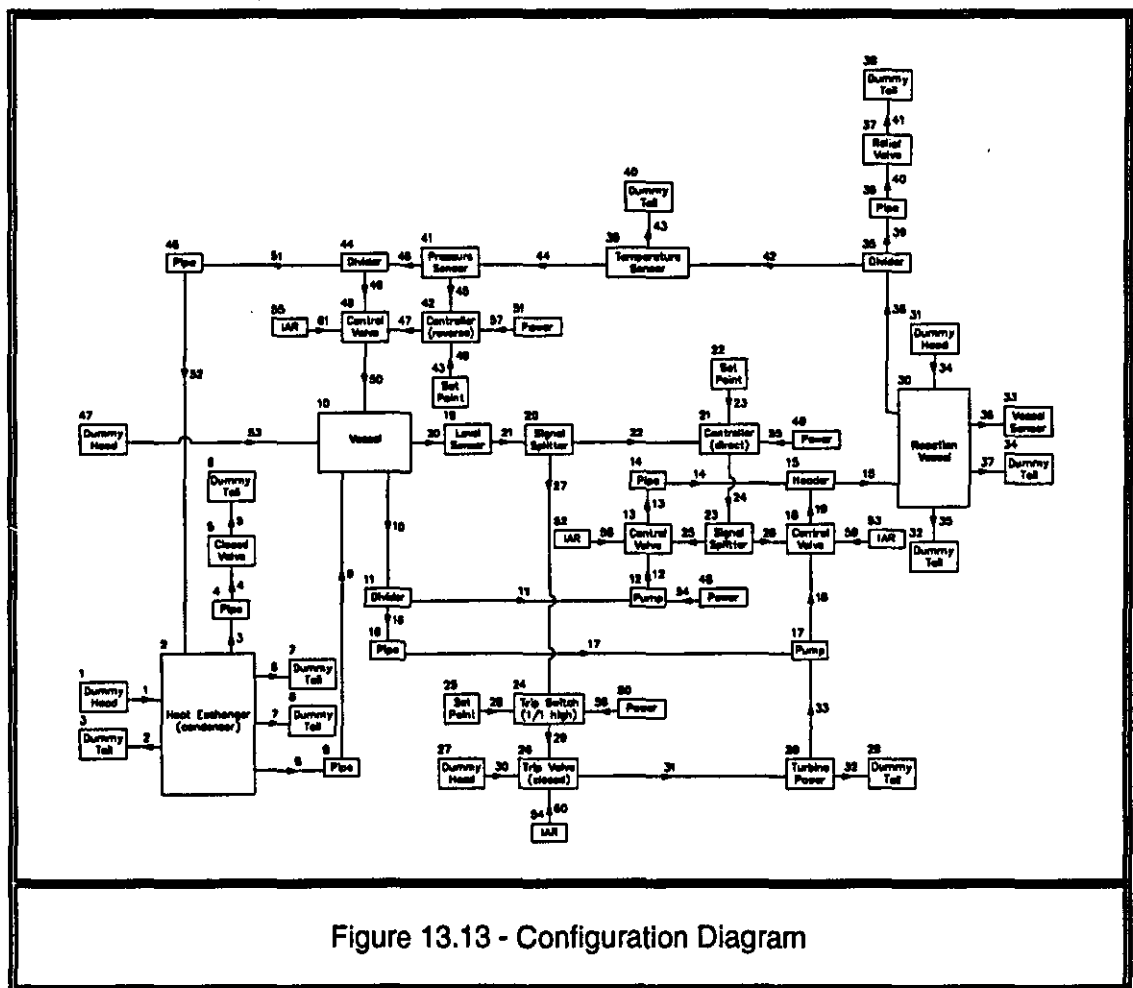
Figure 13.13 - Configuration Diagram

All of the unit models are contained in the model library except for the vessels and the condenser.

The reaction vessel is modelled using the heat exchanger generation routine because the modelling is only concerned with the cooling system and, therefore, it is not necessary to model the reaction parameters. The causes of high evaporation rate are high reactant inlet temperature or high reactant flow rate in the tubes. The effects of reactant compositions are not considered. The coolant inlet is a pumped liquid but the outlet is a vapour under back-pressure from the vessel's shell side and hence is affected by the pressure.

The condenser is modelled as a shell and tube heat exchanger with a relief port on the shell side. The tubes contain the water for condensing the coolant vapour. Both inlet and outlet to the shell side are affected by the vessel pressure.

The surge drum is a product of the vessel model generation routine. The vapour and liquid inlet ports are affected by the vessel pressure but since the liquid outlet is pumped the vessel pressure is assumed not to affect it. The liquid make-up port is only used occasionally to replace lost fluid, therefore it is modelled as a non-flow port connected to a dummy head unit.

The configuration has a single divider-header combination around units 11 and 15. This gives a continuity of flow between streams 10 and 15 which means that low flow out of the header, unit 15, can only have the low flow deviation as a cause entering at the divider, unit 11. Other possible loops can be found but each of these contains a unit in which accumulation of mass can occur thus negating the restriction on the inlet and outlet deviations.

The level control system is straightforward but there is only one control loop because the second control valve, unit 18, although operating, does not manipulate the flow because it is a non-flow leg. The flow is manipulated in streams 11 to 14 only.

CONTROL LOOP

NUMBER: 1
SENSED VARIABLE: L
VARIABLE SENSED IN UNIT: 19
CONTROL VALVE UNIT NUMBER: 13
OTHER UNITS IN CONTROL LOOP: 20 21 22 23 49 52
VARIABLE L REGULATED IN CONNECTIONS: 20
FLOW MANIPULATED IN CONNECTIONS: 11 12 13 14
LOOP IS NOT OF FEEDFORWARD TYPE

The pressure control system used in this example is unusual in the way that it is used to increase or decrease the condensation rate in the condenser. This is achieved by reducing or raising, respectively, the pressure in the surge drum.

The following description looks at the effect of an increase in surge drum vessel pressure. To counteract this the hydrostatic head between the drum and the condenser must change to equalise the pressure between the two units. An increase in vessel pressure causes the flow of liquid from the condenser to the surge drum to decrease, reducing the level in the surge drum. This has two effects, the first is that the flow rate from the drum to the reaction vessel decreases due to the action of the level controller and the second is that the level in the condenser increases. A higher level in the condenser covers more cooling tubes, thus, decreasing the condensation rate. An opposite effect is achieved for a decrease in the surge drum pressure.

The pressure control loop behaves as follows. The sensor and the control valve are in different streams separated by the divider unit and the loop does not receive information about the effects of its actions. Therefore it is a feedforward loop. The controller manipulates the flow into the surge drum thereby regulating the pressure in the drum. This causes a problem due to the two types of pressure variable used. In pipes there is the pipe pressure and in vessels there is the vessel pressure. They are kept separate because they do not have the same deviations. However, the current programming set-up used means that pressure in a vessel cannot be specified as a controlled variable because its variable index number is greater than 9. This restriction can be overcome in this example because it is possible to specify the pressure in the inlet stream as the regulated variable because of the direct link between it and the vessel pressure. The problem cannot be solved simply because the indices form part of the programming methodology.

As defined in the description of the process the temperature in the reaction vessel is also regulated by this loop. The control loop definition is

CONTROL LOOP

NUMBER: 2
SENSED VARIABLE: P
VARIABLE SENSED IN UNIT: 41
CONTROL VALVE UNIT NUMBER: 45
OTHER UNITS IN CONTROL LOOP: 42 43 51 55
VARIABLE P REGULATED IN CONNECTIONS: 50
VARIABLE T REGULATED IN CONNECTIONS: 36
FLOW MANIPULATED IN CONNECTIONS: 49 50
LOOP IS OF THE FEEDFORWARD TYPE

The high level trip operation is simple but the definition must be considered carefully. The trip opens the valve to allow steam to flow to the turbine. This powers the pump which then draws liquid through the non-flow leg of the divider-header combination. This results in a change in the configuration and models which is a difficult situation for automatic synthesis to handle. However, we are only concerned with the failure of the trip system to activate and not the consequences of successful operation. The consequences are flow through the divider-header combination which cannot be a cause of the chosen top event of low level in the reaction vessel.

Unlike other trip systems in this thesis, where only the trip switch, sensor and utilities form part of the trip system, the turbine and pump must also be given. This is necessary because should either of these fail to operate then the trip action is defeated. Also it is not only the steam line which has flow when the trip activates but the non-flow leg of the divider-header combination as well. The trip definition is

CLOSED VALVE TRIP SYSTEM

TRIP VALVE UNIT NUMBER: 26
OTHER UNITS IN TRIP SYSTEM ARE: 17 19 20 24 25 28 50 54
CONNECTIONS HAVING FLOW WHEN VALVE OPENS: 16 17 18 19 30 31 32

13.4.2 Fault Tree Synthesis

The top event chosen is low coolant level in the reaction vessel UNDRLEVL Unit 30. The top part of the fault tree is shown in Figure 13.14.

The causes of low level are low G 15 LO or no G 15 NONE flow of coolant, a high evaporation rate Q 37 HI, or a leak from the vessel LK-LP-EN. The high evaporation rate may be caused by high flow of reactants through the vessel Q 34 HI or G 35 HI, or high temperature of the reactant stream T 34 HI. All of these events are diamond events.

The cause of no coolant flow is traced to the header unit where there must be no flow in both the inlet streams G 14 NONE and G 19 NONE. Stream 19 has been identified in the configuration as the non-flow leg of the divider-header combination and therefore *no flow* will be the normal event. However, the stream has been identified as having flow when the trip system operates. Assuming the trip operates normally then for the leg to continue to have no flow the control valve must fail closed. Therefore, the cause of G 19 NONE is CL-F-NA. The causes of this are developed as for any other control loop except that at the sensor the synthesis also traces causes of no level in the holding vessel L 20 NONE. This is traced to no flow into the vessel G 9 NONE, caused by no pressure in the

condenser P 6 NONE. This may be caused by no condensation flow Q 7 NONE, which is a result of fouling or loss of cooling water Q 1 NONE or G 2 NONE, or no flow into the condenser G 52 NONE, which is developed elsewhere.
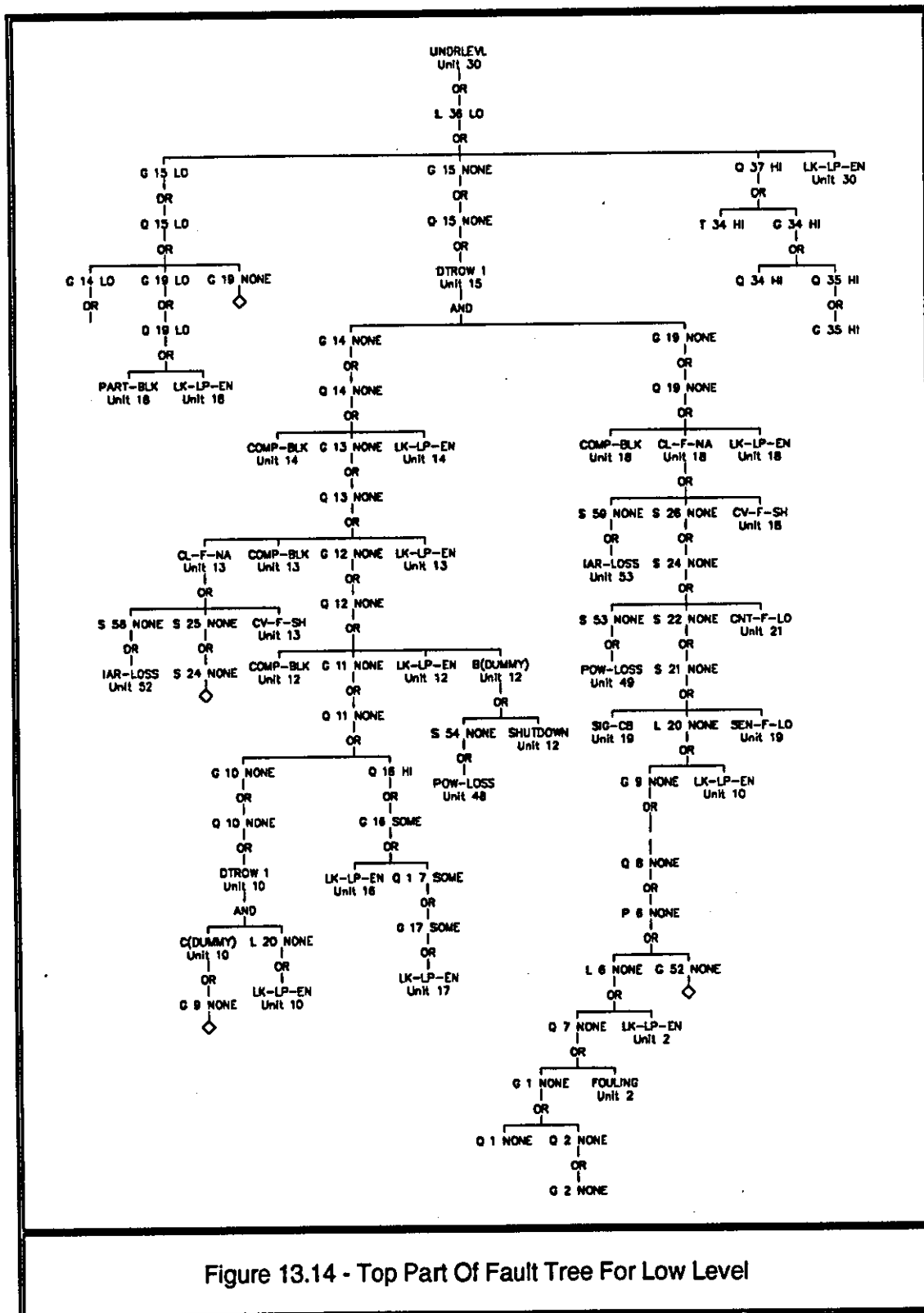
Figure 13.14 - Top Part Of Fault Tree For Low Level

No flow in stream 14 may also be caused by the control loop failing with no aperture CL-F-NA Unit 13. The development is almost identical to that for unit 18 except for the different control valves. The diamond at the end of the S 24 NONE branch indicates a continuation elsewhere in the fault tree. No flow through the control valve Q 13 NONE may also be caused by no flow into it G 12 NONE. The causes of this are pump failure or power loss, or no flow at the start of the divider outlet leg Q 11 NONE. No flow may be caused by high flow in the other divider outlet leg Q 16 HI or no flow into the divider G 10 NONE. The high flow is converted to some flow G 16 SOME because it is a non-flow leg and the causes of this are traced to leaks in the pipe or pump. No flow into the divider is caused by no flow out of the holding vessel Q 10 NONE. A decision table provides the causes of this as no level L 20 NONE ANDed with no flow into the vessel G 9 NONE. This event has been developed for a previous branch.

Low flow of coolant has causes of low flow through the flow leg of the header unit G 14 LO or low G 19 LO or no G 19 NONE flow in the non-flow leg of the header. The no flow branch has been developed previously and low flow in the non-flow leg has causes of a partial blockage or a leak. The development of the low flow fault for the flow leg of the header is shown in Figure 13.15.
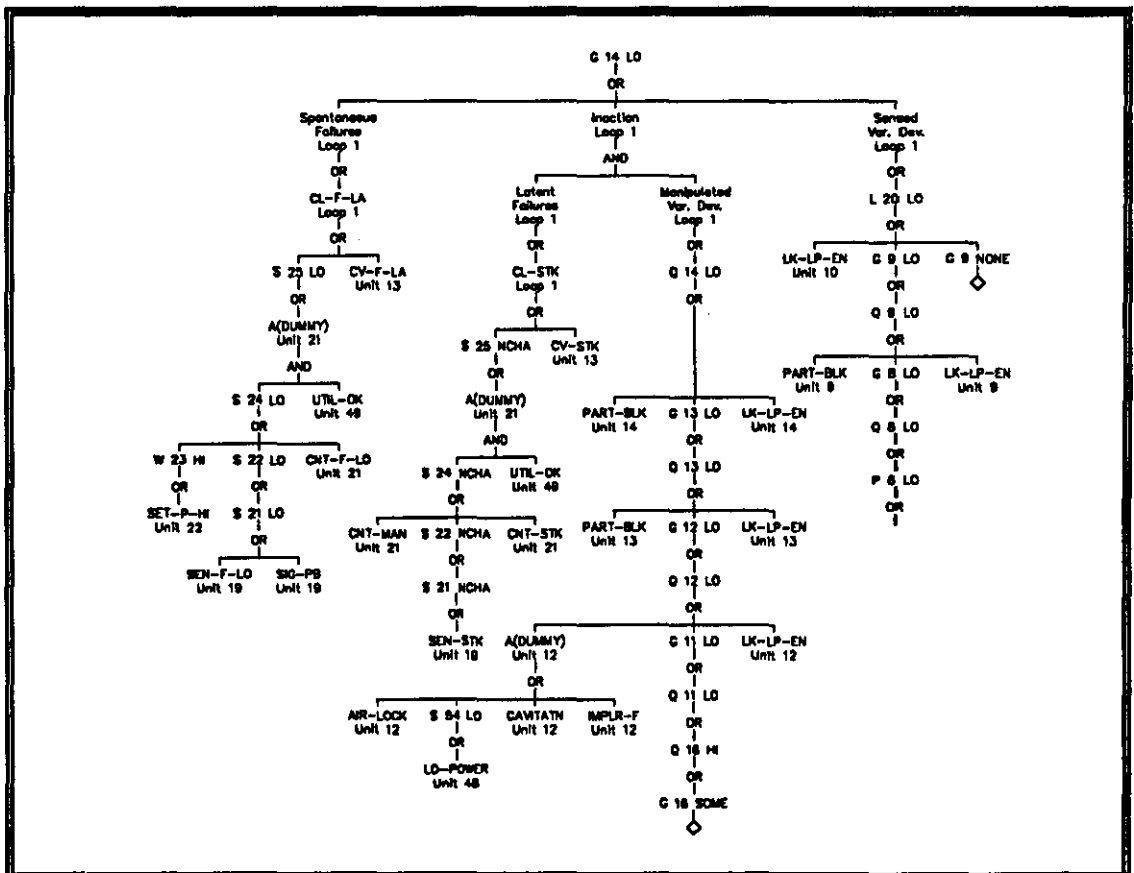
Figure 13.15 - The Causes Of Low Flow

Stream 14 is manipulated by control loop 1 so the appropriate control loop template is applied. The spontaneous failures and latent failures branches follow the same format as those in the earlier examples and are not explained any further.

The manipulated variable deviation branch traces low flow through the rest of the divider-header combination. Low flow through the pump Q 12 LO may be caused by pump failures such as cavitation or impeller failure or low flow into the pump G 11 LO. Low flow into the divider-header combination G 10 LO is not given as a cause of low flow into the pump because this is caused by low level in the vessel which is the sensed variable deviation. Low flow in the divider-header combination is given by high flow in the other outlet leg Q 16 HI. This is converted to some flow by the synthesis algorithm which has been developed in another branch.

The sensed variable deviation is low level in the surge drum L 20 LO and there are two variable deviation causes. No inlet flow from the condenser G 9 NONE is traced to events in the condenser which have been developed previously. Low inlet flow to the surge drum G 9 LO is traced to low pressure in the condenser P 6 LO. This is developed as shown in Figure 13.16.

Figure 13.16 - Low Pressure In The Condenser

Low pressure in the condenser P 6 LO has a variety of causes. A high condensation rate Q 7 HI is caused by high flow rate of cooling water through the condenser Q 1 HI or G 2 HI, or low temperature of the water T 1 LO.

Low level in the condenser L 6 LO may be caused by low or no condensation rate Q 7 LO or Q 7 NONE. The former is traced to low flow of cooling water, cooling water at a high temperature, or fouling in the condenser. The latter has been traced further up the fault tree.

If the hand valve, unit 5, on the relief port of the condenser fails or is directed open then this will cause some flow through the non-flow port Q 3 SOME, causing a low pressure.

Low flow of vapour to the condenser G 52 LO is traced back to the divider, unit 44. Here the stream is split, one traces low flow back to the divider for the pressure relief valve and the other is the leg to the surge drum. The former has causes of a leak to the environment or the relief valve being open RV-F-OP. The latter traces the deviation of high flow Q 49 HI. This is the manipulated variable deviation for the second control loop so the template is applied.

The spontaneous failures branch is straightforward. The sensed variable deviation branch traces low pressure back to the relief valve where a leak or the relief valve failing open are causes. The manipulated variable deviation is traced to a leak in the control valve.

No flow of vapour to the condenser G 52 NONE is again traced back to the divider. No flow from the reaction vessel direction G 48 NONE is traced to high flow in the relief valve leg Q 39 HI which has already been developed. Another cause of no flow to the condenser is high flow in the leg to the surge drum Q 49 HI which has already been developed.

This completes the fault tree for low level in the reaction vessel.

13.4.3 Conclusions

The reaction vessel cooling system example has highlighted a few areas for concern in the methodology of the Faultfinder package.

The pressure and flow relationship in vessels is quite complex and although a general methodology has been developed there are still some residual questions to be answered. While the effect of level on pressure is modelled, the effect of pressure on level is not directly modelled. The methodology achieves the result by modelling the effect of pressure on the inlet and outlet flows. This ultimately affects the level. This system was used in the original methodology. One desirable effect of this approach is that it avoids any 'looping' due to the interaction between level and pressure. What has not been explored is whether this is the best solution and whether this is an instance of a potentially more general problem of interaction between variables and possible resultant looping problems.

The pressure control system actually regulates the pressure in the surge drum but the current program does not permit this. Only those variables with indices less than 10 can be incorporated into the control methodology. Fortunately this is no more than a nuisance because it is possible to use the pipe pressure on the inlet to the drum.

There also exists uncertainty in the ability for the package to deal correctly with trip systems such as the one found in this example. The trip initiates changes in the state of some of the models which may or may not have a bearing on the fault tree. For the given fault tree there do not appear to be any problems.

13.5 AKZO Chlorine Storage Facility

This plant was studied as part of the risk analysis report to the Rijnmond Public Authority (6) and formed part of a final year project using the Faultfinder package.

The chlorine is produced on site and stored at atmospheric temperature in a set of five horizontal vessels, pressurised to 9 bar, with a capacity for 100 tonnes each. All the vessels are identical but one remains empty at all times to act as a dump tank should one of the others develop a fault.

The facility can be operated in various ways but for the purpose of this example a single vessel is considered undergoing a filling operation at 6.5 bar. A schematic representation of the vessel developed from the description given in the report is shown in Figure 13.17.
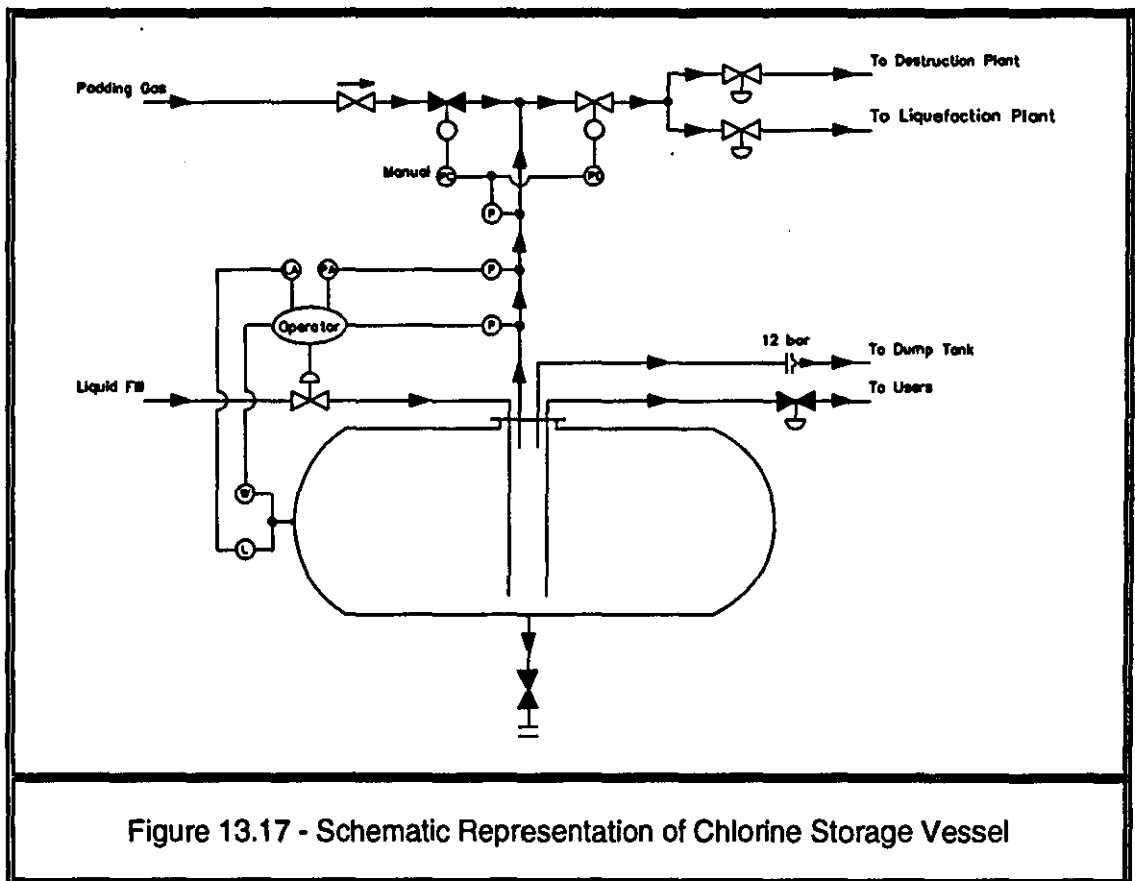
Figure 13.17 - Schematic Representation of Chlorine Storage Vessel

The vessel has a single manhole through which four pipes enter. In addition to this there is a single exit pipe at the base connected to a closed hand valve and a blank flange. Of the four pipes entering through the man-hole two project to the base of the vessel. One is for the liquid chlorine inlet from the production plant and the other is for the liquid chlorine outlet to the user facilities.

The other two pipes remain in the gas space above the liquid. The first of these extends to a series of pressure sensors, valves and connections used to maintain a constant pressure. The first two pressure sensors are connected to a pressure gauge and a pressure alarm gauge set at 11 bar in the control room. The final sensor passes to a pressure controller which gradually releases the padding gas to either the destruction plant or the liquefaction plant. The sensor is also connected to a controller on the chlorine padding gas inlet but during the filling stage this is set to manual and the valve remains closed.

The last pipe through the manhole is connected to a bursting disk set at 12 bar which is in turn connected to the dump tank. The test pressure of the vessel is 18 bar.

The degree of filling of the vessel is monitored in the control room by means of weight sensors on the vessel legs. The vessel is also fitted with an external level indicator which operates an alarm on high level.

All valves on piping through the manhole are remotely controlled pneumatic valves from the control room and close should the air supply be lost.

13.5.1 Configuration And Modelling

The storage vessel was created using the vessel model generation routine. Only the liquid inlet port and the excess gas outlet port are flow ports and only the liquid inlet port is from a pumped supply, all others are affected by the pressure.

A number of other components had to be created but the models for these could be easily based on the models for other units. Examples were the closed remotely controlled valve and blank flange which are similar to the closed hand valve. The open remotely controlled valve is similar to the open hand valve and the bursting disc is identical to the relief valve. The high alarm is modelled as a trip switch sending out a signal when the input signal is high.

Two other units, the operator and the manual controller, were also created. The operator has four inputs, the high level alarm, the high pressure alarm, the weight indicator and the pressure indicator. The unit is modelled as a trip switch sending a signal to the trip valve should the alarm emit a signal or the indicators produce a high signal. The manual controller simply has an event statement which attributes the controller being set to automatic mode by the operator and a decision table ANDing this with some form of input signal producing an output signal.

The configuration diagram is shown in Figure 13.18.
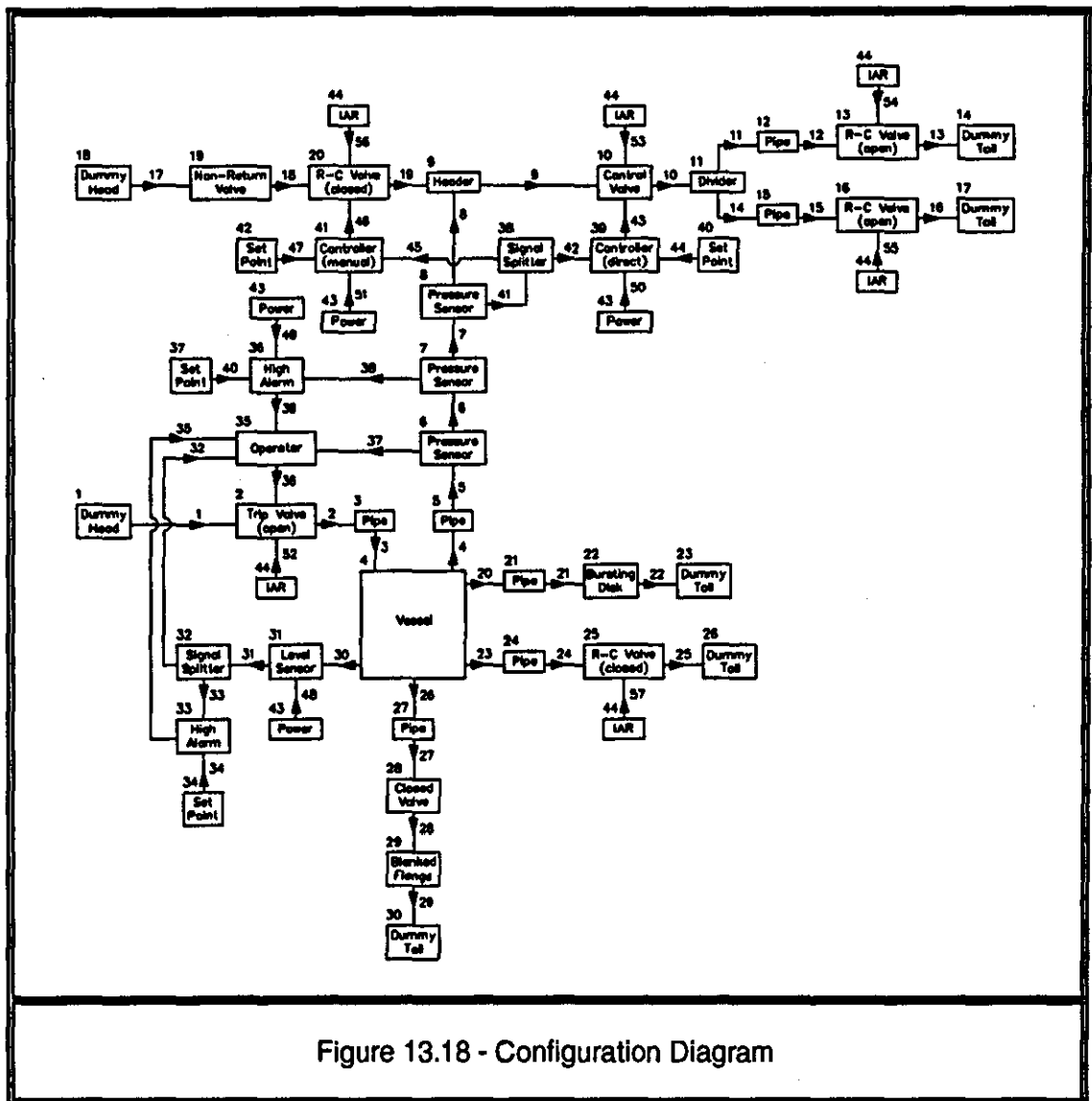
Figure 13.18 - Configuration Diagram

Determination of the configuration sub-systems is straightforward. The closed remotely controlled valve and the manual controller do not form a control loop, which leaves just the one in the configuration. There are four trip systems all operating on the same trip valve via the operator.

The level and the weight in a vessel have the same propagation equations and event statements, therefore the configuration can be simplified by using the same sensor and variable for both. The output from the sensor for the level is split producing separate inputs to the operator for the weight indicator and the level alarm.

The purpose of this example is to illustrate the ability of the package to produce risk assessment type fault trees as produced for the Rijnmond report (7). These involve linking together basic fault type events rather than variable deviations. The top event chosen is vessel rupture VES-RUPT, and the model used is

V X6VES HI : H2O-PRES	high concentration
I H2O-PRES : CORROSN	water present
F EXT-HEAT : CORROSN	external heat source
F VIBRATN : FATIGUE	vibrations
I CORROSN : A(DUMMY)	corrosion
I FATIGUE : A(DUMMY)	metal fatigue
F FLASHING : LOW-TEMP	flashing liquid
F EXT-COLD : LOW-TEMP	external cold source
I LOW-TEMP : BRITFRAC	low temperature
V R4OUT NONE : NORELIEF	no relief through port
V L6VES HI : OVR-FILL	high level
I OVR-FILL : OVERPRES	over filling
F EXT-IMP : SUP-FAIL	external impact
F FLOODING : SUP-FAIL	flooding
F SUBSDNCE : SUP-FAIL	land subsidence
F EXT-EXPL : SUP-FAIL	external explosion
F MAJ-FIRE : SUP-FAIL	major fire
I OVERPRES : CRIT-STR	over pressure
I SUP-FAIL : CRIT-STR	support failure
I MECH-DEF : VES-RUPT	mechanical defect
I BRITFRAC : VES-RUPT	brittle fracture
I CRIT-STR : VES-RUPT	critical stress

I NORELIEF V P6VES HI T OVERPRES	no relief, high pressure
I A(DUMMY) F INSP-ERR T MECH-DEF	dummy event, inspection error

All the faults to be linked together have to be contained in the top event model along with the variable deviations which are to be traced by the package. This model is specific to the situation being studied and cannot be used for other examples.

13.5.2 Fault Tree Synthesis

The top part of the fault tree is shown Figure 13.19. This is similar to the fault tree produced for the Rijnmond report, part of which is shown in Figure 12.2 on page 12-3 of this thesis.
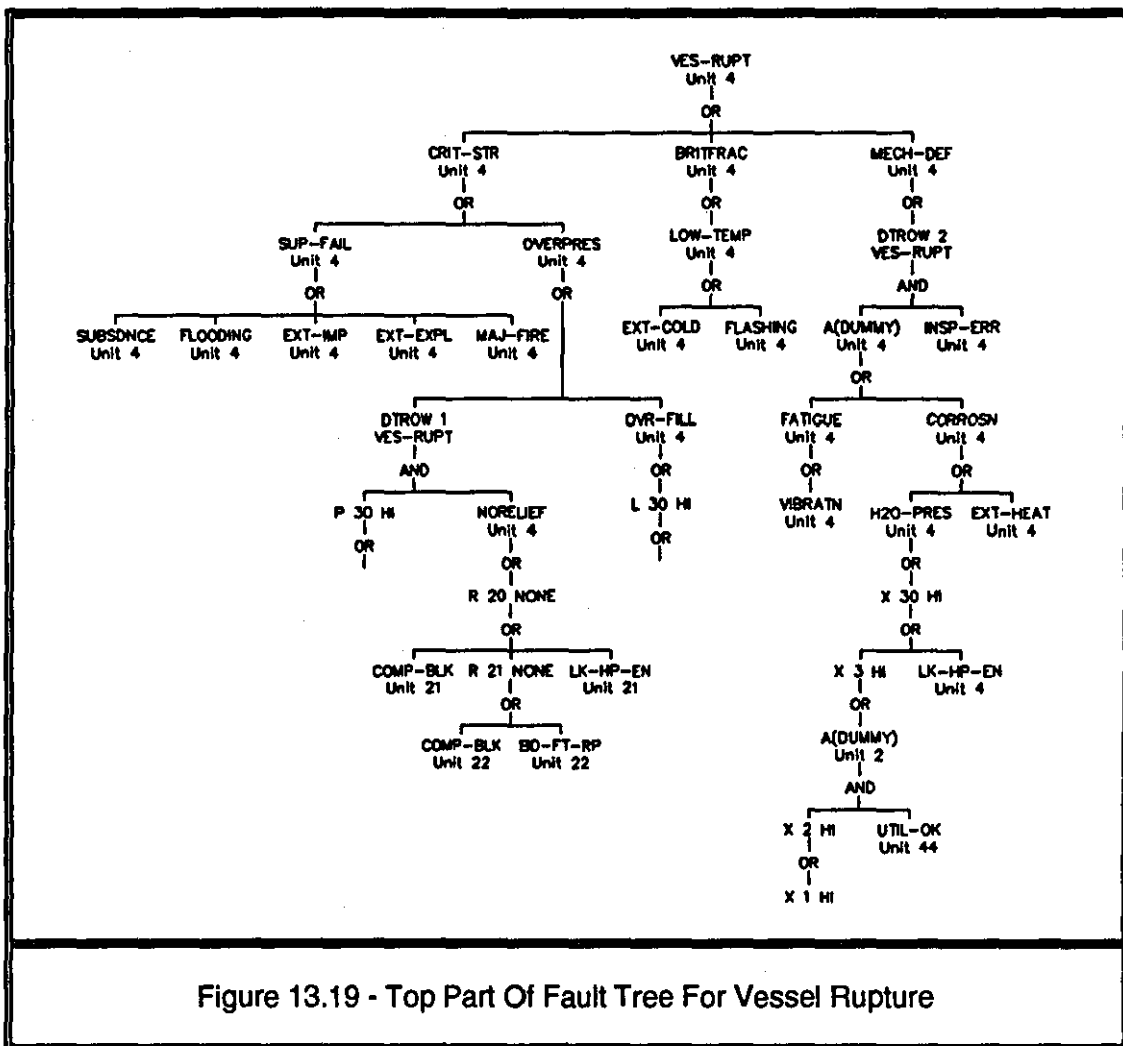
Figure 13.19 - Top Part Of Fault Tree For Vessel Rupture

This shows the way the basic events are linked together to form the actual top event. The mechanical defects branch has causes given by the second decision table in the top event model. These are the intermediate dummy event A(DUMMY) and the fault inspection error INSP-ERR. The intermediate event combines the faults corrosion CORROSN and fatigue. The latter is caused by vibrations VIBRATN and the former an external heat source EXT-HEAT or water present in the vessel H2O-PRES. The water present fault is converted to the variable deviation X 30 HI which is traced back to leaks in the liquid inlet pipe or water entering with the liquid chlorine X 1 HI.

The brittle fracture fault BRITFRAC is made up entirely of basic faults. A brittle fracture may be caused by a low temperature LOW-TEMP, which has causes of an external cold source EXT-COLD or the liquid chlorine flashing in the vessel.

Causes of the vessel reaching its critical stress are support failure SUP-FAIL or an over pressure OVERPRES. The former has a list of five causes under an OR gate. Over pressure may be caused by over filling OVR-FILL, which converts to high liquid level in

the vessel L 30 HI, or a combination of high pressure P 30 HI ANDed with no relief which in turn converts to R 20 NONE. The no relief deviation is traced to the pipe being blocked or the bursting disc failing to rupture BD-FT-RP.

The high level and high pressure deviations are traced to their causes in the normal way and present no new synthesis techniques. For this reason only a brief description of their causes is given and they are not shown in the figure.

The high level deviation is traced to high flow of liquid chlorine at the inlet Q 3 HI. However, high flow is a protectable event by all four trip loops, therefore, the branch is ANDed with each trip loop functional failure. The branches are similar to other trip loop failures.

The high pressure deviation has causes of high level, high temperature, or low or no flow in the gas outlet pipe. The gas outlet is manipulated by the control loop so the appropriate control loop template is used on each deviation.

This completes the fault tree for vessel rupture.

13.5.3 Conclusions

The AKZO chlorine storage facility illustrates the adaptability of the Faultfinder package to produce the kind of fault trees used in hazard assessments. This can be seen by comparing the Faultfinder fault tree and the Rijnmond report fault tree. Both trees contain various levels of mechanical faults linked together, some of which are eventually traced to variable deviations.

However, it could be argued that by the time the top event has been modelled and entered into the computer the fault tree could be developed manually. This suggests that if there is a high degree of interaction between mechanical faults and variable deviations to model then there is not much benefit in using an automated approach because the amount of effort saved is not great.

An advantage of using an automated system would be gained if the mechanical faults could be kept separate from the variable deviations. These could then be retrieved from a file and combined with the top event when constructing the fault tree. Insufficient time on this project has not allowed this idea to be developed any further.

The main advantage of using the Faultfinder package is its ability to produced multiple fault trees for different top events very quickly and easily once the initial configuration data and models have been entered.

13.6 British Gas LNG Vaporiser System

The research into and the development of the Faultfinder package has been supported both financially and technically with examples by British Gas. Mullhi, a previous worker, studied an example of a butane vaporiser and published a paper (8). In order to demonstrate the ease of modelling similar systems a second vaporiser example was put forward by British Gas. This time a liquid natural gas (LNG) vaporiser was used. The P & I diagram is given in Figure 13.20.

Figure 13.20 - British Gas LNG Vaporiser System
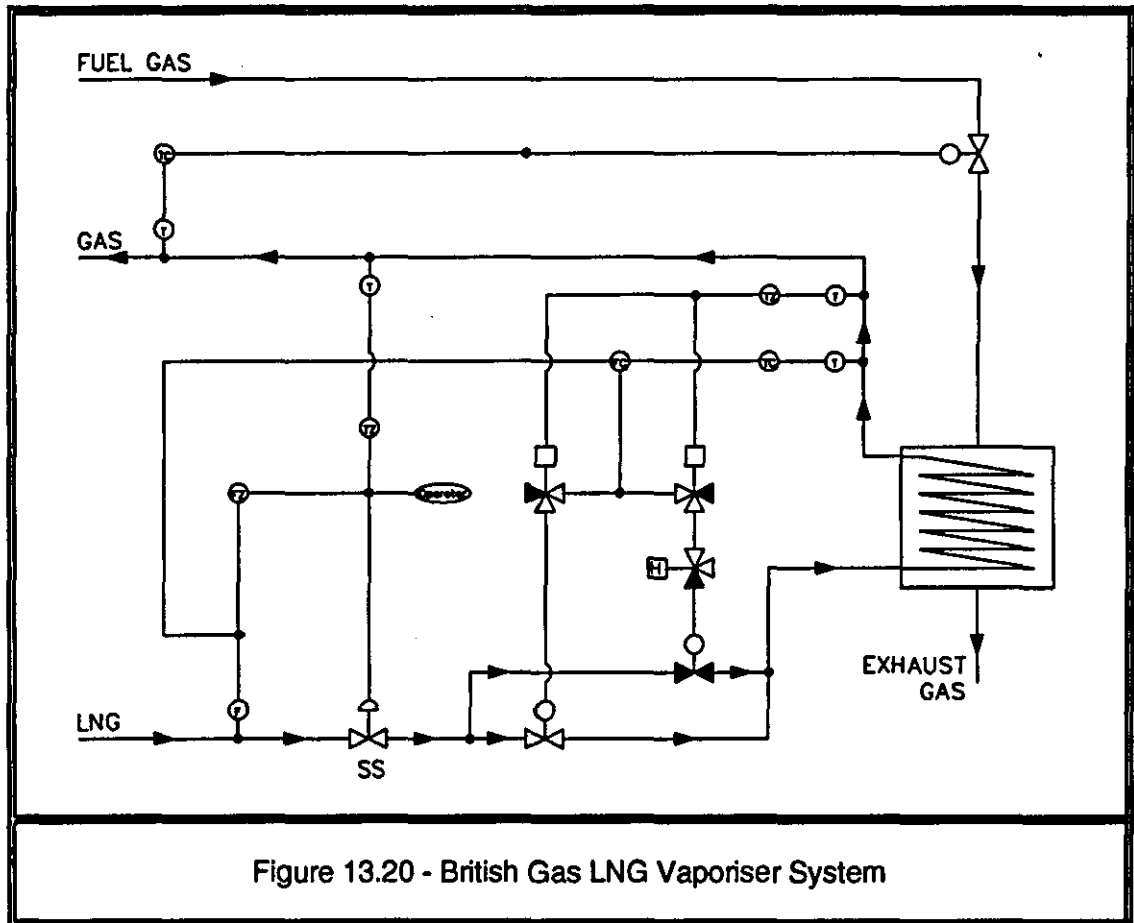
Liquid natural gas is pumped from storage to the vaporiser. The inlet line contains a divider-header combination before the vaporiser. Each leg of the combination contains a control valve, however, only one leg contains flow at any one time. The pneumatic signal to the second control valve is vented to the atmosphere by a manually set vent valve. The loss of air to the valve causes it to stay shut.

The flow of LNG to the vaporiser is controlled by a flow rate control loop on the inlet stream. However, should the vaporiser burners be unable to cope an over-ride temperature controller operates. The flow of fuel gas to the burners is manipulated by an independent temperature controller. Should the control fail a temperature trip system operates to vent

the control signal from the control valve on the LNG inlet stream causing the valve to shut. As a further precaution a second temperature trip system has been installed which closes a slam shut valve on the LNG inlet stream. This valve can also be closed by an operator and is also connected to a high flow trip which operates on very high flow in the inlet stream which may be caused by catastrophic failure of the pipeline.

Although the two systems are very similar the protection devices operate for different reasons. The purpose of the butane vaporiser system is to provide a butane vapour supply to supplement the natural gas supply at times of peak demand. A pump, capable of delivering liquid butane at 300 psig, supplies a vaporiser rated at 250 psig. Hence the function of the protective devices is to prevent the vaporiser from being overpressured. In the LNG vaporiser the protection is against liquid entering the exit line, detectable by a low temperature in the exit pipeline.

13.6.1 Configuration And Modelling

The configuration diagram for the system is shown in Figure 13.21 (overleaf).

Figure 13.21 - Configuration Diagram

13-32a
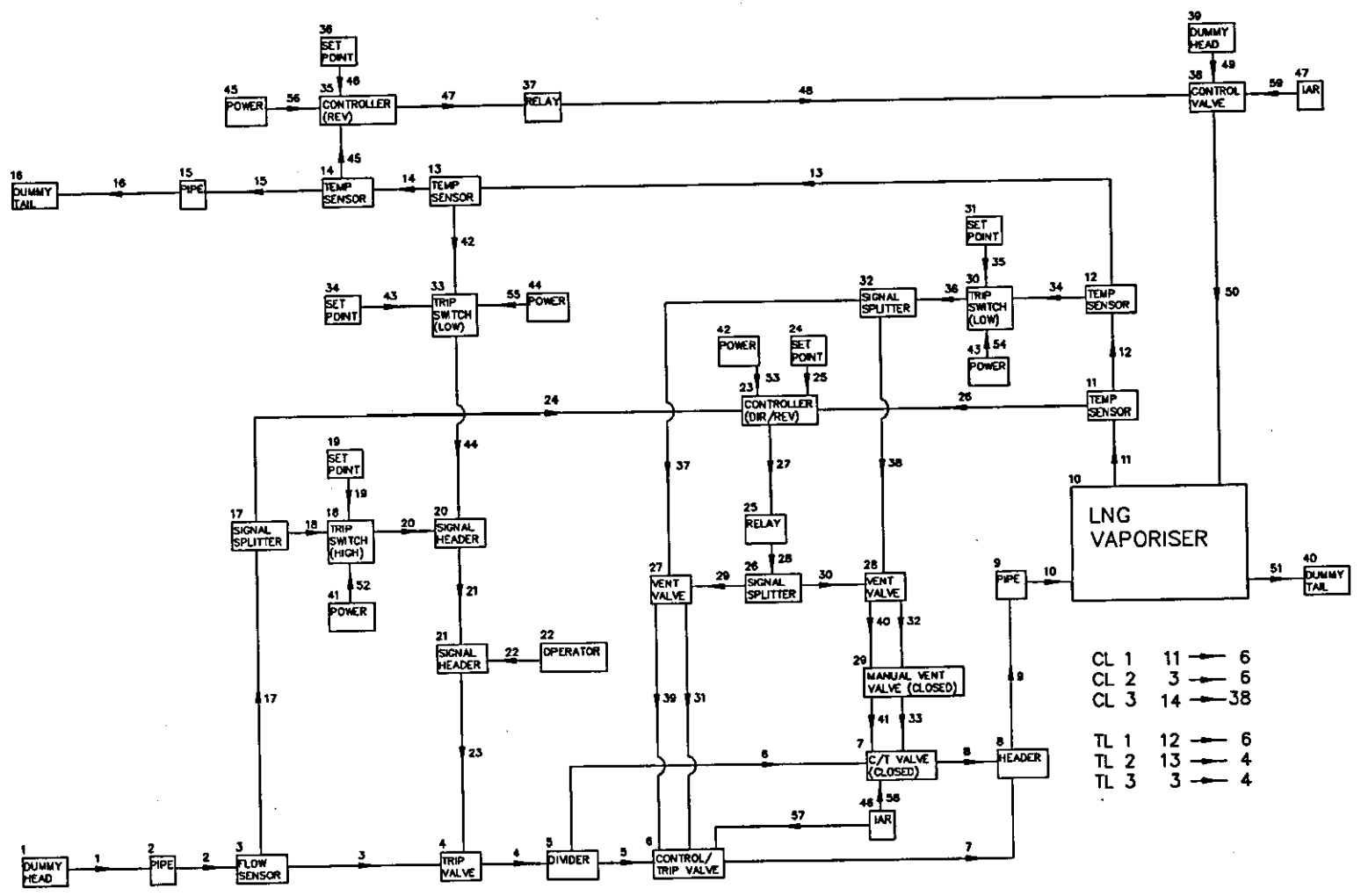
All models for this configuration are contained in the unit model library except for the vaporiser itself. This model was created using the heat exchanger model generation routine. Heat exchanger characteristics were used because the process of burning fuel gas to vaporise a process fluid can be reduced to heat transfer between the two streams. The temperature of the vaporised gas is modelled as a function of the inlet flow rate and temperature, and the fuel gas inlet flow rate and temperature.

T2OUT= F(-G1IN, T1IN, G3IN, T3IN)

Hence, high flow of LNG or low inlet temperature of LNG will cause a low exit temperature. Also a low flow of burner fuel or low temperature in the burner fuel will cause a low exit temperature.

In using a heat exchanger model for the vaporiser the processes of vaporisation and burning have been greatly simplified. On the vaporisation side it has been assumed that the effect of the phase change on the heat transfer efficiency is relatively constant and can be ignored. On the burner side the process of burning the fuel is again assumed to have a constant effect on the heat transfer efficiency and can be ignored. The only change to the model is the addition of an event statement which relates burner failure to low exit temperature.

Although all the other models were contained in the unit model library, some had to be altered to take into account improvements in the modelling methodology implemented during this project. The main ones were the combined control/trip valves in the operating and closed states and the divider and header models. Details of these changes can be found in Section 11.3 and Chapter 10.

Two points to note about the configuration are the use of a combined control/trip valve model and the use of two connections between the vent valve and the combined control/trip valve. The combined model is a control valve model but with the inclusion of trip valve faults. However, instead of using trip valve fails open TV-F-OP the control/trip valve faults of fails high aperture CTV-F-HA, stuck CTV-STK and low aperture CTV-F-LA are used. This was the case in the butane vaporiser example.

The two connections between the models are required because of the methodology used for control loops and trip loops. One connection is for the control signal and one for the trip signal. The reasons for this are given in Section 11.3.

The relay model which sits between the controller models and the control/trip and control valve models is a simple unit which converts an electrical signal to a pneumatic signal.

The configuration has three control loops, the flow controller (loop 2), the over-ride temperature controller (loop 1) and the fuel gas flow controller (loop 3). The over-ride controller is specified first because this is the master loop in a master-slave system. The slave loop is the flow controller.

CONTROL LOOP

NUMBER: 1
SENSED VARIABLE: T
VARIABLE SENSED IN UNIT: 11
CONTROL VALVE UNIT NUMBER: 6
OTHER UNITS IN CONTROL LOOP: 23 24 25 26 27 42 46
VARIABLE T REGULATED IN CONNECTIONS: 11 12 13 14 15 16
FLOW MANIPULATED IN CONNECTIONS: 5 7
LOOP IS NOT OF FEEDFORWARD TYPE

NUMBER: 2
SENSED VARIABLE: Q
VARIABLE SENSED IN UNIT: 3
CONTROL VALVE UNIT NUMBER: 6
OTHER UNITS IN CONTROL LOOP: 17 23 24 25 26 27 42 46
VARIABLE Q REGULATED IN CONNECTIONS: 1 2 3 4
FLOW MANIPULATED IN CONNECTIONS: 5 7
LOOP IS NOT OF FEEDFORWARD TYPE

NUMBER: 3
SENSED VARIABLE: T
VARIABLE SENSED IN UNIT: 14
CONTROL VALVE UNIT NUMBER: 38
OTHER UNITS IN CONTROL LOOP: 35 36 37 45 47
VARIABLE T REGULATED IN CONNECTIONS: 11 12 13 14 15 16
FLOW MANIPULATED IN CONNECTIONS: 49 50 51
LOOP IS NOT OF FEEDFORWARD TYPE

The link between the controller and the closed control/trip valve is not specified as a control loop because under normal operation the valve is closed and the control signal is vented to the atmosphere. Therefore, this control loop is not functioning and any failures will not affect the system

The flow control loop is unusual because it has flow as both the regulated and manipulated variable. Usually flow is one or the other. For example, in a simple feedback flow control

loop, with the sensor and the control valve in the same pipeline, the flow is, or is treated as, regulated and in a simple feedforward control loop between two streams, with the sensor and control valve in separate pipelines, the flow is, or is treated as, manipulated.

In this example, although the sensor and control valve are in the same pipeline, the divider-header combination in which the control valve sits acts as a flow transition point. Therefore the flow is regulated in streams 1 to 4, which is where the sensor is situated, but the flow is manipulated in the divider-header combination leg, streams 5 and 7. The other leg provides a cause for the flow deviation to change.

The configuration has three open valve trip systems. Again the system connected to the closed control/trip valve is not specified. The three loops are the low temperature trip to the control/trip valve (trip 1), the low temperature trip to the slam shut valve (trip 2) and the high flow trip to the slam shut valve (trip 3).

OPEN VALVE TRIP SYSTEM

NUMBER: 1
TRIP VALVE UNIT NUMBER: 6
OTHER UNITS IN TRIP SYSTEM: 12 27 30 31 32 43 46
TRIP IS NOT OF FEEDFORWARD TYPE

NUMBER: 2
TRIP VALVE UNIT NUMBER: 4
OTHER UNITS IN TRIP SYSTEM: 13 20 21 33 34 44
TRIP IS NOT OF FEEDFORWARD TYPE

NUMBER: 3
TRIP VALVE UNIT NUMBER: 4
OTHER UNITS IN TRIP SYSTEM: 3 17 18 19 20 21 41
TRIP IS NOT OF FEEDFORWARD TYPE

The top event of interest for this configuration is low temperature in the exit pipeline from the vaporiser.

13.6.2 Fault Tree Synthesis

The top of the fault tree is shown in Figure 13.22.
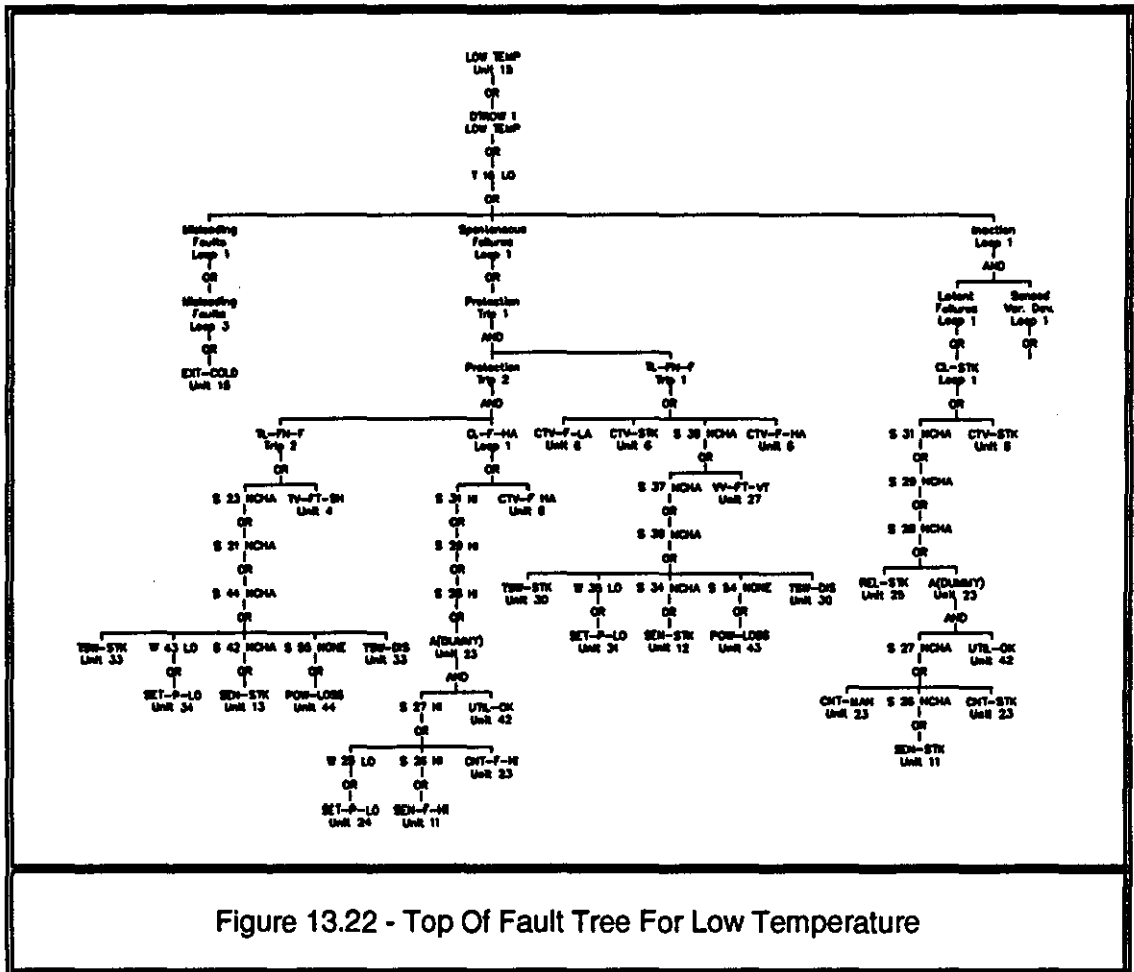
Figure 13.22 - Top Of Fault Tree For Low Temperature

Temperature in the outlet pipe line is the regulated variable of control loops 1 and 3. The control loop template for loop 1 is applied first. The misleading faults are those which occur after the sensing range of the loops. In this case it is an external cold source at the pipe which is also a misleading fault of control loop 3.

The spontaneous failures of the loops are protectable by trip systems 1 and 2. Therefore, the trip functional failures TL-FN-F are ANDed into this branch. Trip 1 is on the control/trip valve and has the control/trip valve open faults whereas trip 2 is on the slam shut valve and has the single valve failing to shut fault. It is the *no change* NCHA deviation which is traced for the functional failure branch. The end causes are trip switch stuck, sensor stuck, power loss or set point low.

The control loop failing high and the latent failures branches are traced to the normal causes.

The sensed variable deviation branch is shown in Figure 13.23.
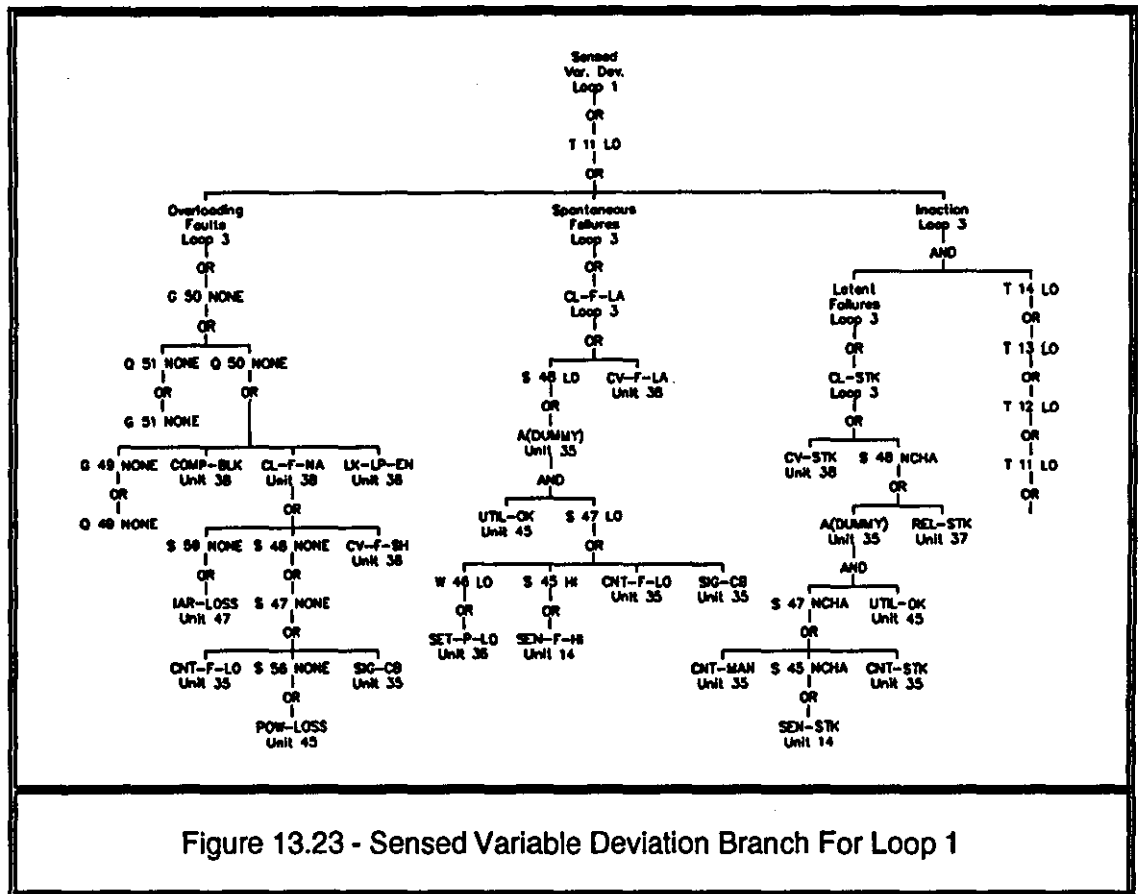
Figure 13.23 - Sensed Variable Deviation Branch For Loop 1

This part of the fault tree shows the application of the control loop template for loop 3. The overloading branch traces the causes of no flow of fuel gas. Control loop 3 failing with no aperture is identified as a cause of this. Again the spontaneous and latent failures branches are straight-forward.

The sensed variable deviation branch is shown in Figure 13.24.

This deals with the interaction of the two streams in the vaporiser. Causes of low temperature at the outlet are fouling, an external cold source or burner failure. Of the variable deviation causes low flow or temperature of the fuel gas supply are identified. There is also low temperature of the LNG supply which is traced to low temperature at the source. A low temperature through the non-flow leg of the divider-header combination must be accompanied by a failure of the control/trip valve or the manual vent valve.

13-37
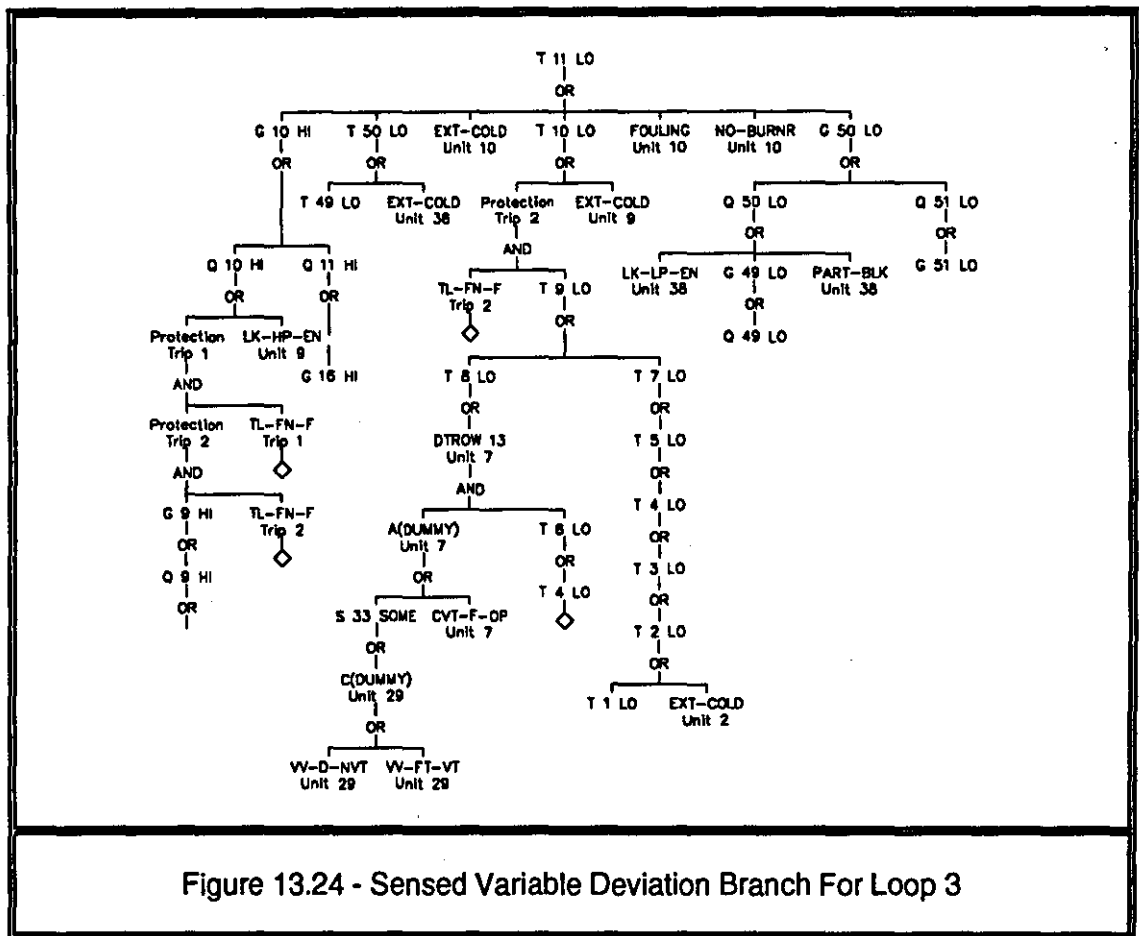
Figure 13.24 - Sensed Variable Deviation Branch For Loop 3

High flow rate of LNG to the vaporiser may also cause low temperature and this may be caused by high flow in the outlet stream. On the other hand it may be caused by high flow at the divider-header combination. This branch is continued in Figure 13.25.

The causes of high flow are high flow in the flow leg or some flow in the non-flow leg. The flow leg is manipulated by control loop 2 so the template is applied. The spontaneous and latent failures branches are simple and the manipulated variable deviation branch is traced to high flow at the source.

Any flow in the non-flow leg must be ANDed with failure of the control/trip valve or manual vent valve. The some flow deviation is traced to high flow at the inlet which is traced to high flow at the source.

This completes the low temperature fault tree for the LNG vaporiser example.
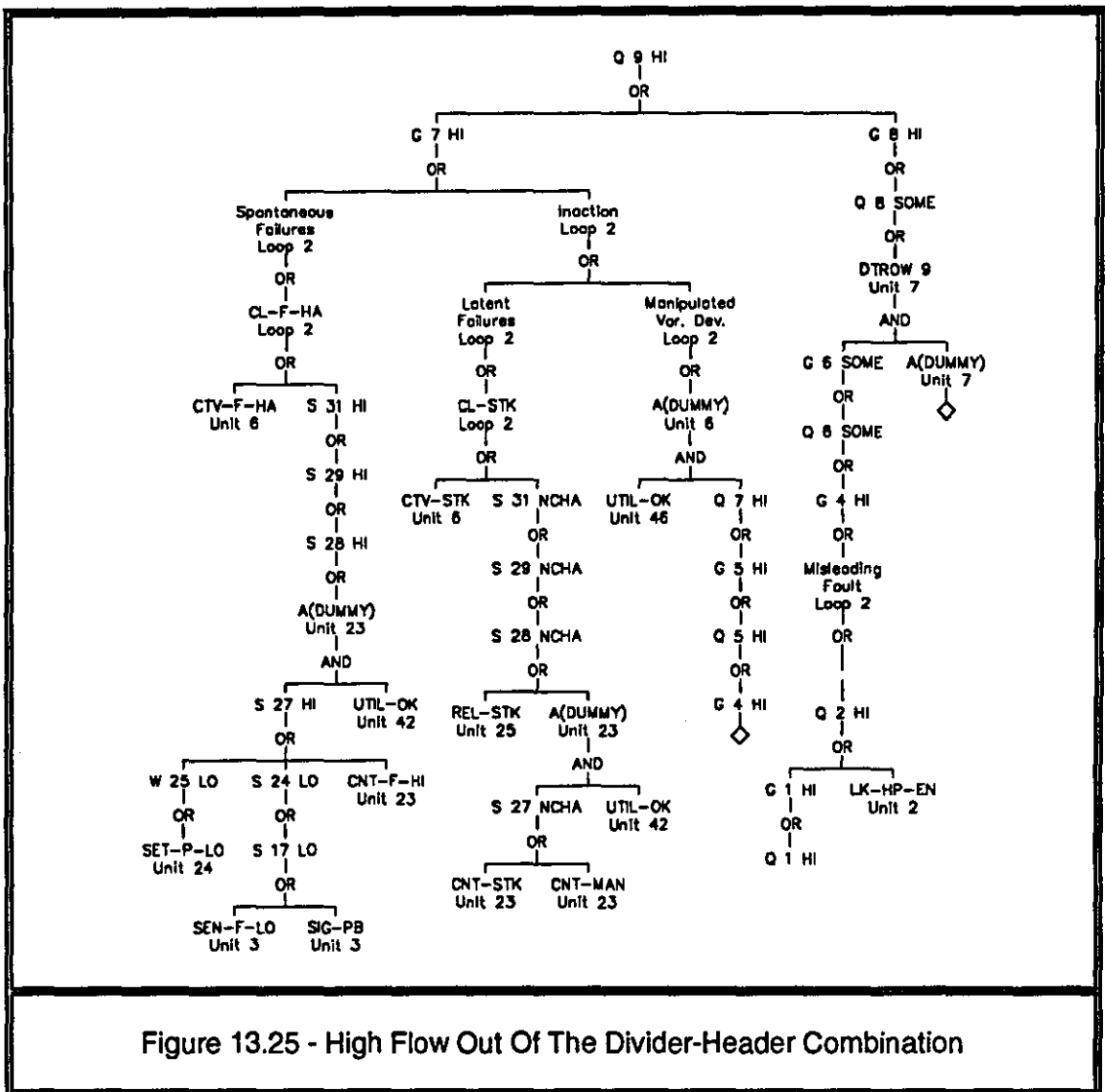
Figure 13.25 - High Flow Out Of The Divider-Header Combination

13.6.3 Conclusions

The LNG vaporiser example illustrates the relative ease with which it was possible to model the system based on the previous butane vaporiser. Although the two had different priorities for control, only the actual vaporiser unit itself required a new model.

Problems not foreseen in the butane vaporiser example were discovered which resulted in improvements in the modelling methodology for dividers, headers and closed valves. These dealt primarily with the non-flow leg where the *some* deviation was involved. To prevent the *some* branch being deleted when the trace exited from the divider-header combination it is necessary to change the deviation to high flow. This is explained in Section 10.3. No other problems were encountered.

13.7 Summary

The five examples given in this chapter cover the main aspects of the work carried out during this project. These are the:

- modelling relationship of flow and pressure in vessels
- specification of a vessel model taxonomy
- creation of a set of vessel model templates
- creation of a vessel model generation routine
- creation of a heat exchanger model generation routine
- improved modelling of dividers, headers and their combinations, especially with reference to the non-flow legs
- improved modelling of closed valve type units
- standardisation of the unit model and top event model libraries.

The following table gives a summary, for each example, of the number of units, connections and special sub-systems (control loops, trip loops, divider-header combinations and secondary failures), the number of new models required, excluding those requiring modifications due to changes in the methodology, and whether either the vessel model or heat exchanger model generation routines were used

| Example | Number of | | | | | |
| | | | | | Generation Routine Uses | |
	Units	Connections	Sub-systems	New Models	Vessel	Heat Exchanger
13.2	19	22	2	0	0	0*
13.3	33	36	3	5	3	0
13.4	55	61	4	3	1	2
13.5	44	57	5	8	1	0
13.6	47	59	7	1	0	1

* The heat exchanger model for the nitric acid cooling system was already contained in the model library but could easily be generated using the heat exchanger generation routine.

Of the seventeen new models developed for these examples, eight were either vessel models or heat exchanger models, all of which could be generated automatically. Most of the remaining models could be derived from existing models in the library or contained very simple modelling.

The two additional new models in example 13.3, the reaction vessel charging system, were unique to this configuration. One was the modified level sensor which contained supplementary event statements to model a hand valve being closed and the effects of a partial or complete blockage in the unit. The other model was the pipe tracer unit which simply converted a utility signal to an electric current signal. Neither of these models posed any difficulties.

The majority of new models required were in example 13.5, the AKZO chlorine storage vessel. This example required seven additional models which had to be developed by the user. However, the new models were very closely linked to existing models in the library and much of the modelling process could be parallelled with these. The closed remotely controlled valve is a closed hand valve with an extra event statement relating some signal input to the valve being open. The blank flange is also based on a closed hand valve but instead of the valve open faults it has flange removed faults. The bursting disk model is the same as the relief valve but the faults relate to a disk rather than to a valve. The high alarm is identical to a trip switch which responds to a high input signal. The operator model is unique to this configuration. It has four inputs, two from the high alarm and two from sensors. An output signal is generated if some signal is received from the alarms or a high signal is received from the sensors. An event statement relates an incorrect output signal to operator error and a decision table is used to relate no output signal when a response is required to an absent operator. The control valve on manual has a set of decision tables for each deviation of the signal variable. For an output signal to occur there must be an input signal and the controller must be set to automatic.

The synthesis of the fault trees for these examples has uncovered a few problems in the methodology and programming which had gone unnoticed. These included uncertainties in the modelling of flow and pressure in vessels. The process works but its application to all situations has not been verified. A similar problem was encountered with the trip system in example 13.4, the reaction vessel cooling system. When the trip operates some of the models change state and it is not known whether the synthesis procedure deals with this correctly. Both these require further investigation.

As the number of variables has increased the index numbers have expanded beyond the one digit representation. The configuration and synthesis programs are unable to model these as control loop variables (see example 13.4) which restricts the capability of the package. Either some extensive re-programming is required or a re-ordering of the variable indices.

The power of the methodology to adapt to different situations has been demonstrated in two ways. First the failure modes of units can be increased without the need to alter the

models by using the secondary failures facility (see example 13.3) and secondly it is possible to develop complete risk assessment type fault trees with strings of mechanical faults linked together simply by using a different top event model (see example 13.5).

13.8 References

1. Lapp S. A., Powers G. L.; Computer-Aided Synthesis Of Fault-Trees; IEEE Transactions On Reliability; R-26; 1977; pp 2–13.

2. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology; 1990; pp 10/1–10/19.

3. Trenchard A. J.; Process Plant Alarm Diagnosis Using Synthesised Fault Tree Knowledge; PhD Thesis, Loughborough University Of Technology; 1990; pp 153–183.

4. Piccinini N., Levy G.; Process Safety Analysis For Better Reactor Cooling System Design In The Ethylene Oxide Reactor; Canadian Journal Of Chemical Engineering; V 62; 1984; pp 541–546.

5. Piccinini N., Levy G.; Ethylene Oxide Reactor, Safety According To Operability Analysis; Canadian Journal Of Chemical Engineering; V 62; 1984; pp 546–558.

6. Khan A. R., Hunt A.; The Propagation Of Faults In Process Plants: Integration Of Fault Propagation Technology Into Computer-Aided Design; In McGreavy C. (eds.), Computer Integrated Process Engineering Conference 1989; pp 35–43.

7. Risk Analysis Of Six Potentially Hazardous Industrial Objects In The Rijnmond Area, A Pilot Study : A Report To The Rijnmond Public Authority; D. Reidel Publishing Company; 1982; pp 151–182

8. Risk Analysis Of Six Potentially Hazardous Industrial Objects In The Rijnmond Area, A Pilot Study : A Report To The Rijnmond Public Authority; D. Reidel Publishing Company; 1982; pp 491.

9. Mullhi J. S., Lees F. P.; The Propagation Of Faults In Process Plants : 5. Fault Tree Synthesis For A Butane Vaporiser System; Reliability Engineering And System Safety, 23; 1988; pp 31–49.

Chapter 14

<u>Conclusions And Recommendations</u>

14.1 Introduction

This thesis has described recent research work carried out on a computer based system for fault tree synthesis at Loughborough University. This has formed part of an on-going programme of work on the FAULTFINDER package spanning several years (1 - 5).

The procedure for fault tree synthesis is:

1. Decompose the piping and instrumentation (P & I) diagram into a configuration diagram consisting of a set of units and the connections between them.

2. For each unit identify a unit model from the library or create a new model if necessary. A unit model is made up of propagation equations, initial event statements and modified decision tables which are converted to a minitree format.

3. Identify all control loops and trip systems. Only the units involved and the variables and connections affected are required.

4. Determine which top event model is to be used. This gives the variable deviation or fault of interest.

5. Synthesise the fault tree using the package.

6. Use the analysis program to determine the minimum cutsets.

7. Repeat steps 4 to 6 for different top events.

The advantage of this methodology is the use of unit models rather than system models as a base for fault tree synthesis because the items are smaller and, therefore, easier to conceive. The models are context independent and can be used in any system configuration as necessary.

Although situated on a DEC Microvax II computer system running VMS, the programs are portable, with a minimum of alterations, to any other computer system or personal computer (PC) with a Fortran compiler.

14.2 Conclusions

A goal in developing this computer package is to make the process of fault tree synthesis as simple as possible and one way to achieve this is to make it fully automatic. This package has progressed towards this aim and is at a stage where, if all the required models are contained in the unit model library, the process *is* automatic requiring only the configuration to be entered. Therefore any work the user has to do centres almost entirely on creating new models. The aspect of this project has been to look at the problems of having to develop a unit model. The research has been concerned primarily with the generation of unit models and, in particular, with the formulation of rules for modelling.

A technique has been devised to make the initial modelling stages easier. The process works by allowing the plant configuration to be broken up into smaller sections which can then be studied separately. The advantage of this is to enable the user to identify any problem areas in the modelling simply by reducing the size of the synthesised fault tree. Once each section is modelled correctly the entire plant can be studied without errors.

The previous methodology for modelling flow and pressure characteristics in vessel type units was not clearly defined and this resulted in various styles of model being used. The flow and pressure relationships have been revised and a set of rules formulated. The result has been the introduction of a set of core propagation equations based on a taxonomy of vessel connections.

A change in the methodology was necessary since vessels had been identified as the most likely unit to require a new model. A further step in the enhancement of the user friendliness of the modelling aspect has been the development of a model generation routine for vessels. This is an interactive facility which can be called up by the user when a new vessel unit needs to be created. The methodology is based on the rules identified for vessels. A set of vessel model templates has been produced simultaneously as a comparison for the generation routine and as an additional aid. A set of edit instructions for these is given so the user can tailor the model to their own requirements.

Another class of models which vary considerably is heat exchangers. Using the rules for vessel models coupled with a similar set involving heat transfer a generation routine for heat exchangers has been developed. It is an interactive facility requiring information available on a P & I diagram. The routine has been used to create six example heat exchanger models for use as a reference base.

A smaller quantity of work has been carried out investigating the rules for modelling reaction vessels. These are also varied and as such will usually be unique to a particular

plant. Hence it is necessary to provide some user aids for their development. A set of rules based on a broad outline of considerations has been presented and two example reaction vessels have been created.

Much of the work on rule development has been based on trials with examples from the literature. This has revealed limitation in the synthesis methodology which has been corrected where possible. These include further revisions to the modelling of dividers, headers and associated units within their combinations and control and trip systems.

Dividers and headers, when combined, form information loops within the plant configuration. A previous worker (5) developed a system to automatically identify such units based on a single model for a divider and one for a header. However, the single models were inadequate for modelling both flow legs and non-flow legs. The models have been amended to enable a correct modelling approach.

The associated units within divider-header combinations refer particularly to closed valve type units usually found in the non-flow branch. The models have been altered to correctly model the some flow deviation and the valve open faults. However, further attention is required on this aspect.

Control loops of various types and complexities are correctly modelled by the package. However, the effect of having an over-determined control system was unknown. These systems have more control relationships than control variables. An investigation has been carried out and it has been concluded that over-determined control loops do not pose any problems.

Further work has looked at combined control/trip valve systems. These are protective systems which have a single valve for both control and trip operation. This has posed problems in the past with mutually exclusive events appearing in the fault tree and attempts have been made to remedy them. The solutions were incomplete but an investigation has shown that while the events remain in the fault tree the cutset analysis program can deal with them and mutually exclusive events do not appear in the minimum cutsets. However, this has been identified as an area requiring further attention.

The adaptability of the methodology to produce fault trees for different situations has been investigated. The basic format is a fault tree with variable deviations for transmissive events and basic faults for terminal events. However, fault trees used in hazard assessment techniques not only model deviations in plant variables but also mechanical and design faults. These faults may in turn be caused by variable deviations. Hazard assessments are important because they form part of the safety case for a formal safety assessment of any hazardous location. Such locations include storage facilities and

14-3

pumping stations. A technique has been developed which enables mechanical faults to be linked together and has been achieved by incorporating them into the top event model rather than the unit model.

Other work has been carried out improving the capabilities and user friendliness of the programs. This includes increases in the acceptable sizes of the unit models and system configurations and an increase in the possible size of a fault tree. This has been possible through advances in computer hardware facilities. The user friendliness of the programs has been increased with the addition of answer options presented to the user at the prompts for information.

14.3 Recommendations For Future Work

The FAULTFINDER package has been shown to be versatile with an ability to cope with many plant configurations. However, improvements have been identified which may enhance the capabilities of the package but have not been implemented due to the time limit on this project. The recommendations fall into six categories: new methodology; programming of methodology already developed; rationalisation of programmed methodology; improved program structure, improved software compatibility and improved software presentation.

Sections of the methodology which need to be defined are:

- The handling of divider-header combinations with more than two parallel legs of which one has no flow eg. three pumps in parallel with one on standby (see Section 10.6).
- The search procedure for the automatic identification of control loops (see Section 4.5.2).
- The search procedure for the automatic identification of trip systems (see Section 4.5.2).

Sections of the methodology which need to be programmed are:

- The effects of control loop overload in the control loop template methodology (4).
- The simplification of the flow capacity definitions for the divider-header combinations by implementing a routine to identify divider-header pairs such that the flow capacity in adjoining legs automatically get the same value (see Section 10.2.3 and 10.5).

Sections of the programmed methodology which need to be rationalised are:

- The treatment of non-flow legs in divider-header combinations and in particular the consistency checking of AND gates with respect to closed valves (see Section 10.3.2).
- The synthesis process involving combined control/trip valves and in particular the effect of mutually exclusive events (see Section 11.3).

Improvements which need to be implemented into the programs are:

- The re-ordering of the variable indices such that all variables affected by control and trip systems can be defined properly. This requires considerable program investigation because many of the rules are written into the program code (see Section 13.4.1).
- A feasibility study into increasing the number of ports on a unit from 9 to double figures. Again considerable program investigation is required (see Section 7.3.1 and 8.3.1).
- The re-ordering of the fault event library list to use the new ordered format. Again considerable program investigation is required (see Section 5.4).
- The re-ordering of the unit model library list to use the new format. All of the worked examples will have to be converted (see Section 5.6).

Improvements to the compatibility are:

- The linking to a CAD system to enable the configuration to be derived from a computer generated P & I diagram (see Section 4.5.3).

Improvements to the presentation are:

- The adoption of a PC based version considering the widespread acceptance of these machines. This is possible because the disk space and memory requirements are not large.
- The program interface is in basic text format and should be improved possibly with the adoption of a PC windows system and a menu driven response format.
- The programs themselves have had improvements added to them and been edited in an unstructured way which, over many years, has resulted in an opaque mass. They would greatly benefit from a rewrite and even a conversion from Fortran to a quicker programming language.

Currently there are no problems imposed by the dimensionality aspects programmed into the code such as the number of connections affected by a control loop (Section 4.4.4). However, at a later date this may not be true. The restrictions are embedded in the code

14-5

and are indicative of the Fortran program language. Conversion to another language which does not use an array type structure for data storage would be advantageous in this situation.

14.4 References

1. Andow P. K.; A Method For Process Alarm Analysis; PhD Thesis, Loughborough University Of Technology, 1973.

2. Martin-Solis G. A.; Fault Tree Synthesis For Real Time And Design Applications On Process Plant; PhD Thesis, Loughborough University Of Technology, 1978.

3. Shafaghi A.; Plant Modelling For Safety System Analysis; PhD Thesis, Loughborough University Of Technology, 1982.

4. Kelly B. E.; The Modelling Of Fault Conditions In Chemical Process Plant; PhD Thesis, Loughborough University Of Technology, 1986.

5. Mullhi J. S.; Rules For Computer-Aided Synthesis Of Fault Trees For Process Plant; PhD Thesis, Loughborough University Of Technology, 1990.

APPENDICES

Appendix A

System Decomposition And Data Input

A.1 Break Point Subroutine

```
        SUBROUTINE BREAK (DATFIL)
C       -------------------------------------------------------------------------
C       Subroutine to collect data on which units become dummy heads,
C       which become dummy tails and which are ignored completely when
C       setting up break points in a configuration.
C       -------------------------------------------------------------------------
        INTEGER PMAX,TOPOL(400,20),STREA(500,30),HEAD(15),TAIL(15),
     *  CONTRIP(30,5),PPEF(20,10),MFEF(20,5),NDATA(20)


        COMMON/DATA1/NUN,NST,PMAX,NHEAD,NTAIL,NDH,NLOOP,NZTRP,
     *  NTRIP,NMF,NPP
        COMMON/ARRAY1/TOPOL,STREA,HEAD,TAIL,CONTRIP,PPEF,MFEF,NDATA


        INTEGER HD,TL,DUD,NEWHD(30),NEWTL(30),REDUN(100),ANSWER
        CHARACTER DATFIL*10,BRKFIL*10,OPT*4


1       FORMAT(/,' ',A,' ? ',$)
2       FORMAT(/,' ',A)
3       FORMAT(' ',A,' ',$)
4       FORMAT(20I4)
5       FORMAT(/,' ',A,A)
C       -------------------------------------------------------------------------
C       Is data read in from terminal or from file.
C       -------------------------------------------------------------------------
        PRINT3,'IS THE DATA ENTRY FROM TERMINAL OR FILE, T/F ?'
        OPT='TF'
        IF(ANSWER(OPT).EQ.1)THEN
C       -------------------------------------------------------------------------
C       Read in from file DATFIL//'.BPT'
C       -------------------------------------------------------------------------
        BRKFIL=DATFIL(1:LENGTH(DATFIL))//'.BPT'
        OPEN(UNIT=5,FILE='[.RESULTS]'//BRKFIL,STATUS='OLD',ERR=500)
        READ(5,4,ERR=550)HD,TL,DUD
        READ(5,4,ERR=600)(NEWHD(I),I=1,HD)
```

```fortran
      READ(5,4,ERR=650)(NEWTL(I),I=1,TL)
      READ(5,4,ERR=700)(REDUN(I),I=1,DUD)
      CLOSE(5)
      PRINT*,'BREAK POINT DATA TRANSFERRED.'
      ELSE
C     -----------------------------------------------------------------
C     Read in data from terminal.
C     -----------------------------------------------------------------
      PRINT2,'BREAK POINT UNITS THAT BECOME DUMMY HEADS OR TAILS MUST'
      PRINT*,'ONLY HAVE TWO PORTS, AN INLET AND AN OUTLET PORT.'
C     -----------------------------------------------------------------
C     Read in new dummy head units.
C     -----------------------------------------------------------------
50    PRINT1,'NUMBER OF NEW DUMMY HEADS (UP TO 30)'
      READ(*,*,ERR=50)HD
      IF(HD.NE.0)THEN
60     PRINT3,'ENTER LIST OF UNITS SEPARATED BY COMMAS;'
       READ(*,*,ERR=60)(NEWHD(I),I=1,HD)
      ENDIF
C     -----------------------------------------------------------------
C     Read in new dummy tail units.
C     -----------------------------------------------------------------
70    PRINT1,'NUMBER OF NEW DUMMY TAILS (UP TO 30)'
      READ(*,*,ERR=70)TL
      IF(TL.NE.0)THEN
80     PRINT3,'ENTER LIST OF UNITS SEPARATED BY COMMAS;'
       READ(*,*,ERR=80)(NEWTL(I),I=1,TL)
      ENDIF
C     -----------------------------------------------------------------
C     Read in redundant units.
C     -----------------------------------------------------------------
90    PRINT1,'NUMBER OF REDUNDANT UNITS (UP TO 100)'
      READ(*,*,ERR=90)DUD
      IF(DUD.NE.0)THEN
100    PRINT3,'ENTER LIST OF UNITS SEPARATED BY COMMAS;'
       READ(*,*,ERR=100)(REDUN(I),I=1,DUD)
      ENDIF
C     -----------------------------------------------------------------
C     Write data to file DATFIL//'.BPT'.
C     -----------------------------------------------------------------
```

```fortran
      BRKFIL=DATFIL(1:LENGTH(DATFIL))//'.BPT'
      OPEN(UNIT=5,FILE='[.RESULTS]'//BRKFIL,STATUS='UNKNOWN',ERR=500)
      WRITE(5,4)HD,TL,DUD
      WRITE(5,4)(NEWHD(I),I=1,HD)
      WRITE(5,4)(NEWTL(I),I=1,TL)
      WRITE(5,4)(REDUN(I),I=1,DUD)
      CLOSE(5)
      PRINT5,'DATA WRITTEN TO FILE ',BRKFIL
      ENDIF
C     -------------------------------------------------------------------
C     Set the new library reference number of new dummy heads,
C     add connection number to the HEAD array and reset outlet
C     port number to 1.  Set the incoming connections end unit
C     number to zero and reset inlet port number to 2.
C     -------------------------------------------------------------------
      IF(HD.NE.0)THEN
       DO 150,I=1,HD
        TOPOL(NEWHD(I),1)=6
        NHEAD=NHEAD+1
        DO 150,J=1,NST
         IF(STREA(J,1).EQ.NEWHD(I))THEN
          HEAD(NHEAD)=J
          STREA(J,2)=1
         ENDIF
         IF(STREA(J,3).EQ.NEWHD(I))THEN
          STREA(J,3)=0
          STREA(J,4)=2
         ENDIF
150     CONTINUE
      ENDIF
C     -------------------------------------------------------------------
C     Set the new library reference number of new dummy tails and
C     add connection number to the TAIL array.  Set the outgoing
C     connections start unit number to zero.
C     -------------------------------------------------------------------
      IF(TL.NE.0)THEN
       DO 200,I=1,TL
        TOPOL(NEWTL(I),1)=7
        NTAIL=NTAIL+1
        DO 200,J=1,NST
         IF(STREA(J,3).EQ.NEWTL(I)) TAIL(NTAIL)=J
```

```fortran
          IF(STREA(J,1).EQ.NEWTL(I)) STREA(J,1)=0
200       CONTINUE
          ENDIF
C         --------------------------------------------------------------
C         Set the library reference number of redundant units to zero.
C         --------------------------------------------------------------
          IF(DUD.NE.0)THEN
          DO 250,I=1,DUD
250        TOPOL(REDUN(I),1)=0
          ENDIF
C         --------------------------------------------------------------
C         Reset the numbers of dummy heads and tails in the NDATA
C         array and return.
C         --------------------------------------------------------------
          NDATA(3)=NHEAD
          NDATA(4)=NTAIL

          RETURN
C         --------------------------------------------------------------
C         Error trapping section.
C         --------------------------------------------------------------
500       PRINT5,'CAN NOT OPEN FILE ',DATFIL
          GOTO 1000
550       PRINT5,'CAN NOT READ FIRST LINE IN FILE ',DATFIL
          GOTO 1000
600       PRINT5,'CAN NOT READ DUMMY HEADS IN FILE ',DATFIL
          GOTO 1000
650       PRINT5,'CAN NOT READ DUMMY TAILS IN FILE ',DATFIL
          GOTO 1000
700       PRINT5,'CAN NOT READ REDUNDANT UNITS IN FILE ',DATFIL

1000      CALL EXIT

          END
```

A-4

Appendix B

Fault And Unit Library

B.1 Fault And Event Library List

The list consists of sixty two rows of eight names and one row of four names. This is the format used in the program data file. There is a total of five hundred possible fault and event names. Names are limited to eight characters in length and a translations of most of the abbreviations can be found in Tables 5.3 and 5.4.

The numbering system for the fault and event index goes from left to right one row at a time. The rows are numbered to help identify the index of a particular fault or event.

The '--------' is used to indicate a position which is not in current use.

1 A(DUMMY)	B(DUMMY)	C(DUMMY)	D(DUMMY)	E(DUMMY)	F(DUMMY)	G(DUMMY)	H(DUMMY)
2 DUMMY1	DUMMY2	DUMMY3	DUMMY4	DUMMY5	DUMMY6	DUMMY7	DUMMY8
3 DUMMY9	DUMMY10	DUMMY11	DUMMY12	DUMMY13	DUMMY14	DUMMY15	DUMMY16
4 INTERNAL	EXTERNAL	ENABLING	BRANCH	LEAKAGE	BLOCKAGE	COMPLOOP	--------
5 HI-FLOW	LO-FLOW	NO-FLOW	HI-PRESS	LO-PRESS	NO-PRESS	--------	--------
6 SEQ-ABRT	SEQ-F-AF	SEQ-F-AT	NORMAL	IMPOSS	OTH-CAUS	--------	--------
7 CL-F-HA	CL-F-LA	CL-F-NA	CL-STK	TL-FN-F	TL-OR-F	--------	--------
8 LK-LP-EN	LK-HP-EN	PART-BLK	COMP-BLK	EXT-HEAT	EXT-COLD	INT-LK	--------
9 HV-D-SH	HV-F-SH	HV-FT-SH	HV-D-OP	HV-F-OP	HV-FT-OP	--------	--------
10 SV-D-SH	SV-F-SH	SV-FT-SH	SV-D-OP	SV-F-OP	SV-FT-OH	--------	--------
11 RCV-D-SH	RCV-F-SH	RCV-FT-S	RCV-D-OP	RCV-F-OP	RCV-FT-O	--------	--------
12 CV-D-SH	CV-F-SH	CV-FT-SH	CV-D-OP	CV-F-OP	CV-FT-OP	--------	--------
13 CV-F-HA	CV-F-LA	CV-F-NA	CV-STK	CV-F-VL	CV-F-NM	--------	--------
14 TV-D-SH	TV-F-SH	TV-FT-SH	TV-D-OP	TV-F-OP	TV-FT-OP	--------	--------
15 3WV-D-SH	3WV-F-SH	3WV-FT-S	3WV-D-OP	3WV-F-OP	3WV-FT-O	--------	--------
16 3WV-F-HA	3WV-F-LA	3WV-F-NA	3WV-STK	--------	--------	--------	--------
17 CTV-D-SH	CTV-F-SH	CTV-FT-S	CTV-D-OP	CTV-F-OP	CTV-FT-O	--------	--------
18 CTV-F-HA	CTV-F-LA	CTV-F-NA	CTV-STK	--------	--------	--------	--------
19 ESV-D-SH	ESV-F-SH	ESV-FT-S	ESV-D-OP	ESV-F-OP	ESV-FT-O	--------	--------
20 RV-D-SH	RV-F-SH	RV-FT-SH	RV-D-OP	RV-F-OP	RV-FT-OP	RV-UNDSZ	--------
21 VV-D-SH	VV-F-SH	VV-FT-SH	VV-D-OP	VV-F-OP	VV-FT-OP	--------	--------
22 NRV-F-SH	NRV-FT-S	NRV-F-OP	NRV-FT-O	--------	--------	--------	--------
23 SMP-FT-S	SMP-D-OP	SMP-F-OP	--------	--------	--------	--------	--------
24 GOV-D-SH	GOV-F-SH	GOV-FT-S	GOV-D-OP	GOV-F-OP	GOV-FT-O	--------	--------
25 PRV-D-SH	PRV-F-SH	PRV-FT-S	PRV-D-OP	PRV-F-OP	PRV-FT-O	--------	--------
26 BD-FT-OP	BD-FT-RP	BD-UNDSZ	BD-RUPT	--------	--------	--------	--------

#							
27	BLK-FAIL	BLK-REM	-------	-------	-------	-------	-------	-------
28	CNT-F-HI	CNT-F-LO	CNT-STK	CNT-MAN	CNT-F-VL	CNT-F-NM	--------	-------
29	TSW-F-ON	TSW-F-OF	TSW-STK	TSW-DIS	ACT-F-ON	ACT-F-OF	ACT-STK	-------
30	OP-FN-F	OP-OR-F	OP-ERROR	OP-OVLD	OP-ABSNT	--------	--------	-------
31	COM-FN-F	COM-OR-F	COMP-ERR	COM-OVLD	-------	-------	--------	-------
32	SEL-PC	SEL-PCO	SEL-PCH	SEL-PLVC	SEL-PCL	SEL-VC	SEL-BROK	--------
33	SW-AT-P3	SW-BROKN	-------	--------	--------	-------	-------	-------
34	REL-F-ON	REL-F-OF	REL-STK	--------	--------	-------	-------	-------
35	SEN-F-HI	SEN-F-LO	SEN-STK	--------	--------	-------	-------	-------
36	ALM-F-ON	ALM-F-OF	ALM-DIS	ALM-FAIL	AL-BROKN	--------	--------	--------
37	SIG-PB	SIG-CB	-------	--------	--------	-------	-------	-------
38	SET-P-HI	SET-P-LO	-------	--------	--------	-------	-------	-------
39	UTILLOSS	POW-LOSS	STM-LOSS	IAR-LOSS	-------	-------	--------	-------
40	PAR-LOSS	ING-LOSS	NIT-LOSS	CWT-LOSS	PWT-LOSS	VAC-LOSS	-------	--------
41	FWT-LOSS	CO2-LOSS	FOAMLOSS	FFS-FAIL	SPRINK-F	--------	-------	-------
42	UTIL-OK	AIR-OK	POWER-OK	PRESS-OK	CL-IS-OK	--------	--------	-------
43	POWER-HI	POWER-LO	POWER-ON	POWER-OF	-------	--------	-------	-------
44	SHUTDOWN	STARTUP	FT-ST-UP	--------	-------	--------	-------	--------
45	IMPLR-F	FAN-BRKN	AIR-LOCK	CAVITATN	PUMP-SUR	RACING	W-HAMMER	------
46	FOULING	VAP-BLKT	FLOODING	FROTHING	CAT-DETN	POOR-MIX	INT-EXPL	RAP-RCTN
47	RAP-FLSH	HEATFAIL	I-HEAT-F	I-COOL-F	OV-HT-TK	H2O-PRES	--------	--------
48	CORROSN	FATIGUE	BRITFRAC	VIBRATN	LOW-TEMP	FLASHING	--------	-------
49	FLAM-MIX	FIRE-IGN	ELEC-SPK	EXT-FIRE	MAJ-FIRE	--------	--------	-------
50	SUP-FAIL	OV-ST-SP	OV-LOAD	OVR-FILL	CRIT-STR	OVERPRES	--------	-------
51	DESN-ERR	INSP-ERR	MECH-DEF	--------	--------	--------	-------	-------
52	E-QUAKE	SUBSDNCE	L-SLIDE	VHI-WIND	EXT-IMP	AIR-IMP	VEH-IMP	EXT-EXPL
53	ISOLATN	SYS-ISOL	-------	--------	--------	-------	-------	-------
54	NO-TORCH	NO-SPARK	NO-PILOT	NO-BURNR	NORELIEF	--------	--------	--------
55	INSUF-PU	PR-PURGE	P-FL-STA	P-X-LIT	T-FL-LT	PR-FL-ST	PR-X-LT	TOR-NLIT
56	T-N-INS	THV-B-IN	THVB-OUT	PIL-NLIT	PIL-LIT	--------	-------	-------
57	-------	-------	-------	-------	-------	-------	-------	-------
58	-------	--------	--------	--------	--------	-------	--------	--------
59	--------	-------	-------	-------	--------	-------	--------	--------
60	-------	-------	-------	-------	-------	-------	-------	-------
61	-------	-------	-------	--------	--------	-------	--------	-------
62	-------	-------	-------	-------	--------	-------	-------	-------
63	-------	-------	-------	-------				

B.2 Unit Model Library List

This is a list of unit names that should make up the unit model library. It is not a comprehensive list but has been compiled to give as many types of unit that may be found in a process plant.

The models with a '#' are those which are currently contained in the library.

1		PROCESS AIR SUPPLY
2		PROCESS STEAM SUPPLY
3		PROCESS WATER SUPPLY
4	#	PROCESS NITROGEN SUPPLY
5		COOLING WATER SUPPLY
6		FIRE WATER SUPPLY
7		FUEL GAS SUPPLY
8		INERT GAS SUPPLY
10	#	INSTRUMENT AIR SUPPLY
11	#	ELECTRICAL POWER SUPPLY
12	#	STEAM POWER SUPPLY
15	#	STEAM TURBINE FOR POWER SUPPLY
20	#	DRAIN
21	#	VENT STACK
22	#	FLARE RELIEF STACK
25	#	VACUUM UNIT
30	#	DUMMY HEAD
31	#	DUMMY TAIL
32	#	SHORT LENGTH OF PIPE
33	#	LONG LENGTH OF PIPE
34	#	DIVIDER (2 OUTPUTS)
35	#	HEADER (2 INPUTS)
40	#	BINARY MIXER (2 INPUTS)

50	#	CONTROL VALVE (FAILS OPEN)
51	#	CONTROL VALVE (FAILS CLOSED)
52		3-WAY CONTROL VALVE (FAILS OPEN)
53	#	3-WAY CONTROL VALVE (FAILS CLOSED)
60	#	TRIP VALVE (OPEN, FAILS OPEN)
61	#	TRIP VALVE (OPEN, FAILS CLOSED)
62		TRIP VALVE (CLOSED, FAILS OPEN)
63	#	TRIP VALVE (CLOSED, FAILS CLOSED)
70		EMERGENCY SHUTDOWN VALVE (OPEN, FAILS CLOSED)
71		EMERGENCY SHUTDOWN VALVE (CLOSED, FAILS OPEN)
72	#	VENT VALVE (OPEN)
73	#	VENT VALVE (CLOSED)
74		COMBINED CONTROL/TRIP VALVE (OPERATING, FAILS OPEN)
75		COMBINED CONTROL/TRIP VALVE (OPEN)
76	#	COMBINED CONTROL/TRIP VALVE (OPERATING, FAILS CLOSED)
77	#	COMBINED CONTROL/TRIP VALVE (CLOSED)
78	#	MANUAL VENT VALVE (OPEN)
79		MANUAL VENT VALVE (CLOSED)
80	#	PRESSURE RELIEF VALVE
81	#	HAND VALVE (OPEN)
82	#	HAND VALVE (CLOSED)
83	#	REMOTELY CONTROLLED VALVE (OPEN)
84	#	REMOTELY CONTROLLED VALVE (CLOSED)
85	#	SOLENOID VALVE (OPEN)
86		SOLENOID VALVE (CLOSED)
87	#	PRESSURE REDUCING VALVE
88	#	GOVERNOR VALVE
89		EXCESS FLOW VALVE
90	#	NON-RETURN VALVE
100	#	LOCAL SENSOR
101	#	SAMPLE POINT
102	#	FLUSHING LINK
103	#	DRAIN POINT
104		IN-LINE FILTER
105		STEAM TRAP

106 # BLANK FLANGE

107 # BURSTING DISC

108 # ORIFICE PLATE

115 # MULTI-PURPOSE PIPE MOUNTED SENSOR

116 # MULTI-PURPOSE DEAD LEG SENSOR

117 # MULTI-PURPOSE VESSEL MOUNTED SENSOR

118   POTENTIOMETER ON VALVE STEM

119 # DIFFERENTIAL PRESSURE TRANSDUCER

125 # ELECTRIC CABLE

126   INSTRUMENT AIR PIPE

127 # I/P TRANSDUCER

128   P/I TRANSDUCER

129 # TRIP ACTIVATOR

130 # ELECTRICAL RELAY

131 # SIGNAL SPLITTER (2 OUTPUTS)

132 # VESSEL PORT SPLITTER (2 OUTPUTS)

133 # SIGNAL HEADER (2 INPUTS)

134 # HIGH SIGNAL SELECTOR (2 INPUTS)

135 # LOW SIGNAL SELECTOR (2 INPUTS)

136 # SET POINT UNIT

137 # PROCESS OPERATOR

138   PROCESS CONTROL COMPUTER

139 # HIGH ALARM

140 # LOW ALARM

141   VARIABLE RESISTOR

145 # PRESSURE CONTROL AND OVER-RIDE CONTROL SELECTOR

150 # CONTROLLER (DIRECT ACTING)

151 # CONTROLLER (REVERSE ACTING)

152 # CASCADE CONTROLLER (DIRECT/DIRECT)

153 # CASCADE CONTROLLER (DIRECT/REVERSE)

154   CASCADE CONTROLLER (REVERSE/DIRECT)

155 # CASCADE CONTROLLER (REVERSE/REVERSE)

156 # DOUBLE OUTPUT CONTROLLER (DIRECT/REVERSE)

157 # CONTROLLER ON MANUAL

165 # TRIP SWITCH (1/1 HIGH)
166 # TRIP SWITCH (1/1 LOW)
167 # TRIP SWITCH (1/1 NONE)
168   TRIP SWITCH (1/2 HIGH)
169 # TRIP SWITCH (1/2 LOW)
170   TRIP SWITCH (1/2 NONE)
171   TRIP SWITCH (1/3 HIGH)
172   TRIP SWITCH (1/3 LOW)
173   TRIP SWITCH (1/3 NONE)
174 # TRIP SWITCH (2/3 HIGH)
175   TRIP SWITCH (2/3 LOW)
176   TRIP SWITCH (2/3 NONE)

190   FLOW SWITCH
191 # PRESSURE SWITCH
192   TEMPERATURE SWITCH
193   LEVEL SWITCH
194   LIMIT SWITCH FOR VALVE

200   RECIPROCATING COMPRESSOR (RUNNING)
201   RECIPROCATING COMPRESSOR (STOPPED)
202   CENTRIFUGAL COMPRESSOR (RUNNING)
203   CENTRIFUGAL COMPRESSOR (STOPPED)
204   RECIPROCATING PUMP (RUNNING)
205   RECIPROCATING PUMP (STOPPED)
206 # CENTRIFUGAL PUMP (RUNNING)
207 # CENTRIFUGAL PUMP (STOPPED)
208 # CENTRIFUGAL PUMP WITH WATER SEAL (RUNNING)
209 # CENTRIFUGAL PUMP WITH WATER SEAL (STOPPED)

220 # HEAT EXCHANGER (S&T/FT, HEATING SHELL/AIR SIDE)
221 # HEAT EXCHANGER (S&T, HEATING TUBE SIDE)
222   HEAT EXCHANGER (F&P/DP, HEATING FRAME/OUT- SIDE)
223   HEAT EXCHANGER (F&P/DP, HEATING PLATE/IN- SIDE)
224 # HEAT EXCHANGER (PARTIAL CONDENSER, S&T, COOLING SHELL SIDE)
225 # HEAT EXCHANGER (CONDENSER, S&T, COOLING SHELL SIDE)
226   HEAT EXCHANGER (CONDENSER, S&T/FT, COOLING TUBE SIDE)
227   HEAT EXCHANGER (CONDENSER, F&P/DP, COOLING FRAME/OUT-
        SIDE)
228 # HEAT EXCHANGER (CONDENSER, F&P/DP, COOLING PLATE/IN- SIDE)
229   HEAT EXCHANGER (PARTIAL REBOILER, S&T, HEATING SHELL SIDE)

230		HEAT EXCHANGER (REBOILER, S&T, HEATING SHELL SIDE)
231	#	HEAT EXCHANGER (REBOILER, S&T, HEATING TUBE SIDE)
232		HEAT EXCHANGER (REBOILER, F&P/DP, HEATING FRAME/OUT- SIDE)
233		HEAT EXCHANGER (REBOILER, F&P/DP, HEATING PLATE/IN- SIDE)
235	#	HEAT EXCHANGER (CONDENSER, RELIEF, S&T, 7 PORTS)
245		PIPE HEATER
246		BOILER
247		VAPORISER
248		FURNACE
249		BURNER
250	#	ELECTRICAL VAPORISER
251	#	BGAS BUTANE VAPORISER
252	#	BGAS LNG VAPORISER
253	#	BGAS LNG VAPORISER (NO FUEL GAS LIMITATION)
270	#	TEMPLATE VESSEL MODEL (OPEN, LIQUID)
271	#	TEMPLATE VESSEL MODEL (CLOSED, LIQUID)
272	#	TEMPLATE VESSEL MODEL (PRESSURISED, GAS)
273	#	TEMPLATE VESSEL MODEL (VACUUM, GAS)
274	#	TEMPLATE VESSEL MODEL (PRESSURISED, GAS AND LIQUID)
280		STORAGE VESSEL (PRESSURISED, GAS)
281		STORAGE VESSEL (FLOATING ROOF, GAS)
282		STORAGE VESSEL (OPEN, LIQUID)
283		STORAGE VESSEL (OPEN, FIXED ROOF, LIQUID)
284		STORAGE VESSEL (OPEN, FLOATING ROOF, LIQUID)
285		STORAGE VESSEL (PRESSURISED, LIQUID)
290	#	VESSEL MODEL FOR OLEFIN DIMERISATION PLANT
291	#	VESSEL MODEL FOR LAWLEY PIPELINE
292	#	AMMONIA SPHERE
293	#	AKZO CHLORINE STORAGE VESSEL
310		STORAGE SILO FOR SOLIDS
311		CONTAINMENT HOPPER FOR SOLIDS
312		POWER CYLINDER FOR SOLIDS
313		SCREW FEEDER AND MOTOR

314   FEEDER BELT AND MOTOR FOR SOLIDS

315   CONVEYOR BELT FOR SOLIDS

316   GRINDING MILL

320   REACTION VESSEL (TUBULAR, GAS)

321   REACTION VESSEL (TUBULAR, LIQUID)

322   REACTION VESSEL (CONTINUOUS STIRRED TANK)

323   REACTION VESSEL (PACKED BED, GAS)

324   REACTION VESSEL (PACKED BED, LIQUID)

325   REACTION VESSEL (FLUIDISED BED, GAS)

326   REACTION VESSEL (FLUIDISED BED, LIQUID)

327   REACTION VESSEL (CATALYTIC GAUZE, GAS)

328   REACTION VESSEL (SEMI-BATCH, FILLING)

329   REACTION VESSEL (SEMI-BATCH, REACTING)

330   REACTION VESSEL (SEMI-BATCH, EMPTYING)

331   REACTION VESSEL (BATCH, FILLING)

332   REACTION VESSEL (BATCH, REACTING)

333   REACTION VESSEL (BATCH, EMPTYING)

340 # ETHENE-OXIDE REACTION VESSEL (SHELL AND TUBE)

350   ABSORPTION COLUMN (PACKED)

351   ABSORPTION COLUMN (PLATE)

352   ADSORPTION COLUMN

353   WET SCRUBBER (GAS-LIQUID)

354   WET SCRUBBER (GAS-SOLID)

355   STEAM STRIPPER (LIQUID-GAS)

356   STEAM STRIPPER (LIQUID-LIQUID)

357   DISTILLATION COLUMN (SUPER-HEATED VAPOUR FEED)

358 # DISTILLATION COLUMN (SATURATED VAPOUR FEED)

359   DISTILLATION COLUMN (BOILING LIQUID FEED)

360   DISTILLATION COLUMN (SUB-COOLED LIQUID FEED)

361   RECTIFYING COLUMN

362   STRIPPING COLUMN

363   DECANTING VESSEL

364   ION EXCHANGER

365   FLASH DRUM

366   HYDROCYCLONE

367   CYCLONE

368   FILTER (GAS-SOLID)

369   FILTER (LIQUID-SOLID)

370		ELECTROSTATIC PRECIPITATOR
371		THICKENER/CLARIFIER
372		CENTRIFUGE
373		ROTARY DRIER
374		EVAPORATOR (CONCENTRATING)
375		EVAPORATOR (CRYSTALISING)
400	#	BVD2 E1 COLUMN SECTION
410	#	BGAS BURNER (SHUTDOWN)
411	#	BGAS BURNER (START UP)
412	#	BGAS BURNER (RUN CONDITION)
413	#	BGAS TORCH VALVE (CLOSED)
414	#	BGAS TORCH VALVE (LIT)
415	#	BGAS PILOT BURNER (LIGHTING MAIN BURNER)
416	#	BGAS PILOT BURNER (RUN CONDITION)
417	#	BGAS FLAME DETECTOR
418	#	BGAS FAN
419	#	BGAS FLAME DETECTOR (AUTO)
420	#	BGAS AUTO PILOT BURNER (START UP)
421	#	BGAS AUTO PILOT BURNER (RUNNING)
422	#	BGAS AUTO MAIN BURNER (BEING LIT)
423	#	BGAS AUTO MAIN BURNER (RUNNING)

Appendix C

Vessel Model Templates

C.1 Open Vessel Model Template

This is a liquid storage vessel which is open to the atmosphere. All the port openings are assumed to be below the liquid surface because this is the normal design procedure for preventing electrostatic build up. The first four ports are inlet ports, the second four are outlet ports and the ninth port is the vessel port.

Port 1 normally has flow and is connected to a running pump with no pressure effects. The pump is defined as a reciprocating pump. Reverse flow is allowed because a leak in the piping from the pump will cause liquid to run out of the vessel through this port. Port 2 is defined as connected to a running, high pressure centrifugal pump which is not affected by the vessel pressure. Ports 3 and 4 are the same as ports 1 and 2, respectively, except that the pumps for these two are not running and a closed valve is connected to it so there is no flow.

The port 5 outlet is not affected by the vessel pressure and the pump is defined as a reciprocating pump. Since the vessel is open to the atmosphere a leak in the pipe-work for this port will result in liquid running out of the vessel in the normal direction of flow. For reverse flow to occur liquid must pass through the pump in the wrong direction. In the assumptions it has been noted that reverse flow through a reciprocating pump is impossible, therefore, this port cannot sustain reverse flow. Port 6 is not connected to a pump and flow occurs under gravity so is affected by the vessel pressure. Port 7 is the same as port 5 except the pump is not running and a closed valve is connected so there is no flow. Port 8 is the same as port 6 except a closed valve prevents liquid flowing out of it.

The entire model is shown on the following pages.

1)   MODEL   NUMBER   NAME
            90   TEMPLATE VESSEL MODEL (LIQUID, OPEN)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 27
NO. OF PROPAGATION EQUATIONS: 36
NO. OF EVENT STATEMENTS: 26
NO. OF DECISION TABLES: 11
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

USE THIS MODEL TO DEVELOP YOUR OWN MODELS OR USE IT AS IT IS.

PORTS CAN BE REMOVED IF SO DESIRED BY FOLLOWING THIS EXAMPLE.

IF PORT 2 IS NOT REQUIRED, DELETE PROPAGATION EQUATIONS 5 TO 8. DELETE THE G2IN, T2IN AND X2IN TERMS FROM EQUATIONS 33, 35 AND 36 RESPECTIVELY. DELETE EVENT STATEMENTS 3 AND 4. DELETE THE G2IN NONE, R2IN NONE, G2IN REV AND R2IN REV TERMS FROM EVENT STATEMENTS 16 AND 17. DELETE THE G2IN SOME AND R2IN SOME TERMS FROM EVENT STATEMENT 23. WHEN THE G2IN NONE TERM IS REMOVED FROM THE FIRST DECISION TABLE THIS LEAVES ONE ELEMENT SO DELETE THE DECISION TABLE AND ADD V G1IN NONE: C(DUMMY) TO THE LIST OF EVENT STATEMENTS.

ALTER THE NUMBERS FOR THE NUMBER OF PROPAGATION EQUATIONS, EVENT STATEMENTS AND DECISION TABLES AS NECESSARY AT THE TOP OF THE MODEL.

HAVING CHANGED THE MODEL IT MUST BE GIVEN A NEW FILE NAME AND CORRESPONDING MODEL NUMBER.

IT IS NOT NECESSARY TO CHANGE THE PORT NUMBERS BUT ENSURE THE CORRECT PORT NUMBERS ARE GIVEN IN THE MASTER PROGRAM.

THIS IS AN OPEN VESSEL USED TO STORE LIQUID, ALL PORTS ARE BELOW THE LEVEL OF THE LIQUID SURFACE.

PORT 1: NORMAL FLOW INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED

PORT 2: NORMAL FLOW INLET FROM HIGH PRESSURE CENTRIFUGAL PUMP

PORT 3: NO FLOW INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED

PORT 4: NO FLOW INLET FROM HIGH PRESSURE CENTRIFUGAL PUMP

PORT 5: NORMAL FLOW OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED

PORT 6: NORMAL FLOW OUTLET UNDER GRAVITY

PORT 7: NO FLOW OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED

PORT 8: NO FLOW OUTLET UNDER GRAVITY.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN)

R1IN= F (-P9VES)

U1IN= F (T9VES)

Y1IN= F (X9VES)

G2IN= F (Q2IN)

R2IN= F (-P9VES)

U2IN= F (T9VES)
Y2IN= F (X9VES)
G3IN= F (Q3IN)
R3IN= F (-P9VES)
U3IN= F (T9VES)
Y3IN= F (X9VES)
G4IN= F (Q4IN)
R4IN= F (-P9VES)
U4IN= F (T9VES)
Y4IN= F (X9VES)
Q5OUT= F (G5OUT)
P5OUT= F (P9VES)
T5OUT= F (T9VES)
X5OUT= F (X9VES)
Q6OUT= F (G6OUT, P9VES)
P6OUT= F (P9VES)
T6OUT= F (T9VES)
X6OUT= F (X9VES)
Q7OUT= F (G7OUT)
P7OUT= F (P9VES)
T7OUT= F (T9VES)
X7OUT= F (X9VES)
Q8OUT= F (G8OUT, P9VES)
P8OUT= F (P9VES)
T8OUT= F (T9VES)
X8OUT= F (X9VES)
L9VES= F (G1IN, G2IN, -Q5OUT, -Q6OUT)
P9VES= F (L9VES)
T9VES= F (T1IN, T2IN, T3IN, T4IN)
X9VES= F (X1IN, X2IN, X3IN, X4IN)

4)   EVENT STATEMENTS

V G1IN NONE: L9VES LO
V G1IN REV: L9VES LO, L9VES NONE
V G2IN NONE: L9VES LO
V G2IN REV: L9VES LO, L9VES NONE
V G3IN SOME: L9VES HI
V G3IN REV: L9VES LO, L9VES NONE
V G4IN SOME: L9VES HI
V G4IN REV: L9VES LO, L9VES NONE
I C(DUMMY): L9VES NONE
V Q5OUT NONE: L9VES HI
V Q6OUT NONE: L9VES HI
V Q6OUT REV: L9VES HI
V Q7OUT SOME: L9VES LO
V Q8OUT SOME: L9VES LO

V Q8OUT REV: L9VES HI
V P9VES HI: G1IN NONE, R1IN NONE, G2IN NONE, R2IN NONE, G1IN REV, R1IN REV
V P9VES HI: G2IN REV, R2IN REV, G3IN REV, R3IN REV, G4IN REV, R4IN REV
V P9VES NONE: Q6OUT REV, P6OUT REV, Q8OUT REV, P8OUT REV
V L9VES NONE: P9VES LO
F LK-LP-EN: L9VES LO, L9VES NONE
F EXT-HEAT: T9VES HI
F EXT-COLD: T9VES LO
S NORMAL: G1IN SOME, R1IN SOME, G2IN SOME, R2IN SOME, G3IN NONE, R3IN NONE
S NORMAL: G4IN NONE, R4IN NONE, Q5OUT SOME, P5OUT SOME, Q6OUT SOME, P6OUT SOME
S NORMAL: Q7OUT NONE, P7OUT NONE, Q8OUT NONE, P8OUT NONE
S IMPOSS: Q5OUT REV, P5OUT REV, Q7OUT REV, P7OUT REV

5)   DECISION TABLES

V G1IN NONE V G2IN NONE T C(DUMMY)
I C(DUMMY) V L9VES NONE T Q5OUT NONE, P5OUT NONE
I C(DUMMY) V L9VES NONE T Q6OUT NONE, P6OUT NONE
V Q6OUT REV V U6OUT HI T T9VES HI
V Q6OUT REV V U6OUT LO T T9VES LO
V Q6OUT REV V Y6OUT HI T X9VES HI
V Q6OUT REV V Y6OUT LO T X9VES LO
V Q8OUT REV V U8OUT HI T T9VES HI
V Q8OUT REV V U8OUT LO T T9VES LO
V Q8OUT REV V Y8OUT HI T X9VES HI
V Q8OUT REV V Y8OUT LO T X9VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 5 AND 6.

NO MULTI-COMPONENT FEATURES

C.2 Closed Liquid Vessel Model Template

This is a closed vessel used solely to store liquid. Although there is a gas space above the liquid surface containing vapour the quantity of vapour cannot be changed by using the inlet or outlet ports. The vapour is compressed as the level of liquid rises. The vessel temperature will also have an effect on the vapour so this has to be incorporated into the vessel pressure term. The vessel pressure is below atmospheric pressure which allows a demonstration of the effects of no reverse flow on the inlet ports. Each of the four inlet and four outlet ports are assumed to be below the surface of the liquid.

Port 1 is an inlet connected to a running pump operating like a reciprocating pump. Since the vessel is below atmospheric pressure a leak in the connection will cause a flow of air into the vessel, for this reason it is assumed impossible for flow to occur in the reverse direction. This port is not affected by the vessel pressure. Port 2 is an inlet with flow under gravity caused by a unit at a higher elevation or pressure, it is affected by the vessel pressure. Port 3 is the same as port 1 but the pump is stopped and a closed valve prevents flow. Port 4 is the same as port 2 but a closed valve prevents flow.

The port 5 outlet is connected to a running pump similar to a reciprocating pump. The port allows reverse flow because the higher, external pressure will cause air to flow in through the port. The port is not affected by the vessel pressure. Port 6 is connected to a running pump so the vessel pressure does not affect it. The pump on this port has been defined as a centrifugal pump. Port 7 and port 8 are the same as port 5 and port 6, respectively, except that the pumps are not running and a closed valve prevents flow.

It should be noted that in this model reverse flow through ports 5 and 7 has been defined as causing high pressure *and* level. However, this reverse flow is likely to be air and will only affect the pressure. This event statement is the result of the systematic approach used in the process and the level term should be deleted or ignored in this case. A similar result is obtained for the leak from a high pressure environment, LK-HP-EN fault. In this case the concentration term appears but it is quite possible for the air to dissolve in the liquid and thus cause an impurity in the liquid so the term is in fact valid.

The entire model is shown on the following pages.

1)   MODEL   NUMBER   NAME
          91   TEMPLATE VESSEL MODEL (LIQUID, CLOSED)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 24
NO. OF PROPAGATION EQUATIONS: 32
NO. OF EVENT STATEMENTS: 28
NO. OF DECISION TABLES: 19
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

USE THIS MODEL TO DEVELOP YOUR OWN MODELS OR USE IT AS IT IS.
PORTS CAN BE REMOVED IF SO DESIRED BY FOLLOWING THIS EXAMPLE.
IF PORT 3 IS NOT REQUIRED, DELETE PROPAGATION EQUATIONS 9 TO 12. DELETE THE T3IN
AND X3IN TERMS FROM EQUATIONS 35 AND 36 RESPECTIVELY. DELETE EVENT STATEMENT
4. DELETE THE G3IN NONE, R3IN NONE, G3IN REV AND R3IN REV TERMS FROM EVENT
STATEMENTS 25 AND 28.
ALTER THE NUMBERS FOR THE NUMBER OF PROPAGATION EQUATIONS, EVENT STATEMENTS
AND DECISION TABLES AS NECESSARY AT THE TOP OF THE MODEL.
HAVING CHANGED THE MODEL IT MUST BE GIVEN A NEW FILE NAME AND CORRESPONDING
MODEL NUMBER.

IT IS NOT NECESSARY TO CHANGE THE PORT NUMBERS BUT ENSURE THE CORRECT PORT
NUMBERS ARE GIVEN IN THE MASTER PROGRAM.

THIS IS A CLOSED VESSEL USED TO STORE LIQUID, ALL PORTS ARE BELOW THE LEVEL OF THE
LIQUID SURFACE AND THE VESSEL PRESSURE IS BELOW ATMOSPHERIC.
PORT 1: NORMAL FLOW INLET FROM RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 2: NORMAL FLOW INLET UNDER GRAVITY
PORT 3: NO FLOW INLET FROM RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 4: NO FLOW INLET UNDER GRAVITY
PORT 5: NORMAL FLOW OUTLET TO RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 6: NORMAL FLOW OUTLET TO CENTRIFUGAL PUMP
PORT 7: NO FLOW OUTLET TO RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 8: NO FLOW OUTLET TO CENTRIFUGAL PUMP.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN)
R1IN= F (-P9VES)
G2IN= F (Q2IN, -P9VES)
R2IN= F (-P9VES)
U2IN= F (T9VES)
Y2IN= F (X9VES)
G3IN= F (Q3IN)
R3IN= F (-P9VES)
G4IN= F (Q4IN, -P9VES)

R4IN= F (-P9VES)
U4IN= F (T9VES)
Y4IN= F (X9VES)
Q5OUT= F (G5OUT)
P5OUT= F (P9VES)
T5OUT= F (T9VES)
X5OUT= F (X9VES)
Q6OUT= F (G6OUT)
P6OUT= F (P9VES)
T6OUT= F (T9VES)
X6OUT= F (X9VES)
Q7OUT= F (G7OUT)
P7OUT= F (P9VES)
T7OUT= F (T9VES)
X7OUT= F (X9VES)
Q8OUT= F (G8OUT)
P8OUT= F (P9VES)
T8OUT= F (T9VES)
X8OUT= F (X9VES)
L9VES= F (G1IN, G2IN, -Q5OUT, -Q6OUT)
P9VES= F (L9VES, T9VES)
T9VES= F (T1IN, T2IN, T3IN, T4IN)
X9VES= F (X1IN, X2IN, X3IN, X4IN)

4)   EVENT STATEMENTS

V G1IN NONE: L9VES LO, P9VES LO
V G2IN NONE: L9VES LO, P9VES LO
V G2IN REV: L9VES LO, L9VES NONE, P9VES LO
V G3IN SOME: L9VES HI, P9VES HI
V G4IN SOME: L9VES HI, P9VES HI
V G4IN REV: L9VES LO, L9VES NONE, P9VES LO
I C(DUMMY): L9VES NONE
V Q5OUT NONE: L9VES HI, P9VES HI
V Q5OUT REV: L9VES HI, P9VES HI
V Q6OUT NONE: L9VES HI, P9VES HI
V Q6OUT REV: L9VES HI, P9VES HI
V Q7OUT SOME: L9VES LO, P9VES LO
V Q7OUT REV: L9VES HI, P9VES HI
V Q8OUT SOME: L9VES LO, P9VES LO
V Q8OUT REV: L9VES HI, P9VES HI
V P9VES HI: G1IN NONE, R1IN NONE, G2IN NONE, R2IN NONE
V P9VES HI: G2IN REV, R2IN REV, G4IN REV, R4IN REV
V P9VES NONE: Q5OUT REV, P5OUT REV, Q6OUT REV, P6OUT REV
V P9VES NONE: Q7OUT REV, P7OUT REV, Q8OUT REV, P8OUT REV
V L9VES NONE: P9VES LO
F LK-LP-EN: L9VES LO, L9VES NONE, P9VES LO

F LK-HP-EN: P9VES HI, P9VES NOR, X9VES HI
F EXT-HEAT: P9VES HI, T9VES HI
F EXT-COLD: P9VES LO, T9VES LO
S NORMAL: G1IN SOME, R1IN SOME, G2IN SOME, R2IN SOME, G3IN NONE, R3IN NONE
S NORMAL: G4IN NONE, R4IN NONE, Q5OUT SOME, P5OUT SOME, Q6OUT SOME, P6OUT SOME
S NORMAL: Q7OUT NONE, P7OUT NONE, Q8OUT NONE, P8OUT NONE
S IMPOSS: G1IN REV, R1IN REV, G3IN REV, R3IN REV

5)   DECISION TABLES

V G1IN NONE V G2IN NONE T C(DUMMY)
I C(DUMMY) V L9VES NONE T Q5OUT NONE, P5OUT NONE
I C(DUMMY) V L9VES NONE T Q6OUT NONE, P6OUT NONE
V Q5OUT REV V U5OUT HI T T9VES HI
V Q5OUT REV V U5OUT LO T T9VES LO
V Q5OUT REV V Y5OUT HI T X9VES HI
V Q5OUT REV V Y5OUT LO T X9VES LO
V Q6OUT REV V U6OUT HI T T9VES HI
V Q6OUT REV V U6OUT LO T T9VES LO
V Q6OUT REV V Y6OUT HI T X9VES HI
V Q6OUT REV V Y6OUT LO T X9VES LO
V Q7OUT REV V U7OUT HI T T9VES HI
V Q7OUT REV V U7OUT LO T T9VES LO
V Q7OUT REV V Y7OUT HI T X9VES HI
V Q7OUT REV V Y7OUT LO T X9VES LO
V Q8OUT REV V U8OUT HI T T9VES HI
V Q8OUT REV V U8OUT LO T T9VES LO
V Q8OUT REV V Y8OUT HI T X9VES HI
V Q8OUT REV V Y8OUT LO T X9VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 5 AND 6.

NO MULTI-COMPONENT FEATURES

C.3 Pressurised Gas Vessel Model Template

This is a closed vessel used for storing gas above atmospheric pressure. If a leak occurs in the pipe-work to a port connection then the gas will leak out. Therefore it is possible for an inlet port to have reverse flow even if it is connected to a reciprocating type pump. There are four inlet ports and four outlet ports

Port 1 is an inlet which normally has flow and is connected to a running pump (or compressor) where pressure effects are important. The pump is defined as a centrifugal pump. Port 2 is also connected to a running pump but there are no pressure effects. In this case the pump is defined as a reciprocating pump. As mentioned before it is possible for this port to have reverse flow. Ports 3 and 4 are the same as port 1 and 2, respectively, except that the pump is not running and a closed valve prevents flow.

Port 5 is an outlet connected to a running pump with no pressure effects. The pump is defined as a reciprocating pump but on this port reverse flow is not allowed. Port 6 is not connected to a pump but flow occurs due to the pressure inside the vessel. The port is defined as having flow under back-pressure. Ports 7 and 8 are the same as ports 5 and 6, respectively, but no flow occurs due to a closed valve.

Note that in this model the LK-HP-EN event statement has been removed because the vessel is above atmospheric pressure.

The entire model is shown in the following pages.

1)   MODEL   NUMBER   NAME
                 89   TEMPLATE VESSEL MODEL (GAS, CLOSED)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 23
NO. OF PROPAGATION EQUATIONS: 35
NO. OF EVENT STATEMENTS: 24
NO. OF DECISION TABLES: 8
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

USE THIS MODEL TO DEVELOP YOUR OWN MODELS OR USE IT AS IT IS.
PORTS CAN BE REMOVED IF SO DESIRED BY FOLLOWING THIS EXAMPLE.
IF PORT 3 IS NOT REQUIRED, DELETE PROPAGATION EQUATIONS 9 TO 12. DELETE THE T3IN
AND X3IN TERMS FROM PROPAGATION EQUATIONS 34 AND 35. DELETE EVENT STATEMENTS
5 AND 6. DELETE THE G3IN REV AND R3IN REV TERMS FROM EVENT STATEMENT 16. DELETE
THE G3IN NONE AND R3IN NONE TERMS FROM EVENT STATEMENT 21.
ALTER THE NUMBERS FOR THE NUMBER OF PROPAGATION EQUATIONS, EVENT STATEMENTS
AND DECISION TABLES AS NECESSARY AT THE TOP OF THE MODEL.
HAVING CHANGED THE MODEL IT MUST BE GIVEN A NEW FILE NAME AND CORRESPONDING
MODEL NUMBER.

IT IS NOT NECESSARY TO CHANGE THE PORT NUMBERS BUT ENSURE THE CORRECT PORT
NUMBERS ARE GIVEN IN THE MASTER PROGRAM.

THIS IS A CLOSED VESSEL USED TO STORE GASES ABOVE ATMOSPHERIC PRESSURE.
PORT 1: NORMAL FLOW INLET FROM CENTRIFUGAL PUMP
PORT 2: NORMAL FLOW INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 3: NO FLOW INLET FROM CENTRIFUGAL PUMP
PORT 4: NO FLOW INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 5: NORMAL FLOW OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 6: NORMAL FLOW OUTLET UNDER BACK PRESSURE
PORT 7: NO FLOW OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 8: NO FLOW OUTLET UNDER BACK PRESSURE.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN, -P9VES)
R1IN= F (-P9VES)
U1IN= F (T9VES)
Y1IN= F (X9VES)
G2IN= F (Q2IN)
R2IN= F (-P9VES)
U2IN= F (T9VES)
Y2IN= F (X9VES)
G3IN= F (Q3IN, -P9VES)
R3IN= F (-P9VES)

U3IN= F (T9VES)
Y3IN= F (X9VES)
G4IN= F (Q4IN)
R4IN= F (-P9VES)
U4IN= F (T9VES)
Y4IN= F (X9VES)
Q5OUT= F (G5OUT)
P5OUT= F (P9VES)
T5OUT= F (T9VES)
X5OUT= F (X9VES)
Q6OUT= F (G6OUT, P9VES)
P6OUT= F (P9VES)
T6OUT= F (T9VES)
X6OUT= F (X9VES)
Q7OUT= F (G7OUT)
P7OUT= F (P9VES)
T7OUT= F (T9VES)
X7OUT= F (X9VES)
Q8OUT= F (G8OUT, P9VES)
P8OUT= F (P9VES)
T8OUT= F (T9VES)
X8OUT= F (X9VES)
P9VES= F (G1IN, G2IN, -Q5OUT, -Q6OUT, T9VES)
T9VES= F (T1IN, T2IN, T3IN, T4IN)
X9VES= F (X1IN, X2IN, X3IN, X4IN)

4)   EVENT STATEMENTS

V G1IN NONE: P9VES LO
V G1IN REV: P9VES LO
V G2IN NONE: P9VES LO
V G2IN REV: P9VES LO
V G3IN SOME: P9VES HI
V G3IN REV: P9VES LO
V G4IN SOME: P9VES HI
V G4IN REV: P9VES LO
V Q5OUT NONE: P9VES HI
V Q6OUT NONE: P9VES HI
V Q6OUT REV: P9VES HI
V Q7OUT SOME: P9VES LO
V Q8OUT SOME: P9VES LO
V Q8OUT REV: P9VES HI
V P9VES HI: G1IN NONE, R1IN NONE, G2IN NONE, R2IN NONE, G1IN REV, R1IN REV
V P9VES HI: G2IN REV, R2IN REV, G3IN REV, R3IN REV, G4IN REV, R4IN REV
V P9VES NONE: Q6OUT REV, P6OUT REV, Q8OUT REV, P8OUT REV
F LK-LP-EN: P9VES LO
F EXT-HEAT: P9VES HI, T9VES HI

F EXT-COLD: P9VES LO, T9VES LO
S NORMAL: G1IN SOME, R1IN SOME, G2IN SOME, R2IN SOME, G3IN NONE, R3IN NONE
S NORMAL: G4IN NONE, R4IN NONE, Q5OUT SOME, P5OUT SOME, Q6OUT SOME, P6OUT SOME
S NORMAL: Q7OUT NONE, P7OUT NONE, Q8OUT NONE, P8OUT NONE
S IMPOSS: Q5OUT REV, P5OUT REV, Q7OUT REV, P7OUT REV

5)   DECISION TABLES

V Q6OUT REV V U6OUT HI T T9VES HI
V Q6OUT REV V U6OUT LO T T9VES LO
V Q6OUT REV V Y6OUT HI T X9VES HI
V Q6OUT REV V Y6OUT LO T X9VES LO
V Q8OUT REV V U8OUT HI T T9VES HI
V Q8OUT REV V U8OUT LO T T9VES LO
V Q8OUT REV V Y8OUT HI T X9VES HI
V Q8OUT REV V Y8OUT LO T X9VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 5 AND 6.

NO MULTI-COMPONENT FEATURES

C.4 Vacuum Gas Vessel Model Template

This is a closed vessel used for storing gas below atmospheric pressure. This implies that a leak on a port connection will cause air to flow into the vessel. Hence it is impossible for inlet ports to have reverse flow if they are connected to a reciprocating type pump. There are five inlet ports and three outlet ports.

Port 1 is an inlet connected to a running pump (or compressor) with no pressure effects. The pump is defined as a reciprocating pump and hence, the port cannot have reverse flow. Port 2 is connected to a running pump defined as a centrifugal pump and port 3 has flow caused by back-pressure from an upstream unit. Both these ports are affected by the vessel pressure. Port 4 is the same as port 1 except the pump is not running and a closed valve prevents flow. Port 5 is the same as port 3 except a closed valve prevents flow.

The outlet ports are all connected to pumps. None of the ports are affected by the vessel pressure. Port 6 is defined as connected to a running reciprocating pump and reverse flow is possible through the port due to the lower internal pressure. Port 7 is defined as connected to a running centrifugal pump. Port 8 is the same as port 6 but the pump is not running and a closed valve prevents flow.

The entire model is shown on the following pages.

1)	MODEL	NUMBER	NAME
		92		TEMPLATE VESSEL MODEL (GAS, CLOSED)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 24
NO. OF PROPAGATION EQUATIONS: 31
NO. OF EVENT STATEMENTS: 26
NO. OF DECISION TABLES: 12
NO. OF FAILURE MODES: 1

2)	ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

USE THIS MODEL TO DEVELOP YOUR OWN MODELS OR USE IT AS IT IS.
PORTS CAN BE REMOVED IF SO DESIRED BY FOLLOWING THIS EXAMPLE.
IF PORT 3 IS NOT REQUIRED, DELETE PROPAGATION EQUATIONS 9 TO 12. DELETE THE G3IN,
T3IN AND X3IN TERMS FROM PROPAGATION EQUATIONS 33, 34 AND 35. DELETE EVENT
STATEMENTS 4 , 5 AND 16. DELETE THE G3IN SOME AND R3IN SOME TERMS FROM EVENT
STATEMENT 23.
ALTER THE NUMBERS FOR THE NUMBER OF PROPAGATION EQUATIONS, EVENT STATEMENTS
AND DECISION TABLES AS NECESSARY AT THE TOP OF THE MODEL.
HAVING CHANGED THE MODEL IT MUST BE GIVEN A NEW FILE NAME AND CORRESPONDING
MODEL NUMBER.

IT IS NOT NECESSARY TO CHANGE THE PORT NUMBERS BUT ENSURE THE CORRECT PORT
NUMBERS ARE GIVEN IN THE MASTER PROGRAM.

THIS IS A CLOSED VESSEL USED TO STORE GAS BELOW ATMOSPHERIC PRESSURE.
PORT 1: NORMAL FLOW INLET FROM RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 2: NORMAL FLOW INLET FROM CENTRIFUGAL PUMP
PORT 3: NORMAL FLOW INLET UNDER BACK-PRESSURE
PORT 4: NO FLOW INLET FROM RECIPROCATING PUMP, NO REVERSE FLOW ALLOWED
PORT 5: NO FLOW INLET UNDER BACK-PRESSURE
PORT 6: NORMAL FLOW OUTLET TO RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 7: NORMAL FLOW OUTLET TO CENTRIFUGAL PUMP
PORT 8: NO FLOW OUTLET TO RECIPROCATING PUMP, REVERSE FLOW ALLOWED.

3)	PROPAGATION EQUATIONS

G1IN= F (Q1IN)
R1IN= F (-P9VES)
G2IN= F (Q2IN, -P9VES)
R2IN= F (-P9VES)
U2IN= F (T9VES)
Y2IN= F (X9VES)
G3IN= F (Q3IN, -P9VES)
R3IN= F (-P9VES)
U3IN= F (T9VES)
Y3IN= F (X9VES)

C-14

G4IN= F (Q4IN)
R4IN= F (-P9VES)
G5IN= F (Q5IN, -P9VES)
R5IN= F (-P9VES)
U5IN= F (T9VES)
Y5IN= F (X9VES)
Q6OUT= F (G6OUT)
P6OUT= F (P9VES)
T6OUT= F (T9VES)
X6OUT= F (X9VES)
Q7OUT= F (G7OUT)
P7OUT= F (P9VES)
T7OUT= F (T9VES)
X7OUT= F (X9VES)
Q8OUT= F (G8OUT)
P8OUT= F (P9VES)
T8OUT= F (T9VES)
X8OUT= F (X9VES)
P9VES= F (G1IN, G2IN, G3IN, -Q6OUT, -Q7OUT, T9VES)
T9VES= F (T1IN, T2IN, T3IN, T4IN, T5IN)
X9VES= F (X1IN, X2IN, X3IN, X4IN, X5IN)

4)   EVENT STATEMENTS

V G1IN NONE: P9VES LO
V G2IN NONE: P9VES LO
V G2IN REV: P9VES LO
V G3IN NONE: P9VES LO
V G3IN REV: P9VES LO
V G4IN SOME: P9VES HI
V G5IN SOME: P9VES HI
V G5IN REV: P9VES LO
V Q6OUT NONE: P9VES HI
V Q6OUT REV: P9VES HI
V Q7OUT NONE: P9VES HI
V Q7OUT REV: P9VES HI
V Q8OUT SOME: P9VES LO
V Q8OUT REV: P9VES HI
V P9VES HI: G1IN NONE, R1IN NONE, G2IN NONE, R2IN NONE, G2IN REV, R2IN REV
V P9VES HI: G3IN NONE, R3IN NONE, G3IN REV, R3IN REV
V P9VES NONE: Q6OUT REV, P6OUT REV, Q7OUT REV, P7OUT REV
V P9VES NONE: Q8OUT REV, P8OUT REV
F LK-LP-EN: P9VES LO
F LK-HP-EN: P9VES HI, P9VES NOR, X9VES HI
F EXT-HEAT: P9VES HI, T9VES HI
F EXT-COLD: P9VES LO, T9VES LO
S NORMAL: G1IN SOME, R1IN SOME, G2IN SOME, R2IN SOME, G3IN SOME, R3IN SOME

C-15

S NORMAL: G4IN NONE, R4IN NONE, G5IN NONE, R5IN NONE, Q6OUT SOME, P6OUT SOME
S NORMAL: Q7OUT SOME, P7OUT SOME, Q8OUT NONE, P8OUT NONE
S IMPOSS: G1IN REV, R1IN REV, G4IN REV, R4IN REV

5)   DECISION TABLES

V Q6OUT REV V U6OUT HI T T9VES HI
V Q6OUT REV V U6OUT LO T T9VES LO
V Q6OUT REV V Y6OUT HI T X9VES HI
V Q6OUT REV V Y6OUT LO T X9VES LO
V Q7OUT REV V U7OUT HI T T9VES HI
V Q7OUT REV V U7OUT LO T T9VES LO
V Q7OUT REV V Y7OUT HI T X9VES HI
V Q7OUT REV V Y7OUT LO T X9VES LO
V Q8OUT REV V U8OUT HI T T9VES HI
V Q8OUT REV V U8OUT LO T T9VES LO
V Q8OUT REV V Y8OUT HI T X9VES HI
V Q8OUT REV V Y8OUT LO T X9VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 3, 6 AND 7.

NO MULTI-COMPONENT FEATURES

C.5 Gas And Liquid Vessel Model Template

This is a closed vessel which can be used to store gases and liquids together. All the liquid ports are assumed to be below the liquid surface and the gas ports above the surface such that there is no break through of fluid types for the ports. The vessel pressure is above atmospheric, therefore fluid will flow out of the vessel should a leak occur in the piping. There are five inlet ports and three outlet ports.

Port 1 is a liquid inlet connected to a running pump. The vessel pressure affects this port and the pump is defined as a low pressure centrifugal pump. Port 2 is the same but the pump is stopped and a closed valve prevents flow. Port 3 is a gas inlet connected to a running pump (or compressor) which is not affected by the vessel pressure. The pump is defined as a reciprocating pump but reverse flow can occur because the higher internal pressure means fluid will flow out of the vessel should there be a leak. Port 4 is the same but the pump is stopped and a closed valve prevents flow. Port 5 is a gas inlet connected to a closed valve and stopped pump. The pump is defined as a centrifugal pump because any flow through this port would be affected by the vessel pressure.

Outlet port 6 is a liquid outlet connected to a running pump which is not affected by the vessel pressure. The pump is defined as a centrifugal pump. Port 7 is a gas outlet connected to a running pump defined as a reciprocating pump, reverse flow is not allowed through the port. This port is not affected by the vessel pressure. Port 8 is a gas outlet connected to a closed valve, any flow would be caused by the internal vessel pressure.

Note that in this model the LK-HP-EN event statement has been removed because the vessel is above atmospheric pressure.

The model is shown on the following pages.

1)   MODEL   NUMBER   NAME
              93   TEMPLATE VESSEL MODEL (GAS AND LIQUID, CLOSED)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 27
NO. OF PROPAGATION EQUATIONS: 36
NO. OF EVENT STATEMENTS: 29
NO. OF DECISION TABLES: 9
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

USE THIS MODEL TO DEVELOP YOUR OWN MODELS OR USE IT AS IT IS.
PORTS CAN BE REMOVED IF SO DESIRED BY FOLLOWING THIS EXAMPLE.
IF PORT 8 IS NOT REQUIRED, DELETE PROPAGATION EQUATIONS 29 TO 32. DELETE EVENT STATEMENTS 16 AND 17. DELETE THE Q8OUT REV AND P8OUT REV TERMS FROM EVENT STATEMENT 21. DELETE THE Q8OUT NONE AND P8OUT NONE TERMS FROM EVENT STATEMENT 28. DELETE DECISION TABLES 6 TO 9.
ALTER THE NUMBERS FOR THE NUMBER OF PROPAGATION EQUATIONS, EVENT STATEMENTS AND DECISION TABLES AS NECESSARY AT THE TOP OF THE MODEL.
HAVING CHANGED THE MODEL IT MUST BE GIVEN A NEW FILE NAME AND CORRESPONDING MODEL NUMBER.

IT IS NOT NECESSARY TO CHANGE THE PORT NUMBERS BUT ENSURE THE CORRECT PORT NUMBERS ARE GIVEN IN THE MASTER PROGRAM.

THIS IS A CLOSED VESSEL USED TO STORE LIQUIDS AND GASES TOGETHER, ALL LIQUID PORTS ARE BELOW THE LEVEL OF THE LIQUID SURFACE AND THE VESSEL PRESSURE IS ABOVE ATMOSPHERIC.
PORT 1: NORMAL FLOW LIQUID INLET FROM LOW PRESSURE CENTRIFUGAL PUMP
PORT 2: NO FLOW LIQUID INLET FROM LOW PRESSURE CENTRIFUGAL PUMP
PORT 3: NORMAL FLOW GAS INLET FROM RECIPROCATING PUMP, REVERSE FLOW
        ALLOWED
PORT 4: NO FLOW GAS INLET FROM RECIPROCATING PUMP, REVERSE FLOW ALLOWED
PORT 5: NO FLOW GAS INLET FROM CENTRIFUGAL PUMP
PORT 6: NORMAL FLOW LIQUID OUTLET TO CENTRIFUGAL PUMP
PORT 7: NORMAL FLOW GAS OUTLET TO RECIPROCATING PUMP, NO REVERSE FLOW
        ALLOWED
PORT 8: NO FLOW GAS OUTLET UNDER BACK-PRESSURE.

3)   PROPAGATION EQUATIONS

$G1IN = F (Q1IN, -P9VES)$
$R1IN = F (-P9VES)$
$U1IN = F (T9VES)$
$Y1IN = F (X9VES)$
$G2IN = F (Q2IN, -P9VES)$
$R2IN = F (-P9VES)$

C-18

U2IN= F (T9VES)
Y2IN= F (X9VES)
G3IN= F (Q3IN)
R3IN= F (-P9VES)
U3IN= F (T9VES)
Y3IN= F (X9VES)
G4IN= F (Q4IN)
R4IN= F (-P9VES)
U4IN= F (T9VES)
Y4IN= F (X9VES)
G5IN= F (Q5IN, -P9VES)
R5IN= F (-P9VES)
U5IN= F (T9VES)
Y5IN= F (X9VES)
Q6OUT= F (G6OUT)
P6OUT= F (P9VES)
T6OUT= F (T9VES)
X6OUT= F (X9VES)
Q7OUT= F (G7OUT)
P7OUT= F (P9VES)
T7OUT= F (T9VES)
X7OUT= F (X9VES)
Q8OUT= F (G8OUT, P9VES)
P8OUT= F (P9VES)
T8OUT= F (T9VES)
X8OUT= F (X9VES)
L9VES= F (G1IN, -Q6OUT)
P9VES= F (G3IN, -Q7OUT, L9VES, T9VES)
T9VES= F (T1IN, T2IN, T3IN, T4IN, T5IN)
X9VES= F (X1IN, X2IN, X3IN, X4IN, X5IN)

4)   EVENT STATEMENTS

V G1IN NONE: L9VES LO, P9VES LO
V G1IN REV: L9VES LO, L9VES NONE, P9VES LO
V G2IN SOME: L9VES HI, P9VES HI
V G2IN REV: L9VES LO, L9VES NONE, P9VES LO
V G3IN NONE: P9VES LO
V G3IN REV: P9VES LO
V G4IN SOME: P9VES HI
V G4IN REV: P9VES LO
V G5IN SOME: P9VES HI
V G5IN REV: P9VES LO
V G1IN NONE: C(DUMMY)
I C(DUMMY): L9VES NONE
V Q6OUT NONE: L9VES HI, P9VES HI
V Q6OUT REV: L9VES HI, P9VES HI

C-19

V Q7OUT NONE: P9VES HI
V Q8OUT SOME: P9VES LO
V Q8OUT REV: P9VES HI
V P9VES HI: G1IN NONE, R1IN NONE, G1IN REV, R1IN REV, G2IN REV, R2IN REV
V P9VES HI: G3IN NONE, R3IN NONE, G3IN REV, R3IN REV G4IN REV, R4IN REV
V P9VES HI: G5IN REV R5IN REV
V P9VES NONE: Q6OUT REV, P6OUT REV, Q8OUT REV, P8OUT REV
V L9VES NONE: P9VES LO
F LK-LP-EN: L9VES LO, L9VES NONE, P9VES LO
F EXT-HEAT: P9VES HI, T9VES HI
F EXT-COLD: P9VES LO, T9VES LO
S NORMAL: G1IN SOME, R1IN SOME, G2IN NONE, R2IN NONE, G3IN SOME, R3IN SOME
S NORMAL: G4IN NONE, R4IN NONE, G5IN NONE, R5IN NONE, Q6OUT SOME, P6OUT SOME
S NORMAL: Q7OUT SOME, P7OUT SOME, Q8OUT NONE, P8OUT NONE
S IMPOSS: Q7OUT REV, P7OUT REV

5)   DECISION TABLES

I C(DUMMY) V L9VES NONE T Q6OUT NONE, P6OUT NONE
V Q6OUT REV V U6OUT HI T T9VES HI
V Q6OUT REV V U6OUT LO T T9VES LO
V Q6OUT REV V Y6OUT HI T X9VES HI
V Q6OUT REV V Y6OUT LO T X9VES LO
V Q8OUT REV V U8OUT HI T T9VES HI
V Q8OUT REV V U8OUT LO T T9VES LO
V Q8OUT REV V Y8OUT HI T X9VES HI
V Q8OUT REV V Y8OUT LO T X9VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 3, 6 AND 7.

NO MULTI-COMPONENT FEATURES

C-20

C.6 Vessel Model Generation Subroutine

```
        SUBROUTINE TANKMOD

C       ----------------------------------------------------------------------------------
C       This subroutine is called if automatic vessel model generation is
C       specified.  It automatically generates the propagation equations,
C       event statements and decision tables for the vessel model.
C       ----------------------------------------------------------------------------------
        CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
    *   FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
        INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),INLET,
    *   VES,NF(10),ET(10),RF(10),FLAG(10)

        COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
        COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
        COMMON/TWELVE/LG,INLET,VES,NF,ET,RF,FLAG

        CHARACTER OPT*4,POCH
        INTEGER ANSWER,CL,OUTLET,PO,L1

1       FORMAT(/,' ',A,' ? ',$)
2       FORMAT(' ',A,' ? ',$)
3       FORMAT(/,' ',A)
4       FORMAT(' ',A,I2,') ? ',$)
5       FORMAT(/,' ',A,' ',I2)
C       ----------------------------------------------------------------------------------
C       The FLAGS set by this subroutine are:
C       FLAG(1)=1 for no checking of event statements from the GENEV routine.
C       FLAG(2)=0 for closed vessels but is set to the vessel port number for
C         open vessels to indicate that the pressure terms are to be ignored.
C       FLAG(3) is the number of flow ports.
C       FLAG(4) is the number of liquid inlet flow ports.
C       FLAG(5) is non zero if an overflow decision table is used for C(DUMMY)
C         which takes the mini-top to be D(DUMMY).
C       ----------------------------------------------------------------------------------
        FLAG(1)=1
        PRINT3,'      VESSEL MODEL GENERATION ROUTINE'
        PRINT*,'      ----------------------------------------------------'
```

```
C       ----------------------------------------------------------------------
C       Determine the vessels overall characteristics
C       ----------------------------------------------------------------------
        PRINT3,'QUESTIONS REFER TO NORMAL OPERATING CONDITIONS.'
C       ----------------------------------------------------------------------
C       LG(10) gives fluid type in vessel : 0-liquid, 1-gas, 2-both.
C       Any level effects are handle in the vessel pressure equation.
C       ----------------------------------------------------------------------
        PRINT1,'IS THE VESSEL FOR STORING LIQUID, GAS OR BOTH, L/G/B'
        OPT='LGB'
        LG(10)=ANSWER(OPT)
C       ----------------------------------------------------------------------
C       CL is used to indicate if the vessel is open (0) or closed (1).
C       ----------------------------------------------------------------------
        IF (LG(10).NE.0) THEN
         CL=1
        ELSE
         PRINT2,'IS THE TANK AN OPEN OR CLOSED VESSEL, O/C'
         OPT='OC'
         CL=ANSWER(OPT)
        END IF
C       ----------------------------------------------------------------------
C       Generate the initial description array and print user information.
C       ----------------------------------------------------------------------
        IL=IL+1
        AD(IL)='THIS IS A'
        L1=LENGTH(AD(IL))
        IF (CL.EQ.1) THEN
         AD(IL)=AD(IL)(1:L1)//' CLOSED'
        ELSE
         AD(IL)=AD(IL)(1:L1)//' OPEN'
        END IF
        L1=LENGTH(AD(IL))
        AD(IL)=AD(IL)(1:L1)//' VESSEL USED TO STORE'
        L1=LENGTH(AD(IL))
        IF (LG(10).EQ.0) THEN
         AD(IL)=AD(IL)(1:L1)//' LIQUIDS ONLY.'
        ELSE IF (LG(10).EQ.1) THEN
         AD(IL)=AD(IL)(1:L1)//' GASES ONLY.'
        ELSE
         AD(IL)=AD(IL)(1:L1)//' LIQUIDS AND GASES TOGETHER.'
```

```
        END IF
        IF (LG(10).NE.1) THEN
         IL=IL+1
         IF (LG(10).EQ.0) THEN
          AD(IL)='ALL'
         ELSE
          AD(IL)='ALL LIQUID INLET'
         END IF
         L1=LENGTH(AD(IL))
         AD(IL)=AD(IL)(1:L1)//' PORT OPENINGS ARE BELOW THE LEVEL OF THE LIQUID SURFACE.'
        END IF

        PRINT*,' '
        DO 40,I=1,IL
40      PRINT*,AD(I)
C       --------------------------------------------------------------------------------
C       Determine the number of inlet and outlet ports.
C       --------------------------------------------------------------------------------
50      PRINT3,'MAXIMUM NUMBER OF PORTS, INCLUDING VESSEL PORT, IS 9,'
60      PRINT1,'NUMBER OF INLET PORTS, (1 TO 7)'
        READ(*,*,ERR=60)INLET
        IF (INLET.LT.1.OR.INLET.GT.7) GOTO60
70      PRINT4,'NUMBER OF OUTLET PORTS, (1 TO',(8-INLET)
        READ(*,*,ERR=70)OUTLET
        IF (OUTLET.LT.1.OR.OUTLET.GT.(8-INLET)) GOTO50
C       --------------------------------------------------------------------------------
C       Calculate the number of the vessel port and if the vessel is closed
C       set FLAG(2) to the vessel port number.
C       --------------------------------------------------------------------------------
        VES=INLET+OUTLET+1
        IF (CL.EQ.0) FLAG(2)=VES
C       --------------------------------------------------------------------------------
C       Determine the characteristics for each port in turn.
C       --------------------------------------------------------------------------------
        DO 100,PO=1,VES-1
         IF (PO.LE.INLET) THEN
          PRINT5,'INLET PORT',PO
          PRINT*,'----------------------'
         ELSE
          PRINT5,'OUTLET PORT',PO
          PRINT*,'--------------------------'
```

```
      END IF
      IL=IL+1
      WRITE(POCH,'(I1)')PO
      AD(IL)='PORT '//POCH
      L1=LENGTH(AD(IL))
C     ------------------------------------------------------------------
C     Set NF(port number) to 0 for flow or 1 for no flow.  Set NF(10)
C     to 1 if it is a no flow port or add 1 to FLAG(3) for flow port.
C     ------------------------------------------------------------------
      PRINT1,'DOES THIS PORT NORMALLY HAVE FLOW, Y/N'
      OPT='YN'
      NF(PO)=ANSWER(OPT)
      IF (NF(PO).EQ.1) THEN
       NF(10)=1
       AD(IL)=AD(IL)(1:L1)//': NO FLOW'
      ELSE
       FLAG(3)=FLAG(3)+1
       AD(IL)=AD(IL)(1:L1)//': NORMAL FLOW'
      END IF
      L1=LENGTH(AD(IL))
C     ------------------------------------------------------------------
C     Set LG(port number) to fluid type, 0 - liquid or 1 - gas.
C     For no flow ports on liquid and gas vessels the position of the port
C     relative to the liquid level is used.
C     If the port is an inlet for liquid and has flow add 1 to FLAG(4).
C     ------------------------------------------------------------------
      IF (LG(10).EQ.2) THEN
       IF (NF(PO).EQ.0) THEN
        PRINT1,'IS THIS PORT FOR LIQUID OR GAS, L/G'
        OPT='LG'
       ELSE
        PRINT1,'IS THIS PORT NORMALLY ABOVE OR BELOW LIQUID LEVEL, A/B'
        OPT='BA'
       END IF
       LG(PO)=ANSWER(OPT)
       IF (LG(PO).EQ.0) THEN
        AD(IL)=AD(IL)(1:L1)//' LIQUID'
       ELSE
        AD(IL)=AD(IL)(1:L1)//' GAS'
       END IF
       L1=LENGTH(AD(IL))
```

```
        ELSE
         LG(PO)=LG(10)
        END IF
        IF (PO.LE.INLET.AND.LG(PO).EQ.0.AND.NF(PO).EQ.0) FLAG(4)=FLAG(4)+1
C       ----------------------------------------------------------------------------
C       Add the port type to the descirption.
C       ----------------------------------------------------------------------------
        IF (PO.LE.INLET) THEN
         AD(IL)=AD(IL)(1:L1)//' INLET FROM'
        ELSE
         AD(IL)=AD(IL)(1:L1)//' OUTLET TO'
        END IF
        L1=LENGTH(AD(IL))
C       ----------------------------------------------------------------------------
C       Set ET(port number) to the type of flow for the port, 0 for positive
C       displacement pumped, 1 for centrifugally pumped and 2 for gravity/
C       back pressure.
C       ----------------------------------------------------------------------------
        IF (NF(PO).EQ.0) THEN
         PRINT3,'IS FLOW POSITIVE DISPLACEMENT OR CENTRIFUGALLY PUMPED OR'
        ELSE
         PRINT3,'WOULD FLOW BE POSITIVE DISPLACEMENT OR CENTRIFUGALLY PUMPED OR'
        END IF
        IF (LG(PO).EQ.1) THEN
         IF (PO.LE.INLET) THEN
          PRINT2,'BACK PRESSURE FED, P/C/G'
         ELSE
          PRINT2,'BACK PRESSURE BLED, P/C/G'
         END IF
        ELSE
         IF (PO.LE.INLET) THEN
          PRINT2,'GRAVITY FED, P/C/G'
         ELSE
          PRINT2,'GRAVITY BLED, P/C/G'
         ENDIF
        END IF
        OPT='PCG'
        ET(PO)=ANSWER(OPT)
```

```
C       ---------------------------------------------------------------
C       For both inlet and outlet ports, leave RF(port number) as 0 for
.  C    centrifugal and gravity systems, these allow reverse flow or flow
C       with pump off.
C       Positive displacement systems may or may not allow reverse flow to
C       occur.  Set to 1 for no reverse flow.
C       RF(10) set to 1 if reverse flow restrictions occur.
C       ---------------------------------------------------------------
        IF (ET(PO).EQ.0) THEN
        AD(IL)=AD(IL)(1:L1)//' +VE DISPLACEMENT PUMP,'
        L1=LENGTH(AD(IL))
        PRINT1,'IS REVERSE FLOW TO BE ALLOWED IN THIS CONNECTION, Y/N'
        OPT='YN'
        RF(PO)=ANSWER(OPT)
        IF (RF(PO).EQ.1) THEN
         RF(10)=1
         AD(IL)=AD(IL)(1:L1)//' NO'
         L1=LENGTH(AD(IL))
        END IF
        AD(IL)=AD(IL)(1:L1)//' REVERSE FLOW ALLOWED.'
        END IF
C       ---------------------------------------------------------------
C       Gravity fed systems for closed vessels - inlet and outlet ports.
C       ---------------------------------------------------------------
        IF (ET(PO).EQ.2) THEN
        IF (LG(PO).EQ.0) THEN
         AD(IL)=AD(IL)(1:L1-3)//' UNDER GRAVITY.'
        ELSE
         AD(IL)=AD(IL)(1:L1-3)//' UNDER BACK PRESSURE.'
        END IF
        IF (FLAG(2).EQ.0) THEN
         PRINT3,'IT IS ASSUMED THAT SONIC FLOW IS UNLIKELY SO THAT'
         PRINT*,'FLOW IS DEPENDENT ON PRESSURE DIFFERENCE.'
        END IF
        END IF
C       ---------------------------------------------------------------
C       Inlet centrifugal systems : Gases use compressor, pressure effects.
C       Liquids may be affected by the pressure, change ET to 0 or 2.
C       ---------------------------------------------------------------
```

```fortran
      IF (PO.LE.INLET.AND.ET(PO).EQ.1) THEN
       IF (LG(PO).EQ.1) THEN
        ET(PO)=2
        AD(IL)=AD(IL)(1:L1)//' CENTRIFUGAL PUMP.'
       ELSE
        PRINT1,'IS VESSEL PRESSURE (INCLUDING LIQUID HEAD) NEAR TO PUMP PRESSURE, Y/N'
        OPT='NY'
        ET(PO)=ANSWER(OPT)
        IF (ET(PO).EQ.1) THEN
         ET(PO)=2
         AD(IL)=AD(IL)(1:L1)//' LOW PRESSURE CENTRIFUGAL PUMP.'
        ELSE
         AD(IL)=AD(IL)(1:L1)//' HIGH PRESSURE CENTRIFUGAL PUMP.'
        END IF
       END IF
      END IF
C     ------------------------------------------------------------------
C     Outlet centrifugal systems are not affected by pressure, set ET to 0.
C     ------------------------------------------------------------------
      IF (ET(PO).EQ.1) THEN
       ET(PO)=0
       AD(IL)=AD(IL)(1:L1)//' CENTRIFUGAL PUMP.'
      END IF
C     ------------------------------------------------------------------
C     Having determined the nature of the port, print the result.
C     ------------------------------------------------------------------
      IF (ET(PO).EQ.0) THEN
       PRINT3,'PRESSURE AND/OR LEVEL EFFECTS ARE NEGLIGABLE.'
      ELSE
       PRINT3,'PRESSURE AND/OR LEVEL EFFECTS WILL BE CONSIDERED.'
      END IF
100   CONTINUE
C     ------------------------------------------------------------------
C     Generate the propagation equation.
C     ------------------------------------------------------------------
      CALL TANKPROP
C     ------------------------------------------------------------------
C     Generate the event statements.
C     ------------------------------------------------------------------
      PRINT3,'GENERATING THE EVENT STATEMENTS - PLEASE WAIT.'
      CALL TANKSTATE
```

```
C       ----------------------------------------------------------------
C       Generate the decision tables.
C       ----------------------------------------------------------------
        PRINT3,'GENERATING THE DECISION TABLES - PLEASE WAIT.'
        CALL TANKTABLE
C       ----------------------------------------------------------------
C       Determine the supplimentary information.
C       The normal state lists the ports where flow normally exists.
C       ----------------------------------------------------------------
        IF (FLAG(3).EQ.1) THEN
        DO 200,PO=1,VES-1
         IF (NF(PO).EQ.0) THEN
          WRITE(POCH,'(I1)')PO
          NS='FLOW IS NORMAL THROUGH PORT '//POCH//'.'
         END IF
200     CONTINUE
        ELSE
         NS='FLOW IS NORMAL THROUGH PORTS'
         DO 210,PO=1,VES-1
          IF (NF(PO).EQ.0) THEN
           WRITE(POCH,'(I1)')PO
           L1=LENGTH(NS)
           NS=NS(1:L1)//' '//POCH//','
          END IF
210      CONTINUE
         NS=NS(1:L1-1)//' AND '//POCH//'.'
        END IF
C       ----------------------------------------------------------------
C       The are no multicomponent features.
C       ----------------------------------------------------------------
        MC='N'
C       ----------------------------------------------------------------
C       The generation of the model is complete.
C       ----------------------------------------------------------------
        RETURN
        END
```

SUBROUTINE TANKPROP

```fortran
C       -----------------------------------------------------------------------
C       Subroutine to generate the propagation equation for the vessel model.
C       -----------------------------------------------------------------------
        CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
     *   FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
        INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),INLET,
     *   VES,NF(10),ET(10),RF(10),FLAG(10)

        COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
        COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
        COMMON/TWELVE/LG,INLET,VES,NF,ET,RF,FLAG

        CHARACTER VESCH,POCH,LEVEL*75,PRESS*75,TEMPFE(45)*80,IN*40,BASICF*10
        INTEGER PO

3       FORMAT(/,' ',A)
C       -----------------------------------------------------------------------
C       Initialize the temporary storage strings.
C       -----------------------------------------------------------------------
        LEVEL='F('
        PRESS='F('
        DO 100,I=1,45
100       TEMPFE(I)=FE(I)
        WRITE(VESCH,'(I1)')VES
C       -----------------------------------------------------------------------
C       Take each port in turn.
C       -----------------------------------------------------------------------
        DO 200,PO=1,VES-1
         WRITE(POCH,'(I1)')PO
         IF (PO.LE.INLET) THEN
C        -----------------------------------------------------------------------
C        Determination of 'IN' propagation equations.
C        -----------------------------------------------------------------------
         TEMPFE(FL+1)='G'//POCH//'IN=F(Q'//POCH//'IN,-P'//VESCH//'VES)'
         IF (ET(PO).EQ.0) TEMPFE(FL+1)=TEMPFE(FL+1)(:11)//')'
         TEMPFE(FL+2)='R'//POCH//'IN=F(-P'//VESCH//'VES)'
         FL=FL+2
         IF (RF(PO).EQ.0) THEN
          TEMPFE(FL+1)='U'//POCH//'IN=F(T'//VESCH//'VES)'
```

```fortran
          TEMPFE(FL+2)='Y'//POCH//'IN=F(X'//VESCH//'VES)'
          FL=FL+2
        END IF
      ELSE
```
C
```
C         ----------------------------------------------------------------
C         Determination of 'OUT' propagation equations.
C         ----------------------------------------------------------------
        TEMPFE(FL+1)='Q'//POCH//'OUT=F(G'//POCH//'OUT,P'//VESCH//'VES)'
        IF (ET(PO).EQ.0) TEMPFE(FL+1)=TEMPFE(FL+1)(:13)//')'
        TEMPFE(FL+2)='P'//POCH//'OUT=F(P'//VESCH//'VES)'
        TEMPFE(FL+3)='T'//POCH//'OUT=F(T'//VESCH//'VES)'
        TEMPFE(FL+4)='X'//POCH//'OUT=F(X'//VESCH//'VES)'
        FL=FL+4
      END IF
C     ----------------------------------------------------------------
C     Write flow term to appropriate level or pressure string for
C     use in the vessel port equations if it is a flow port.
C     ----------------------------------------------------------------
      IF (NF(PO).EQ.0) THEN
        IF (PO.LE.INLET) THEN
          BASICF='G'//POCH//'IN,'
        ELSE
          BASICF='-Q'//POCH//'OUT,'
        END IF
        IF (LG(PO).EQ.0) THEN
          L1=LENGTH(LEVEL)
          LEVEL=LEVEL(1:L1)//BASICF
        ELSE
          L1=LENGTH(PRESS)
          PRESS=PRESS(1:L1)//BASICF
        END IF
      END IF
200   CONTINUE
C     ----------------------------------------------------------------
C     Determination of 'VES' propagation equations.
C     Generate level term, if liquid present, from LEVEL string.
C     ----------------------------------------------------------------
      IF (LG(10).NE.1) THEN
        L1=LENGTH(LEVEL)-1
```

```
        FL=FL+1
        TEMPFE(FL)='L'//VESCH//'VES='//LEVEL(1:L1)//')'
        END IF
C       ----------------------------------------------------------------------------------
C       Generate pressure term : liquid only vessels only have the
C       level term, others taken from PRESS string.  Add temperature if
C       vessel is closed.
C       ----------------------------------------------------------------------------------
        FL=FL+1
        IF (LG(10).EQ.0) THEN
         TEMPFE(FL)='P'//VESCH//'VES=F(L'//VESCH//'VES)'
        ELSE
         L1=LENGTH(PRESS)-1
         TEMPFE(FL)='P'//VESCH//'VES='//PRESS(1:L1)//')'
         IF (LG(10).NE.1) THEN
          L1=LENGTH(TEMPFE(FL))-1
          TEMPFE(FL)=TEMPFE(FL)(1:L1)//',L'//VESCH//'VES)'
         END IF
        END IF
        IF (FLAG(2).EQ.0) THEN
         L1=LENGTH(TEMPFE(FL))-1
         TEMPFE(FL)=TEMPFE(FL)(1:L1)//',T'//VESCH//'VES)'
        END IF
C       ----------------------------------------------------------------------------------
C       Generate the temperature term.
C       ----------------------------------------------------------------------------------
        IN='T1IN,T2IN,T3IN,T4IN,T5IN,T6IN,T7IN,'
        I=5*INLET-1
        FL=FL+1
        TEMPFE(FL)='T'//VESCH//'VES=F('//IN(:I)//')'
C       ----------------------------------------------------------------------------------
C       Generate the composition term.
C       ----------------------------------------------------------------------------------
        IN='X1IN,X2IN,X3IN,X4IN,X5IN,X6IN,X7IN,'
        FL=FL+1
        TEMPFE(FL)='X'//VESCH//'VES=F('//IN(:I)//')'
C       ----------------------------------------------------------------------------------
C       Completion of propagation equations, now check.
C       ----------------------------------------------------------------------------------
        PRINT3,'THE FOLLOWING PROPAGATION EQUATIONS HAVE BEEN CREATED:'
        CALL CHECK(TEMPFE,FL)
```

```fortran
C      ----------------------------------------------------------------
C      Take prop. eqs. from the temporary array and place in the FE array.
C      ----------------------------------------------------------------
       K=FL
       FL=0
       DO 210,I=1,K
        IF (TEMPFE(I)(1:1).NE.' ') THEN
         FL=FL+1
         L1=INDEX(TEMPFE(I),' ')
         IF (L1.EQ.0) L1=80
         FE(FL)=TEMPFE(I)(1:L1)
        END IF
210     CONTINUE

        RETURN
        END



        SUBROUTINE TANKSTATE


C      ----------------------------------------------------------------
C      Subrouitine to generate the event statements for the vessel model.
C      ----------------------------------------------------------------
       CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
      *  FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
       INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),INLET,
      *  VES,NF(10),ET(10),RF(10),FLAG(10)

       COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
       COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
       COMMON/TWELVE/LG,INLET,VES,NF,ET,RF,FLAG

       CHARACTER VESCH,POCH,TEMPFI(50)*80,BASICF*10,TEMPEV(25)*12
       INTEGER PO,FAUPOR(50,12)

3       FORMAT(/,' ',A)
C      ----------------------------------------------------------------
C      Initialize the temporary storage strings.
C      ----------------------------------------------------------------
```

```
          DO 100,I=1,50
          TEMPFI(I)=FI(I)
100       FAUPOR(I,1)=0
          WRITE(VESCH,'(I1)')VES
C         -----------------------------------------------------------------------------------
C         Consider the effect of each inlet port on the level and pressure.
C         -----------------------------------------------------------------------------------
          DO 200,PO=1,INLET
          WRITE(POCH,'(I1)')PO
C         -----------------------------------------------------------------------------------
C         For a non-flow inlet port SOME flow may be a cause of high level
C         or pressure.
C         -----------------------------------------------------------------------------------
          IF (NF(PO).EQ.1) THEN
          JL=JL+1
          IF (LG(PO).EQ.0) THEN
           TEMPFI(JL)='V G'//POCH//' IN SOME:L'//VESCH//'VES HI'
           IF (FLAG(2).EQ.0) THEN
           L1=LENGTH(THEMPFI(JL)
           TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH//'VES HI'
           END IF
          ELSE
           TEMPFI(JL)='V G'//POCH//' IN SOME:P'//VESCH//'VES HI'
          END IF
          ELSE
C         -----------------------------------------------------------------------------------
C         For a flow inlet port NO flow may be a cause of low level
C         or pressure.
C         -----------------------------------------------------------------------------------
          JL=JL+1
          IF (LG(PO).EQ.0) THEN
           TEMPFI(JL)='V G'//POCH//' IN NONE:L'//VESCH//'VES LO'
           IF (FLAG(2).EQ.0) THEN
           L1=LENGTH(THEMPFI(JL)
           TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH//'VES LO'
           END IF
          ELSE
           TEMPFI(JL)='V G'//POCH//' IN NONE:P'//VESCH//'VES LO'
          END IF
          END IF
```

```
C      -----------------------------------------------------------------------
C      If REVerse flow is possible through the inlet port, then
C      this may be a cause of low or no level or low pressure.
C      -----------------------------------------------------------------------
       IF (RF(PO).EQ.0) THEN
       JL=JL+1
       IF (LG(PO).EQ.0) THEN
        TEMPFI(JL)='V G'//POCH//' IN REV:L'//VESCH//'VES LO,L'//VESCH//'VES NONE'
        IF (FLAG(2).EQ.0) THEN
         L1=LENGTH(THEMPFI(JL)
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH//'VES LO'
        END IF
        ELSE
         TEMPFI(JL)='V G'//POCH//' IN REV:P'//VESCH//'VES LO'
        END IF
       END IF
200    CONTINUE
C      -----------------------------------------------------------------------
C      If there is only one liquid inlet flow port then the NONE deviation is
C      a cause of C(DUMMY).  Otherwise the NONE deviation for each port forms
C      a decision table.  This forms part of the no flow out decision table.
C      -----------------------------------------------------------------------
       IF (LG(10).NE.1) THEN
       IF (FLAG(4).EQ.1) THEN
        DO 210,PO=1,INLET
         IF (LG(PO).EQ.0.AND.NF(PO).EQ.0) THEN
          WRITE(POCH,'(I1)')PO
          JL=JL+1
          TEMPFI(JL)='V G'//POCH//' IN NONE:C(DUMMY)'
         END IF
210    CONTINUE
       END IF
C      -----------------------------------------------------------------------
C      The 'no flow in' decision table for C(DUMMY) or the single event
C      statement is also a cause of no level.
C      -----------------------------------------------------------------------
       JL=JL+1
       TEMPFI(JL)='I C(DUMMY):L'//VESCH//'VES NONE'
       END IF
```

```
C       ----------------------------------------------------------------------
C       Now consider the effect of each outlet port on the level and pressure.
C       ----------------------------------------------------------------------
        DO 220,PO=INLET+1,VES-1
        WRITE(POCH,'(I1)')PO
C       ----------------------------------------------------------------------
C       The low deviation may be caused by SOME flow out of a non-flow port.
C       ----------------------------------------------------------------------
        IF (NF(PO).EQ.1) THEN
        JL=JL+1
        IF (LG(PO).EQ.0) THEN
         TEMPFI(JL)='V Q'//POCH//' OUT SOME:L'//VESCH//'VES LO'
         IF (FLAG(2).EQ.0) THEN
         L1=LENGTH(TEMPFI(JL))
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH//'VES LO'
         END IF
        ELSE
         TEMPFI(JL)='V Q'//POCH//' OUT SOME:P'//VESCH//'VES LO'
        END IF
        ELSE
C       ----------------------------------------------------------------------
C       The high deviation may be caused by NO flow out for a normal flow port.
C       ----------------------------------------------------------------------
        JL=JL+1
        IF (LG(PO).EQ.0) THEN
         TEMPFI(JL)='V Q'//POCH//' OUT NONE:L'//VESCH//'VES HI'
         IF (FLAG(2).EQ.0) THEN
         L1=LENGTH(TEMPFI(JL))
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH//'VES HI'
         END IF
        ELSE
         TEMPFI(JL)='V Q'//POCH//' OUT NONE:P'//VESCH//'VES HI'
        END IF
        END IF
C       ----------------------------------------------------------------------
C       The high deviation may also be caused by REVerse flow through
C       an outlet port, if possible.
C       ----------------------------------------------------------------------
```

```
         IF (RF(PO).EQ.0) THEN
          JL=JL+1
          IF (LG(PO).EQ.0) THEN
           TEMPFI(JL)='V Q'//POCH//'OUT REV:L'//VESCH//'VES HI'
           IF (FLAG(2).EQ.0) THEN
            L1=LENGTH(TEMPFI(JL))
            TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH//'VES HI'
           END IF
          ELSE
           TEMPFI(JL)='V Q'//POCH//'OUT REV:P'//VESCH//'VES HI'
          END IF
         END IF
220     CONTINUE
C       -----------------------------------------------------------------------------------------
C       No or reverse flow through an inlet port can be caused by HIgh
C       effective pressure.  Use the EVSTAT routine.
C       -----------------------------------------------------------------------------------------
        BASICF='V P'//VESCH//'VES HI'
        NDEV=0
        DO 230,PO=1,INLET
         WRITE(POCH,'(I1)')PO
         IF (NF(PO).EQ.0) THEN
          TEMPEV(NDEV+1)='G'//POCH//'IN NONE'
          TEMPEV(NDEV+2)='R'//POCH//'IN NONE'
          NDEV=NDEV+2
         END IF
         IF (RF(PO).EQ.0) THEN
          TEMPEV(NDEV+1)='G'//POCH//'IN REV'
          TEMPEV(NDEV+2)='R'//POCH//'IN REV'
          NDEV=NDEV+2
         END IF
230     CONTINUE
        IF (NDEV.NE.0) CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
C       -----------------------------------------------------------------------------------------
C       No or reverse flow through an outlet port can be caused by LOw
C       effective pressure.  Use the EVSTAT routine.
C       -----------------------------------------------------------------------------------------
        BASICF='P'//VESCH//'VES NONE'
        NDEV=0
        DO 240,PO=INLET+1,VES-1
         WRITE(POCH,'(I1)')PO
```

```
      IF (RF(PO).EQ.0) THEN
        TEMPEV(NDEV+1)='Q'//POCH//'OUT REV'
        TEMPEV(NDEV+2)='P'//POCH//'OUT REV'
        NDEV=NDEV+2
      END IF
240   CONTINUE
      IF (NDEV.NE.0) THEN
      J=JL+1
      CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
      DO 245,I=J,JL
245     TEMPFI(I)='V '//TEMPFI(I)
      END IF
C     ------------------------------------------------------------------------------
C     Set up the level and pressure relationship if vessel is
C     closed and liquid is present.
C     ------------------------------------------------------------------------------
      IF (LG(10).NE.1) THEN
      JL=JL+1
      TEMPFI(JL)='V L'//VESCH//'VES NONE:P'//VESCH//'VES LO'
      END IF
C     ------------------------------------------------------------------------------
C     Initialize the FAUPOR array for the basic faults LK-LP-EN,
C     LK-HP-EN (if not an open vessel), EXT-HEAT and EXT-COLD for
C     the vessel port.  Use GENEV for these.
C     ------------------------------------------------------------------------------
      I=1
      FAUPOR(1,1)=1
      IF (FLAG(2).EQ.0) THEN
       FAUPOR(2,1)=2
       I=2
      ENDIF
      FAUPOR(I+1,1)=5
      FAUPOR(I+2,1)=6
      J=1
      I=I+2
      DO 250,K=J,I
       FAUPOR(K,2)=1
250     FAUPOR(K,3)=VES
      CALL GENEV(FAUPOR,TEMPFI)
```

```fortran
C     ----------------------------------------------------------------
C     Create the NORMAL state event statement.
C     ----------------------------------------------------------------
      BASICF='S NORMAL'
      NDEV=0
      DO 260,PO=1,INLET
       WRITE(POCH,'(I1)')PO
       TEMPEV(NDEV+1)='G'//POCH//'IN SOME'
       IF (NF(PO).EQ.1) TEMPEV(NDEV+1)='G'//POCH//'IN NONE'
       TEMPEV(NDEV+2)='R'//POCH//'IN SOME'
       IF (NF(PO).EQ.1) TEMPEV(NDEV+2)='R'//POCH//'IN NONE'
260   NDEV=NDEV+2
      DO 270,PO=INLET+1,VES-1
       WRITE(POCH,'(I1)')PO
       TEMPEV(NDEV+1)='Q'//POCH//'OUT SOME'
       IF (NF(PO).EQ.1) TEMPEV(NDEV+1)='Q'//POCH//'OUT NONE'
       TEMPEV(NDEV+2)='P'//POCH//'OUT SOME'
       IF (NF(PO).EQ.1) TEMPEV(NDEV+2)='P'//POCH//'OUT NONE'
270   NDEV=NDEV+2
      CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
C     ----------------------------------------------------------------
C     Create the IMPOSS state event statement.
C     ----------------------------------------------------------------
      BASICF='S IMPOSS'
      NDEV=0
      DO 280,PO=1,INLET
       WRITE(POCH,'(I1)')PO
       IF (RF(PO).EQ.1) THEN
        TEMPEV(NDEV+1)='G'//POCH//'IN REV'
        TEMPEV(NDEV+2)='R'//POCH//'IN REV'
        NDEV=NDEV+2
       END IF
280   CONTINUE
      DO 290,PO=INLET+1,VES-1
       WRITE(POCH,'(I1)')PO
       IF (RF(PO).EQ.1) THEN
        TEMPEV(NDEV+1)='Q'//POCH//'OUT REV'
        TEMPEV(NDEV+2)='P'//POCH//'OUT REV'
        NDEV=NDEV+2
```

```
      END IF
290   CONTINUE
      IF (NDEV.NE.0) CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
C     -------------------------------------------------------------------
C     Completion of event statements, now check.
C     -------------------------------------------------------------------
      PRINT3,'THE FOLLOWING EVENT STATEMENTS HAVE BEEN CREATED:'
      CALL CHECK(TEMPFI,JL)
C     -------------------------------------------------------------------
C     Take the event statements from the temporary array and
C     place in the FI array.
C     -------------------------------------------------------------------
      K=JL
      JL=0
      DO 300,I=1,K
       IF (TEMPFI(I)(1:1).NE.' ') THEN
       JL=JL+1
       L1=INDEX(TEMPFI(I),' ')
       IF (L1.EQ.0) L1=80
       FI(JL)=TEMPFI(I)(1:L1)
       END IF
300   CONTINUE

      RETURN
      END



      SUBROUTINE TANKTABLE


C     -------------------------------------------------------------------
C     Subroutine to generate the decision tables for the vessel model.
C     -------------------------------------------------------------------
      CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
     *  FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
      INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),INLET,
     *  VES,NF(10),ET(10),RF(10),FLAG(10)

      COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
      COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
      COMMON/TWELVE/LG,INLET,VES,NF,ET,RF,FLAG
```

```fortran
        CHARACTER VESCH,POCH,TEMPDT(25)*80
        INTEGER PO,L1

3       FORMAT(/,' ',A)
C       ------------------------------------------------------------------------
C       Initialize the temporary storage strings.
C       ------------------------------------------------------------------------
        DO 100,I=1,25
100       TEMPDT(I)=DT(I)
        WRITE(VESCH,'(I1)')VES
C       ------------------------------------------------------------------------
C       If liquid is present, then generate the level decision tables.
C       If there is more than one inlet liquid flow ports, string together
C       the NONE deviations as a cause of C(DUMMY).
C       ------------------------------------------------------------------------
        IF (LG(10).NE.1) THEN
        IF (FLAG(4).GT.1) THEN
         DL=DL+1
         DO 200,PO=1,INLET
          IF (LG(PO).EQ.0.AND.NF(PO).EQ.0) THEN
           WRITE(POCH,'(I1)')PO
           L1=LENGTH(TEMPDT(DL))
           IF (L1.EQ.0) THEN
            TEMPDT(DL)='V G'//POCH//' IN NONE'
           ELSE IF (L1.LT.50) THEN
            TEMPDT(DL)=TEMPDT(DL)(1:L1)//' V G'//POCH//' IN NONE'
           ELSE
            TEMPDT(DL)=TEMPDT(DL)(1:L1)//' I D(DUMMY) T C(DUMMY)'
            FLAG(5)=1
            DL=DL+1
            TEMPDT(DL)='V G'//POCH//' IN NONE'
           END IF
          END IF
200      CONTINUE
         L1=LENGTH(TEMPDT(DL))
         IF (FLAG(5).EQ.0) THEN
          TEMPDT(DL)=TEMPDT(DL)(1:L1)//' T C(DUMMY)'
         ELSE
          TEMPDT(DL)=TEMPDT(DL)(1:L1)//' T D(DUMMY)'
         END IF
        END IF
```

```
C        ----------------------------------------------------------------------------
C        For liquid outlet flow ports, generate causes of NO flow and pressure.
C        ----------------------------------------------------------------------------
         DO 210,PO=INLET+1,VES-1
         IF (LG(PO).EQ.0.AND.NF(PO).EQ.0) THEN
         WRITE(POCH,'(I1)')PO
         DL=DL+1
         TEMPDT(DL)='I C(DUMMY) V L'//VESCH//'VES NONE T Q'//POCH//'OUT NONE,P'
     *     //POCH//'OUT NONE'
         END IF
  210    CONTINUE
         END IF
C        ----------------------------------------------------------------------------
C        Generate the decision tables for HIgh and LOw temperature and
C        composition caused by reverse flow through the outlets.
C        ----------------------------------------------------------------------------
         DO 220,PO=INLET+1,VES-1
         IF (RF(PO).EQ.0) THEN
         WRITE(POCH,'(I1)')PO
         TEMPDT(DL+1)='V Q'//POCH//'OUT REV V U'//POCH//'OUT HI T T'//VESCH//'VES HI'
         TEMPDT(DL+2)='V Q'//POCH//'OUT REV V U'//POCH//'OUT LO T T'//VESCH//'VES LO'
         TEMPDT(DL+3)='V Q'//POCH//'OUT REV V Y'//POCH//'OUT HI T X'//VESCH//'VES HI'
         TEMPDT(DL+4)='V Q'//POCH//'OUT REV V Y'//POCH//'OUT LO T X'//VESCH//'VES LO'
         DL=DL+4
         END IF
  220    CONTINUE
C        ----------------------------------------------------------------------------
C        Completion of decision tables, now check.
C        ----------------------------------------------------------------------------
         PRINT3,'THE FOLLOWING DECISION TABLES HAVE BEEN CREATED:'
         CALL CHECK(TEMPDT,DL)
C        ----------------------------------------------------------------------------
C        Take dec. tabs. from the temporary array and place in the DL array.
C        ----------------------------------------------------------------------------
         K=DL
         DL=0
         DO 300,I=1,K
         IF (TEMPDT(I)(1:1).NE.' ') THEN
          DL=DL+1
          L1=INDEX(TEMPDT(I),' ')
          IF (L1.EQ.0) L1=80
```

```
          DT(DL)=TEMPDT(I)(1:L1)
          END IF
300    CONTINUE

       RETURN
       END
```

Appendix D

<u>Heat Exchanger Model Examples</u>

D.1 Partial Reboiler

1)   MODEL   NUMBER   NAME
          200   HEAT EXCHANGER (PARTIAL REBOILER, SHELL & TUBE)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 15
NO. OF PROPAGATION EQUATIONS: 32
NO. OF EVENT STATEMENTS: 28
NO. OF DECISION TABLES: 6
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

SHELL AND TUBE HEAT EXCHANGER WITH COLD FLUID IN THE SHELL SIDE.
ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE TUBE SIDE IS IGNORED.
A PARTIAL PHASE CHANGE TAKES PLACE IN THE SHELL SIDE.
THE TUBE SIDE IS AT A HIGHER PRESSURE THAN THE SHELL SIDE.
TUBE SIDE.
PORT 1: INLET STREAM, NO REVERSE FLOW ALLOWED
PORT 2: OUTLET STREAM, REVERSE FLOW ALLOWED
PORT 3: DRAIN PORT, REVERSE FLOW ALLOWED
SHELL SIDE.
PORT 4: INLET STREAM, PRESSURE EFFECTS, REVERSE FLOW
PORT 5: LIQUID OUTLET STREAM, NO PRESSURE EFFECTS, NO REVERSE FLOW
PORT 6: VAPOUR OUTLET STREAM, PRESSURE EFFECTS, REVERSE FLOW
PORT 7: VESSEL PORT FOR LIQUID IN THE SHELL SIDE
PORT 8: VESSEL PORT FOR VAPOUR IN THE SHELL SIDE
PORT 9: VESSEL PORT FOR LIQUID TO VAPOUR FLOW IN THE SHELL SIDE.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN, Q2OUT)
R1IN= F (R2OUT)
Q2OUT= F (G1IN, G2OUT)
P2OUT= F (P1IN)
T2OUT= F (G1IN, T1IN, -G4IN, T7VES)
XC2OUT= F (XC1IN)
Q3OUT= F (G3OUT, P1IN)
P3OUT= F (P1IN)
T3OUT= F (G1IN, T1IN, -G4IN, T7VES)
XC3OUT= F (XC1IN)

G4IN= F (Q4IN, -P8VES)
R4IN= F (-P8VES)
U4IN= F (T7VES)
YA4IN= F (-XB7VES)
YB4IN= F (XB7VES)
Q5OUT= F (G5OUT)
P5OUT= F (P8VES)
T5OUT= F (T7VES)
XA5OUT= F (-XB7VES)
XB5OUT= F (XB7VES)
Q6OUT= F (G6OUT, P8VES)
P6OUT= F (P8VES)
T6OUT= F (T8VES)
XA6OUT= F (XA8VES)
XB6OUT= F (-XA8VES)
L7VES= F (G4IN, -Q5OUT, -Q9VES)
T7VES= F (P8VES, XB7VES)
XB7VES= F (XB4IN, Q9VES)
P8VES= F (-Q6OUT, L7VES, Q9VES)
T8VES= F (P8VES, -XA8VES)
XA8VES= F (-XB7VES, -Q9VES)
Q9VES= F (G1IN, T1IN, -XB7VES, -P8VES)

4)   EVENT STATEMENTS

V Q5OUT HI: L7VES NONE
V Q3OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, R1IN HI
V Q3OUT SOME: P2OUT LO, P2OUT NONE, P2OUT REV
V G4IN REV: L7VES LO, L7VES NONE, P8VES LO
V Q6OUT REV: P8VES HI
V Q3OUT REV: G1IN LO, G1IN NONE, Q2OUT HI, R1IN LO, R1IN NONE, P2OUT HI
V Q3OUT REV: XD2OUT HI
V L7VES HI: T2OUT LO, Q9VES HI
V L7VES LO: T2OUT HI, Q9VES LO
V L7VES NONE: T2OUT HI, T8VES HI, P8VES LO, Q9VES NONE
V P8VES HI: G4IN NONE, R4IN NONE, G4IN REV, R4IN REV
V P8VES NONE: Q6OUT REV, P6OUT REV
V Q9VES NONE: L7VES HI, P8VES LO
F EXT-HEAT: T2OUT HI, T7VES HI, P8VES HI, T8VES HI
F EXT-COLD: T2OUT LO, T7VES LO, P8VES LO, T8VES LO
F LK-LP-EN: L7VES LO, L7VES NONE, P8VES LO
F INT-LK: G1IN HI, R1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, P2OUT LO
F INT-LK: P2OUT NONE, P2OUT REV
F INT-LK: L7VES HI, P8VES HI
F INT-LK: T2OUT LO, T7VES HI, T8VES HI, XC7VES HI, XC8VES HI
F PART-BLK: G1IN LO, R1IN LO, Q2OUT LO, P2OUT LO
F PART-BLK: T2OUT LO, T7VES LO, T8VES LO

D-2

F FOULING: T2OUT HI, T7VES LO, T8VES LO, Q9VES LO, Q9VES NONE
F FROTHING: T2OUT HI, T7VES LO, T8VES LO, Q9VES LO
F VAP-BLKT: T2OUT HI, T7VES LO, T8VES LO, Q9VES LO
S NORMAL: Q3OUT NONE, G4IN SOME, Q5OUT SOME, Q6OUT SOME, R4IN SOME
S NORMAL: P5OUT SOME, P6OUT SOME
S IMPOSS: G1IN REV, R1IN REV, Q5OUT REV, P5OUT REV, Q9VES REV

5)   DECISION TABLES

V U6OUT HI V Q6OUT REV T T8VES HI
V U6OUT LO V Q6OUT REV T T8VES LO
V YA6OUT HI V Q6OUT REV T XA8VES HI
V YA6OUT LO V Q6OUT REV T XA8VES LO
V YB6OUT HI V Q6OUT REV T XA8VES LO
V YB6OUT LO V Q6OUT REV T XA8VES HI

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 4, 5 AND 6.

MODEL IS MULTI-COMPONENT

D.2 Partial Condenser

1) MODEL NUMBER NAME
    201   HEAT EXCHANGER (PARTIAL CONDENSER, SHELL & TUBE)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 15
NO. OF PROPAGATION EQUATIONS: 34
NO. OF EVENT STATEMENTS: 29
NO. OF DECISION TABLES: 16
NO. OF FAILURE MODES: 1

2) ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

SHELL AND TUBE HEAT EXCHANGER WITH HOT FLUID IN THE SHELL SIDE.
ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE TUBE SIDE IS IGNORED.
A PARTIAL PHASE CHANGE TAKES PLACE IN THE SHELL SIDE.
THE TUBE SIDE IS AT A LOWER PRESSURE THAN THE SHELL SIDE.
TUBE SIDE.
PORT 1: INLET STREAM, REVERSE FLOW ALLOWED
PORT 2: OUTLET STREAM, NO REVERSE FLOW ALLOWED
SHELL SIDE.
PORT 3: INLET STREAM, PRESSURE EFFECTS, REVERSE FLOW
PORT 4: LIQUID OUTLET STREAM, NO PRESSURE EFFECTS, REVERSE FLOW
PORT 5: VAPOUR OUTLET STREAM, PRESSURE EFFECTS, REVERSE FLOW
PORT 6: RELIEF PORT, PRESSURE EFFECTS, REVERSE FLOW
PORT 7: VESSEL PORT FOR LIQUID IN THE SHELL SIDE
PORT 8: VESSEL PORT FOR VAPOUR IN THE SHELL SIDE
PORT 9: VESSEL PORT FOR VAPOUR TO LIQUID FLOW IN THE SHELL SIDE.

3) PROPAGATION EQUATIONS

G1IN= F (Q1IN, Q2OUT)
R1IN= F (R2OUT)
U1IN= F (G3IN, T8VES)
Q2OUT= F (G1IN, G2OUT)
P2OUT= F (P1IN)
T2OUT= F (-G1IN, T1IN, G3IN, T8VES)
XC2OUT= F (XC1IN)
G3IN= F (Q3IN, -P8VES)
R3IN= F (-P8VES)
U3IN= F (T8VES)
YA3IN= F (XA8VES)
YB3IN= F (-XA8VES)
Q4OUT= F (G4OUT)
P4OUT= F (P8VES)
T4OUT= F (T7VES)
XA4OUT= F (-XB7VES)

XB4OUT= F (XB7VES)
Q5OUT= F (G5OUT, P8VES)
P5OUT= F (P8VES)
T5OUT= F (T8VES)
XA5OUT= F (XA8VES)
XB5OUT= F (-XA8VES)
Q6OUT= F (G6OUT, P8VES)
P6OUT= F (P8VES)
T6OUT= F (T8VES)
XA6OUT= F (XA8VES)
XB6OUT= F (-XA8VES)
L7VES= F (-Q4OUT, Q9VES)
T7VES= F (P8VES, XB7VES)
XB7VES= F (-XA8VES, -Q9VES)
P8VES= F (G3IN, -Q5OUT, L7VES, -Q9VES)
T8VES= F (P8VES, -XA8VES)
XA8VES= F (XA3IN, Q9VES)
Q9VES= F (G1IN, -T1IN, -XA8VES, P8VES)

4)   EVENT STATEMENTS

V Q4OUT HI: L7VES NONE
V Q6OUT SOME: P8VES LO
V G3IN REV: P8VES LO
V Q4OUT REV: L7VES HI, P8VES HI
V Q5OUT REV: P7VES HI
V Q6OUT REV: P8VES HI, XD8VES HI
V L7VES HI: T2OUT HI, Q9VES LO
V L7VES LO: T2OUT LO, Q9VES HI
V L7VES NONE: T2OUT LO, P8VES LO, Q9VES HI
V P8VES HI: G3IN NONE, R3IN NONE, G3IN REV, R3IN REV
V P8VES NONE: Q4OUT REV, P4OUT REV, Q5OUT REV, P5OUT REV
V P8VES NONE: Q6OUT REV, P6OUT REV
V Q9VES NONE: L7VES LO, P8VES HI
F EXT-HEAT: U1IN HI, T2OUT HI, T7VES HI, P8VES HI, T8VES HI
F EXT-COLD: U1IN LO, T2OUT LO, T7VES LO, P8VES LO, T8VES LO
F LK-LP-EN: L7VES LO, L7VES NONE, P8VES LO
F INT-LK: G1IN LO, G1IN NONE, G1IN REV, R1IN LO, R1IN NONE, R1IN REV
F INT-LK: Q2OUT HI, P2OUT HI
F INT-LK: L7VES LO, P8VES LO
F INT-LK: U1IN HI, T2OUT HI, T7VES LO, T8VES LO
F INT-LK: YA1IN HI, YB1IN HI, XA2OUT HI, XB2OUT HI
F PART-BLK: G1IN LO, R1IN LO, Q2OUT LO, P2OUT LO
F PART-BLK: U1IN HI, T2OUT HI, T7VES HI, T8VES HI
F FOULING: U1IN LO, T2OUT LO, T7VES HI, T8VES HI, Q9VES LO, Q9VES NONE
F FROTHING: U1IN LO, T2OUT LO, T7VES HI, T8VES HI, Q9VES LO
F VAP-BLKT: U1IN LO, T2OUT LO, T7VES HI, T8VES HI, Q9VES LO

S NORMAL: G3IN SOME, Q4OUT SOME, Q5OUT SOME, Q6OUT NONE, R3IN SOME
S NORMAL: P4OUT SOME, P5OUT SOME
S IMPOSS: Q2OUT REV, P2OUT REV, Q9VES REV

5)   DECISION TABLES

V U4OUT HI V Q4OUT REV T T7VES HI
V U4OUT LO V Q4OUT REV T T7VES LO
V U5OUT HI V Q5OUT REV T T8VES HI
V U5OUT LO V Q5OUT REV T T8VES LO
V U6OUT HI V Q6OUT REV T T8VES HI
V U6OUT LO V Q6OUT REV T T8VES LO
V YA4OUT HI V Q4OUT REV T XB7VES LO
V YA4OUT LO V Q4OUT REV T XB7VES HI
V YB4OUT HI V Q4OUT REV T XB7VES HI
V YB4OUT LO V Q4OUT REV T XB7VES LO
V YA5OUT HI V Q5OUT REV T XA8VES HI
V YA5OUT LO V Q5OUT REV T XA8VES LO
V YB5OUT HI V Q5OUT REV T XA8VES LO
V YB5OUT LO V Q5OUT REV T XA8VES HI
V YD6OUT HI V Q6OUT REV T XD8VES HI
V YD6OUT LO V Q6OUT REV T XD8VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 3, 4 AND 5.

MODEL IS MULTI-COMPONENT

D.3 Total Reboiler

1)   MODEL   NUMBER   NAME

                202   HEAT EXCHANGER (TOTAL REBOILER, FRAME & PLATE)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 15

NO. OF PROPAGATION EQUATIONS: 31

NO. OF EVENT STATEMENTS: 28

NO. OF DECISION TABLES: 12

NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

FRAME AND PLATE HEAT EXCHANGER WITH COLD FLUID IN THE FRAME SIDE.

ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE PLATE SIDE IS IGNORED.

A COMPLETE PHASE CHANGE TAKES PLACE IN THE FRAME SIDE.

THE PLATE SIDE IS AT THE SAME PRESSURE AS THE FRAME SIDE.

PLATE SIDE.

PORT 1: INLET STREAM, REVERSE FLOW ALLOWED

PORT 2: OUTLET STREAM, REVERSE FLOW ALLOWED

PORT 3: RELIEF PORT, REVERSE FLOW ALLOWED

FRAME SIDE.

PORT 4: INLET STREAM, NO PRESSURE EFFECTS, NO REVERSE FLOW

PORT 5: OUTLET STREAM, PRESSURE EFFECTS, REVERSE FLOW

PORT 6: RELIEF PORT, PRESSURE EFFECTS, REVERSE FLOW

PORT 7: DRAIN PORT, PRESSURE EFFECTS, REVERSE FLOW

PORT 8: VESSEL PORT RELATING TO THE FRAME SIDE

PORT 9: VESSEL PORT FOR LIQUID TO VAPOUR FLOW IN THE FRAME SIDE.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN, Q2OUT)

R1IN= F (R2OUT)

U1IN= F (U2OUT, -G4IN, T8VES)

Y1IN= F (Y2OUT)

Q2OUT= F (G1IN, G2OUT)

P2OUT= F (P1IN)

T2OUT= F (G1IN, T1IN, -G4IN, T8VES)

X2OUT= F (X1IN)

Q3OUT= F (G3OUT, P1IN)

P3OUT= F (P1IN)

T3OUT= F (G1IN, T1IN, -G4IN, T8VES)

X3OUT= F (X1IN)

G4IN= F (Q4IN)

R4IN= F (-P8VES)

Q5OUT= F (G5OUT, P8VES)

P5OUT= F (P8VES)

T5OUT= F (T8VES)

X5OUT= F (X8VES)

Q6OUT= F (G6OUT, P8VES)

P6OUT= F (P8VES)

T6OUT= F (T8VES)

X6OUT= F (X8VES)

Q7OUT= F (G7OUT, P8VES)

P7OUT= F (P8VES)

T7OUT= F (T8VES)

X7OUT= F (X8VES)

L8VES= F (G4IN, -Q9VES)

P8VES= F (-Q5OUT, L8VES, Q9VES)

T8VES= F (P8VES, -Q9VES)

X8VES= F (X4IN)

Q9VES= F (G1IN, T1IN, -P8VES)

4)   EVENT STATEMENTS

V Q3OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, R1IN HI

V Q3OUT SOME: P2OUT LO, P2OUT NONE, P2OUT REV

V Q6OUT SOME: P8VES LO

V Q7OUT SOME: L8VES LO, L8VES NONE, P8VES LO

V Q5OUT REV: P8VES HI

V Q3OUT REV: G1IN LO, G1IN NONE, G1IN REV, Q2OUT HI, R1IN LO

V Q3OUT REV: R1IN NONE, R1IN REV, P2OUT HI, Y1IN HI, X2OUT HI

V Q6OUT REV: P8VES HI, X8VES HI

V Q7OUT REV: L8VES HI, P8VES HI, X8VES HI

V L8VES HI: T2OUT LO, Q9VES HI

V L8VES LO: T2OUT HI, Q9VES LO

V L8VES NONE: T2OUT HI, T8VES LO, P8VES HI, Q9VES NONE

V P8VES HI: G4IN NONE, R4IN NONE

V P8VES NONE: Q5OUT REV, P5OUT REV, Q6OUT REV, P6OUT REV

V P8VES NONE: Q7OUT REV, P7OUT REV

V Q9VES NONE: L8VES HI, P8VES LO

F EXT-HEAT: U1IN HI, T2OUT HI, P8VES HI, T8VES HI

F EXT-COLD: U1IN LO, T2OUT LO, P8VES LO, T8VES LO

F LK-LP-EN: L8VES LO, L8VES NONE, P8VES LO

F PART-BLK: G1IN LO, R1IN LO, Q2OUT LO, P2OUT LO

F PART-BLK: G4IN LO, R4IN LO, Q5OUT LO, P5OUT LO

F PART-BLK: U1IN LO, T2OUT LO, T8VES LO

F FOULING: U1IN HI, T2OUT HI, T8VES LO, Q9VES LO, Q9VES NONE

F FROTHING: U1IN HI, T2OUT HI, T8VES LO, Q9VES LO

F VAP-BLKT: U1IN HI, T2OUT HI, T8VES LO, Q9VES LO

S NORMAL: Q3OUT NONE, G4IN SOME, Q5OUT SOME, Q6OUT NONE, Q7OUT NONE

S NORMAL: R4IN SOME, P5OUT SOME

S IMPOSS: G4IN REV, R4IN REV, Q9VES REV

5)   DECISION TABLES

V U5OUT HI V Q5OUT REV T T8VES HI
V U5OUT LO V Q5OUT REV T T8VES LO
V U6OUT HI V Q6OUT REV T T8VES HI
V U6OUT LO V Q6OUT REV T T8VES LO
V U7OUT HI V Q7OUT REV T T8VES HI
V U7OUT LO V Q7OUT REV T T8VES LO
V Y5OUT HI V Q5OUT REV T X8VES HI
V Y5OUT LO V Q5OUT REV T X8VES LO
V Y6OUT HI V Q6OUT REV T X8VES HI
V Y6OUT LO V Q6OUT REV T X8VES LO
V Y7OUT HI V Q7OUT REV T X8VES HI
V Y7OUT LO V Q7OUT REV T X8VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 4 AND 5.

NO MULTI-COMPONENT FEATURES

D.4 Total Condenser

1)   MODEL   NUMBER   NAME

        203   HEAT EXCHANGER (TOTAL CONDENSER, SHELL & TUBE)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 15

NO. OF PROPAGATION EQUATIONS: 31

NO. OF EVENT STATEMENTS: 30

NO. OF DECISION TABLES: 8

NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

SHELL AND TUBE HEAT EXCHANGER WITH HOT FLUID IN THE TUBE SIDE.

ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE SHELL SIDE IS IGNORED.

A COMPLETE PHASE CHANGE TAKES PLACE IN THE TUBE SIDE.

THE SHELL SIDE IS AT A LOWER PRESSURE THAN THE TUBES SIDE.

SHELL SIDE.

PORT 1: INLET STREAM, NO REVERSE FLOW ALLOWED

PORT 2: OUTLET STREAM, NO REVERSE FLOW ALLOWED

PORT 3: DRAIN PORT, REVERSE FLOW ALLOWED

TUBE SIDE.

PORT 4: INLET STREAM, PRESSURE EFFECTS, REVERSE FLOW

PORT 5: OUTLET STREAM, NO PRESSURE EFFECTS, NO REVERSE FLOW

PORT 6: RELIEF PORT, PRESSURE EFFECTS, REVERSE FLOW

PORT 7: DRAIN PORT, PRESSURE EFFECTS, REVERSE FLOW

PORT 8: VESSEL PORT RELATING TO THE TUBES SIDE

PORT 9: VESSEL PORT FOR VAPOUR TO LIQUID FLOW IN THE TUBES SIDE.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN, Q2OUT)

R1IN= F (R2OUT)

Q2OUT= F (G1IN, G2OUT)

P2OUT= F (P1IN)

T2OUT= F (-G1IN, T1IN, G4IN, T8VES)

X2OUT= F (X1IN)

Q3OUT= F (G3OUT, P1IN)

P3OUT= F (P1IN)

T3OUT= F (-G1IN, T1IN, G4IN, T8VES)

X3OUT= F (X1IN)

G4IN= F (Q4IN, -P8VES)

R4IN= F (-P8VES)

U4IN= F (T8VES)

Y4IN= F (X8VES)

Q5OUT= F (G5OUT)

P5OUT= F (P8VES)

T5OUT= F (T8VES)
X5OUT= F (X8VES)
Q6OUT= F (G6OUT, P8VES)
P6OUT= F (P8VES)
T6OUT= F (T8VES)
X6OUT= F (X8VES)
Q7OUT= F (G7OUT, P8VES)
P7OUT= F (P8VES)
T7OUT= F (T8VES)
X7OUT= F (X8VES)
L8VES= F (-Q5OUT, Q9VES)
P8VES= F (G4IN, L8VES, -Q9VES)
T8VES= F (P8VES, -Q9VES)
X8VES= F (X4IN)
Q9VES= F (G1IN, -T1IN, P8VES)

4)   EVENT STATEMENTS

V Q5OUT HI: L8VES NONE
V Q3OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, R1IN HI, P2OUT LO, P2OUT NONE
V Q6OUT SOME: P8VES LO
V Q7OUT SOME: L8VES LO, L8VES NONE, P8VES LO
V G4IN REV: P8VES LO
V Q3OUT REV: G1IN LO, R1IN LO, G1IN NONE, R1IN NONE, Q2OUT HI, P2OUT HI
V Q3OUT REV: X2OUT HI
V Q6OUT REV: P8VES HI, X8VES HI
V Q7OUT REV: L8VES HI, P8VES HI, X8VES HI
V L8VES HI: T2OUT HI, Q9VES LO
V L8VES LO: T2OUT LO, Q9VES HI
V L8VES NONE: T2OUT LO, P8VES LO, Q9VES HI
V P8VES HI: G4IN NONE, R4IN NONE, G4IN REV, R4IN REV
V P8VES NONE: Q6OUT REV, P6OUT REV, Q7OUT REV, P7OUT REV
V Q9VES NONE: L8VES LO, P8VES HI
F EXT-HEAT: T2OUT HI, P8VES HI, T8VES HI
F EXT-COLD: T2OUT LO, P8VES LO, T8VES LO
F LK-LP-EN: G1IN HI, Q2OUT LO, Q2OUT NONE, R1IN HI, P2OUT LO, P2OUT NONE
F INT-LK: G1IN LO, G1IN NONE, R1IN LO, R1IN NONE, Q2OUT HI, P2OUT HI
F INT-LK: L8VES LO, P8VES LO
F INT-LK: T2OUT HI, T8VES LO, X2OUT HI
F PART-BLK: G4IN LO, R4IN LO, Q5OUT LO, P5OUT LO
F PART-BLK: T2OUT HI, T8VES HI
F FOULING: T2OUT LO, T8VES HI, Q9VES LO, Q9VES NONE
F FROTHING: T2OUT LO, T8VES HI, Q9VES LO
F VAP-BLKT: T2OUT LO, T8VES HI, Q9VES LO
S NORMAL: Q3OUT NONE, G4IN SOME, Q5OUT SOME, Q6OUT NONE, Q7OUT NONE
S NORMAL: R4IN SOME, P5OUT SOME
S IMPOSS: G1IN REV, R1IN REV, Q2OUT REV, P2OUT REV, Q5OUT REV

D-11

S IMPOSS: P5OUT REV, Q9VES REV

5)   DECISION TABLES

V U6OUT HI V Q6OUT REV T T8VES HI
V U6OUT LO V Q6OUT REV T T8VES LO
V U7OUT HI V Q7OUT REV T T8VES HI
V U7OUT LO V Q7OUT REV T T8VES LO
V Y6OUT HI V Q6OUT REV T X8VES HI
V Y6OUT LO V Q6OUT REV T X8VES LO
V Y7OUT HI V Q7OUT REV T X8VES HI
V Y7OUT LO V Q7OUT REV T X8VES LO

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 4 AND 5.

NO MULTI-COMPONENT FEATURES

D.5 Heater Unit

1)   MODEL   NUMBER   NAME
            204   HEAT EXCHANGER (HEATER, SHELL & TUBE)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 14
NO. OF PROPAGATION EQUATIONS: 28
NO. OF EVENT STATEMENTS: 29
NO. OF DECISION TABLES: 0
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

SHELL AND TUBE HEAT EXCHANGER WITH COLD FLUID IN THE SHELL SIDE.
ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE TUBE SIDE IS IGNORED.
NO PHASE CHANGE TAKES PLACE IN THE SHELL SIDE.
THE TUBE SIDE IS AT A HIGHER PRESSURE THAN THE SHELL SIDE.
TUBE SIDE.
PORT 1: INLET STREAM, NO REVERSE FLOW ALLOWED
PORT 2: OUTLET STREAM, REVERSE FLOW ALLOWED
PORT 3: RELIEF PORT, REVERSE FLOW ALLOWED
PORT 4: DRAIN PORT, REVERSE FLOW ALLOWED
SHELL SIDE.
PORT 5: INLET STREAM, NO PRESSURE EFFECTS, NO REVERSE FLOW
PORT 6: OUTLET STREAM, NO PRESSURE EFFECTS, REVERSE FLOW
PORT 7: RELIEF PORT, PRESSURE EFFECTS, REVERSE FLOW
PORT 8: DRAIN PORT, PRESSURE EFFECTS, REVERSE FLOW

3)   PROPAGATION EQUATIONS

$G1IN = F (Q1IN, Q2OUT)$
$R1IN = F (R2OUT)$
$Q2OUT = F (G1IN, G2OUT)$
$P2OUT = F (P1IN)$
$T2OUT = F (G1IN, T1IN, -G5IN, T5IN)$
$X2OUT = F (X1IN)$
$Q3OUT = F (G3OUT, P1IN)$
$P3OUT = F (P1IN)$
$T3OUT = F (G1IN, T1IN, -G5IN, T5IN)$
$X3OUT = F (X1IN)$
$Q4OUT = F (G4OUT, P1IN)$
$P4OUT = F (P1IN)$
$T4OUT = F (G1IN, T1IN, -G5IN, T5IN)$
$X4OUT = F (X1IN)$
$G5IN = F (Q5IN, Q6OUT)$
$R5IN = F (R6OUT)$
$Q6OUT = F (G5IN, G6OUT)$

P6OUT= F (P5IN)
T6OUT= F (G1IN, T1IN, -G5IN, T5IN)
X6OUT= F (X5IN)
Q7OUT= F (G7OUT, P5IN)
P7OUT= F (P5IN)
T7OUT= F (G1IN, T1IN, -G5IN, T5IN)
X7OUT= F (X5IN)
Q8OUT= F (G8OUT, P5IN)
P8OUT= F (P5IN)
T8OUT= F (G1IN, T1IN, -G5IN, T5IN)
X8OUT= F (X5IN)

4)   EVENT STATEMENTS

V Q3OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, R1IN HI
V Q3OUT SOME: P2OUT LO, P2OUT NONE, P2OUT REV
V Q4OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, R1IN HI
V Q4OUT SOME: P2OUT LO, P2OUT NONE, P2OUT REV
V Q7OUT SOME: G5IN HI, Q6OUT LO, Q6OUT NONE, Q6OUT REV, R5IN HI
V Q7OUT SOME: P6OUT LO, P6OUT NONE, P6OUT REV
V Q8OUT SOME: G5IN HI, Q6OUT LO, Q6OUT NONE, Q6OUT REV, R5IN HI
V Q8OUT SOME: P6OUT LO, P6OUT NONE, P6OUT REV
V Q3OUT REV: G1IN LO, G1IN NONE, Q2OUT HI, R1IN LO, R1IN NONE, P2OUT HI
V Q3OUT REV: X2OUT HI
V Q4OUT REV: G1IN LO, G1IN NONE, Q2OUT HI, R1IN LO, R1IN NONE, P2OUT HI
V Q4OUT REV: X2OUT HI
V Q7OUT REV: G5IN LO, G5IN NONE, Q6OUT HI, R5IN LO, R5IN NONE, P6OUT HI
V Q7OUT REV: X6OUT HI
V Q8OUT REV: G5IN LO, G5IN NONE, Q6OUT HI, R5IN LO, R5IN NONE, P6OUT HI
V Q8OUT REV: X6OUT HI
F EXT-HEAT: T2OUT HI, T6OUT HI
F EXT-COLD: T2OUT LO, T6OUT LO
F LK-LP-EN: G5IN HI, R5IN HI, Q6OUT LO, Q6OUT NONE
F LK-LP-EN: Q6OUT REV, P6OUT LO, P6OUT NONE, P6OUT REV
F INT-LK: G1IN HI, R1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, P2OUT LO
F INT-LK: P2OUT NONE, P2OUT REV
F INT-LK: G5IN LO, G5IN NONE, R5IN LO, R5IN NONE, Q6OUT HI, P6OUT HI
F INT-LK: T2OUT LO, T6OUT HI, X6OUT HI
F PART-BLK: G1IN LO, R1IN LO, Q2OUT LO, P2OUT LO
F PART-BLK: T2OUT LO, T6OUT LO
F FOULING: T2OUT HI, T6OUT LO
S NORMAL: Q3OUT NONE, Q4OUT NONE, Q7OUT NONE, Q8OUT NONE
S IMPOSS: G1IN REV, R1IN REV, G5IN REV, R5IN REV

5)   DECISION TABLES

N/A

D-14

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 5 AND 6.

NO MULTI-COMPONENT FEATURES

D.6 Cooler Unit

1)   MODEL   NUMBER   NAME
               205   HEAT EXCHANGER (COOLER, SHELL & TUBE)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 14
NO. OF PROPAGATION EQUATIONS: 29
NO. OF EVENT STATEMENTS: 28
NO. OF DECISION TABLES: 0
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

SHELL AND TUBE HEAT EXCHANGER WITH HOT FLUID IN THE SHELL SIDE.
ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE TUBE SIDE IS IGNORED.
NO PHASE CHANGE TAKES PLACE IN THE SHELL SIDE.
THE TUBE SIDE IS AT A LOWER PRESSURE THAN THE SHELL SIDE.
TUBE SIDE.
PORT 1: INLET STREAM, NO REVERSE FLOW ALLOWED
PORT 2: OUTLET STREAM, REVERSE FLOW ALLOWED
PORT 3: RELIEF PORT, REVERSE FLOW ALLOWED
PORT 4: DRAIN PORT, REVERSE FLOW ALLOWED
SHELL SIDE.
PORT 5: INLET STREAM, NO PRESSURE EFFECTS, REVERSE FLOW
PORT 6: OUTLET STREAM, NO PRESSURE EFFECTS, NO REVERSE FLOW
PORT 7: RELIEF PORT, PRESSURE EFFECTS, REVERSE FLOW
PORT 8: DRAIN PORT, PRESSURE EFFECTS, REVERSE FLOW.

3)   PROPAGATION EQUATIONS

G1IN= F (Q1IN, Q2OUT)
R1IN= F (R2OUT)
Q2OUT= F (G1IN, G2OUT)
P2OUT= F (P1IN)
T2OUT= F (-G1IN, T1IN, G5IN, T5IN)
X2OUT= F (X1IN)
Q3OUT= F (G3OUT, P1IN)
P3OUT= F (P1IN)
T3OUT= F (-G1IN, T1IN, G5IN, T5IN)
X3OUT= F (X1IN)
Q4OUT= F (G4OUT, P1IN)
P4OUT= F (P1IN)
T4OUT= F (-G1IN, T1IN, G5IN, T5IN)
X4OUT= F (X1IN)
G5IN= F (Q5IN, Q6OUT)
R5IN= F (R6OUT)
U5IN= F (-G1IN, T1IN)

Q6OUT= F (G5IN, G6OUT)
P6OUT= F (P5IN)
T6OUT= F (-G1IN, T1IN, G5IN, T5IN)
X6OUT= F (X5IN)
Q7OUT= F (G7OUT, P5IN)
P7OUT= F (P5IN)
T7OUT= F (-G1IN, T1IN, G5IN, T5IN)
X7OUT= F (X5IN)
Q8OUT= F (G8OUT, P5IN)
P8OUT= F (P5IN)
T8OUT= F (-G1IN, T1IN, G5IN, T5IN)
X8OUT= F (X5IN)

4)   EVENT STATEMENTS

V Q3OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, R1IN HI
V Q3OUT SOME: P2OUT LO, P2OUT NONE, P2OUT REV
V Q4OUT SOME: G1IN HI, Q2OUT LO, Q2OUT NONE, Q2OUT REV, R1IN HI
V Q4OUT SOME: P2OUT LO, P2OUT NONE, P2OUT REV
V Q7OUT SOME: G5IN HI, Q6OUT LO, Q6OUT NONE, R5IN HI, P6OUT LO, P6OUT NONE
V Q8OUT SOME: G5IN HI, Q6OUT LO, Q6OUT NONE, R5IN HI, P6OUT LO, P6OUT NONE
V Q3OUT REV: G1IN LO, G1IN NONE, Q2OUT HI, R1IN LO, R1IN NONE, P2OUT HI
V Q3OUT REV: X2OUT HI
V Q4OUT REV: G1IN LO, G1IN NONE, Q2OUT HI, R1IN LO, R1IN NONE, P2OUT HI
V Q4OUT REV: X2OUT HI
V Q7OUT REV: G5IN LO, G5IN NONE, G5IN REV, Q6OUT HI, R5IN LO, R5IN NONE
V Q7OUT REV: R5IN REV, P6OUT HI, Y5IN HI, X6OUT HI
V Q8OUT REV: G5IN LO, G5IN NONE, G5IN REV, Q6OUT HI, R5IN LO, R5IN NONE
V Q8OUT REV: R5IN REV, P6OUT HI, Y5IN HI, X6OUT HI
F EXT-HEAT: T2OUT HI, U5IN HI, T6OUT HI
F EXT-COLD: T2OUT LO, U5IN LO, T6OUT LO
F LK-LP-EN: G5IN HI, R5IN HI, Q6OUT LO, Q6OUT NONE, P6OUT LO, P6OUT NONE
F INT-LK: G1IN LO, G1IN NONE, R1IN LO, R1IN NONE, Q2OUT HI, P2OUT HI
F INT-LK: G5IN HI, R5IN HI, Q6OUT LO, Q6OUT NONE, P6OUT LO, P6OUT NONE
F INT-LK: T2OUT HI, T6OUT LO, X2OUT HI
F PART-BLK: G1IN LO, R1IN LO, Q2OUT LO, P2OUT LO
F PART-BLK: T2OUT HI, U5IN HI, T6OUT HI
F FOULING: T2OUT LO, U5IN HI, T6OUT HI
S NORMAL: Q3OUT NONE, Q4OUT NONE, Q7OUT NONE, Q8OUT NONE
S IMPOSS: G1IN REV, R1IN REV, Q6OUT REV, P6OUT REV

5)   DECISION TABLES

N/A

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: FLOW IS NORMAL THROUGH PORTS 1, 2, 5 AND 6.

NO MULTI-COMPONENT FEATURES

D.7 Heat Exchanger Model Generation Subroutine

      SUBROUTINE HEXCHMOD

```
C      ----------------------------------------------------------------------------------
C      This subroutine is called if the automatic heat exchanger model
C      generation is specified. It automatically generates the propagation
C      equations, event statements and decision tables for the model.
C
C      N.B. Use is made of common block TWELVE from the vessel model generation
C      routines, the variables INLET and VES have been renamed PHASE and GRAD.
C      ----------------------------------------------------------------------------------
       CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
     *  FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
       INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),PHASE,
     *  GRAD,NF(10),ET(10),RF(10),FLAG(10)

       COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
       COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
       COMMON/TWELVE/LG,PHASE,GRAD,NF,ET,RF,FLAG

       CHARACTER OPT*4,VESCH(3),POCH(5),PCH,STRING*70,STG1*7,STG2*7
       INTEGER ANSWER,L1

1      FORMAT(/,' ',A,' ? ',$)
2      FORMAT(' ',A,' ? ',$)
3      FORMAT(/,' ',A)
4      FORMAT(' ',A,I2,') ? ',$)
5      FORMAT(/,' ',A,' ',I2)
6      FORMAT(/,' ',A,A)
7      FORMAT(' ',A,A,A,A,A,A,A,A,') ? ',$)
C      ----------------------------------------------------------------------------------
C      The FLAGS set by this subroutine are:
C      FLAG(1)=1 for no checking of event statements from the GENEV routine.
C      FLAG(2) is used by the vessel model generation routine.
C      FLAG(3) set to 1 for a first side relief or drain port,
C              set to 2 for first side relief and drain ports.
C      FLAG(4) set to 1 if extra first side port is to relief,
C              set to 2 if extra first side port is to drain.
C      FLAG(5) set to 1 for a second side relief or drain port,
C              set to 2 for second side relief and drain ports.
```

```
C        FLAG(6) set to 1 if extra second side port is to relief,
C                set to 2 if extra second side port is to drain.
C        FLAG(7) set to the last flow port on the second side.
C        FLAG(8) the total number of ports use including vessel ports.
C        FLAG(9) set to 0 for outer side is second side,
C                set to 1 for outer side is first side.
C        FLAG(10) set to 0 for shell and tube exchanger,
C                 set to 1 for frame and plate exchanger.
C        ----------------------------------------------------------------
         FLAG(1)=1
         MC='N'
30       PRINT3,'         HEAT EXCHANGER MODEL GENERATION ROUTINE'
         PRINT*,'       ----------------------------------------------------------------'
         PRINT3,'QUESTIONS REFER TO NORMAL OPERATING CONDITIONS.'
         PRINT*,'NO DISTINCTION IS MADE BETWEEN COUNTER-CURRENT AND'
     *   // CROSS-CURRENT FLOW.'
C        ----------------------------------------------------------------
C        Determine the type of heat exchanger to be modelled.
C        Set FLAG(10) to 0 for shell/tube or 1 for frame/plate.
C        Set up the type strings, STG1 for the first side or no phase change and
C        STG2 for the second where a phase change is possible
C        ----------------------------------------------------------------
         PRINT1,'IS THIS A SHELL/TUBE (S) OR FRAME/PLATE (F) HEAT EXCHANGER, S/F'
         OPT='SF'
         FLAG(10)=ANSWER(OPT)
         STG1=' TUBES'
         STG2=' SHELL'
         IF (FLAG(10).EQ.1) THEN
          STG1=' PLATE'
          STG2=' FRAME'
         END IF
C        ----------------------------------------------------------------
C        Print information about phase changes.
C        ----------------------------------------------------------------
         PRINT3,'PHASE CHANGES CAN ONLY BE HANDLED ON ONE SIDE OF THE HEAT EXCHANGER,'
         PRINT*,'IF A PHASE CHANGE TAKES PLACE ON BOTH SIDES CHOOSE THE MORE IMPORTANT'
         PRINT*,'ONE AND THE OTHER WILL BE IGNORED.'
         PRINT3,'IF A PARTIAL PHASE CHANGE OCCURS WHERE THERE IS MORE THAN ONE'
     *   // COMPONENT'
         PRINT*,'IT IS NECESSARY TO SPECIFY MULTI-COMPONENT FEATURES, IN ALL OTHER CASES'
         PRINT*,'THEY ARE NOT REQUIRED.'
```

```
C      ----------------------------------------------------------------
C      Determine if a phase change takes place.  PHASE=0 for no phase change,
C      1 for complete phase change or 2 for a partial phase change.
C      ----------------------------------------------------------------
       PRINT1,'IS THERE A COMPLETE OR PARTIAL PHASE CHANGE OR NO PHASE CHANGE, C/P/N'
       OPT='NCP'
       PHASE=ANSWER(OPT)
C      ----------------------------------------------------------------
C      If there is a phase change determine in which side it occurs.
C      Also set FLAG(9) to 1 to indicate that the side where a leak may
C      occur is the first side if this is the case.
C      ----------------------------------------------------------------
       IF (PHASE.NE.0) THEN
       PRINT7,'DOES THE PHASE CHANGE OCCUR IN THE',STG2,'SIDE (',STG2(2:2),') OR THE',
     * STG1,'SIDE (',STG1(2:2)
       OPT=STG2(2:2)//STG1(2:2)
       IF (ANSWER(OPT).EQ.1) THEN
       FLAG(9)=1
       STG1=STG2
       STG2=' TUBES'
       IF (FLAG(10).EQ.1) STG2=' PLATE'
       END IF
C      ----------------------------------------------------------------
C      If there is a partial phase change determine if there is more than
C      one component.  Change MC from 'N' to 'Y' accordingly.
C      ----------------------------------------------------------------
       IF (PHASE.EQ.2) THEN
       PRINT2,'IS THERE MORE THAN ONE COMPONENT IN THIS SIDE'
       OPT='NY'
       IF (ANSWER(OPT).EQ.1) MC='Y'
       END IF
       END IF
C      ----------------------------------------------------------------
C      Determine the first side characteristics, no phase change occurs.
C      ----------------------------------------------------------------
       PRINT6,STG1,'SIDE CHARACTERISTICS'
       PRINT*,'----------------------------------'
C      ----------------------------------------------------------------
C      Notify user of the first side assumptions.
C      ----------------------------------------------------------------
```

```
        PRINT3,'IT IS ASSUMED THERE IS ONLY ONE INLET AND ONE OUTLET FLOW PORT AND'
        PRINT*,'ANY PRESSURE EFFECTS ARE NEGLIGABLE.'
C       -----------------------------------------------------------------------------------------------
C       Determine if there is a first side relief port.
C       -----------------------------------------------------------------------------------------------
        PRINT1,'IS THERE A RELIEF PORT FOR THIS SIDE, Y/N'
        OPT='NY'
        I=ANSWER(OPT)
C       -----------------------------------------------------------------------------------------------
C       Determine if there is a first side drain port.
C       -----------------------------------------------------------------------------------------------
        PRINT2,'IS THERE A DRAIN PORT FOR THIS SIDE, Y/N'
        OPT='NY'
        J=ANSWER(OPT)
C       -----------------------------------------------------------------------------------------------
C       Set FLAG(3) to 1 for a single relief or drain port, 2 for both.
C       Set FLAG(4) to 1 if extra port is to relief or 2 if to drain.
C       -----------------------------------------------------------------------------------------------
        FLAG(3)=I+J
        IF (FLAG(3).EQ.1) THEN
         FLAG(4)=1
         IF (J.EQ.1) FLAG(4)=2
        END IF
C       -----------------------------------------------------------------------------------------------
C       Set NF(port number) for the normal flow characteristics for
C       the relief/drain port(s) to 1 for normally no flow.  The array
C       was zeroed at the start of the program for the other ports.
C       -----------------------------------------------------------------------------------------------
        IF (FLAG(3).NE.0) NF(3)=1
        IF (FLAG(3).EQ.2) NF(4)=1
C       -----------------------------------------------------------------------------------------------
C       Set ET(port number) for the flow characteristics for the flow
C       ports to 1 for no pressure effects.  If there is a relief/drain
C       port(s) there will be pressure effects so set to 0.
C       -----------------------------------------------------------------------------------------------
        ET(1)=1
        ET(2)=1
        IF (FLAG(3).NE.0) ET(3)=0
        IF (FLAG(3).EQ.2) ET(4)=0
```

```
C     ----------------------------------------------------------------
C     Determine if reverse flow is to be allowed, set RF(port number)
C     to 1 for no reverse flow.  Any relief/drain port(s) is left as 0.
C     ----------------------------------------------------------------
      PRINT1,'IS REVERSE FLOW TO BE ALLOWED FOR THE INLET, Y/N'
      OPT='YN'
      RF(1)=ANSWER(OPT)
      PRINT2,'IS REVERSE FLOW TO BE ALLOWED FOR THE OUTLET, Y/N'
      OPT='YN'
      RF(2)=ANSWER(OPT)
C     ----------------------------------------------------------------
C     Determine the temperature gradient in the heat exchanger.
C     GRAD is 0 if this is the hot side or 1 for the cold side.
C     ----------------------------------------------------------------
      PRINT1,'IS THIS SIDE FOR THE HOT OR COLD FLUID, H/C'
      OPT='HC'
      GRAD=ANSWER(OPT)
C     ----------------------------------------------------------------
C     Set NF(10) to 0, for the same, 1 for higher or 2 for lower, for
C     the pressure difference between the first side and the second.
C     ----------------------------------------------------------------
      PRINT2,'IS THIS SIDE AT THE SAME, HIGHER OR LOWER PRESSURE THEN THE OTHER, S/H/L'
      OPT='SHL'
      NF(10)=ANSWER(OPT)
C     ----------------------------------------------------------------
C     Determine the second side characteristics, a phase change is
C     possible
C     ----------------------------------------------------------------
      PRINT6,STG2,'SIDE CHARACTERISTICS'
      PRINT*,' -----------------------------------'
C     ----------------------------------------------------------------
C     If there is no phase change then pressure effects are negligable.
C     ----------------------------------------------------------------
      IF (PHASE.EQ.0) THEN
       PRINT3,'PRESSURE EFFECTS ARE NEGLIGABLE AND WILL BE IGNORED.'
      END IF
C     ----------------------------------------------------------------
C     Determine if there is a second side relief port.
C     ----------------------------------------------------------------
```

```
      PRINT1,'IS THERE A RELIEF PORT FOR THIS SIDE, Y/N'
      OPT='NY'
      I=ANSWER(OPT)
C     ----------------------------------------------------------------
C     Determine if there is a second side drain port.
C     ----------------------------------------------------------------
      PRINT2,'IS THERE A DRAIN PORT FOR THIS SIDE, Y/N'
      OPT='NY'
      J=ANSWER(OPT)
C     ----------------------------------------------------------------
C     Set FLAG(5) to 1 for a relief or drain port, 2 for both.
C     Set FLAG(6) to 1 if extra port is to relief or 2 if to drain.
C     ----------------------------------------------------------------
      FLAG(5)=I+J
      IF (FLAG(5).EQ.1) THEN
       FLAG(6)=1
       IF (J.EQ.1) FLAG(6)=2
      END IF
C     ----------------------------------------------------------------
C     Set FLAG(7) to the last flow port on the second side.
C     Calculate the total number of ports FLAG(8).  This is the sum of the two
C     first side ports plus any relief/drain port(s), plus the two second side
C     ports plus any relief/drain port(s), if there is a complete phase change
C     then two vessel ports are required if and a partial phase change occurs
C     then an additional flow port is required and three vessel ports.
C     ----------------------------------------------------------------
      FLAG(7)=4+FLAG(3)+FLAG(5)
      FLAG(8)=FLAG(7)
      IF (PHASE.EQ.1) FLAG(8)=FLAG(8)+2
      IF (PHASE.EQ.2) THEN
       FLAG(7)=FLAG(7)+1
       FLAG(8)=FLAG(8)+4
      END IF
C     ----------------------------------------------------------------
C     Check number of ports, if it exceeds 9 report the error and start again.
C     ----------------------------------------------------------------
      IF (FLAG(8).GT.9) THEN
       PRINT5,'THE TOTAL NUMBER OF PORTS EXCEEDS 9 BY ',FLAG(8)-9
       PRINT*,'IT IS NOT POSSIBLE TO MODEL THIS PIECE OF EQUIPMENT ON FAULTFINDER.'
```

```
      PRINT*,'THE SPECIFICATION WILL HAVE TO BE CHANGED.'
      GOTO 30
      END IF
C     -------------------------------------------------------------------
C     Set NF(port number) for the normal flow characteristics for
C     the relief/drain port(s) to 1 for normally no flow.
C     -------------------------------------------------------------------
      IF (FLAG(5).NE.0) NF(FLAG(7))=1
      IF (FLAG(5).EQ.2) NF(FLAG(7)-1)=1
C     -------------------------------------------------------------------
C     If there is no phase change set ET(port number) to 1.
C     -------------------------------------------------------------------
      IF (PHASE.EQ.0) THEN
      ET(FLAG(3)+3)=1
      ET(FLAG(3)+4)=1
C     -------------------------------------------------------------------
C     For no phase change, determine if reverse flow is to be allowed,
C     set RF(port number) to 1 for the no reverse flow.
C     -------------------------------------------------------------------
      PRINT1,'IS REVERSE FLOW TO BE ALLOWED FOR THE INLET, Y/N'
      OPT='YN'
      RF(FLAG(3)+3)=ANSWER(OPT)
      PRINT2,'IS REVERSE FLOW TO BE ALLOWED FOR THE OUTLET, Y/N'
      OPT='YN'
      RF(FLAG(3)+4)=ANSWER(OPT)
      ELSE
C     -------------------------------------------------------------------
C     For a phase change, determine if the flow is pumped for each
C     second side flow port.
C     -------------------------------------------------------------------
      K=FLAG(7)-FLAG(5)
      J=FLAG(3)+3
      DO 50,I=J,K
       IF (I.EQ.J) THEN
        STRING='INLET'
       ELSE
        IF (PHASE.NE.2) THEN
         STRING='OUTLET'
        ELSE
         IF (I.NE.K) THEN
          STRING='LIQUID OUTLET'
```

D-25

```
        ELSE
          STRING='VAPOUR OUTLET'
         END IF
        END IF
       END IF
       L1=LENGTH(STRING)
       STRING=STRING(1:L1)//' PORT : IS FLOW POSITIVE DISPLACEMENT OR'
       PRINT3,STRING
       PRINT2,'CENTRIFUGALLY PUMPED OR BACK PRESSURE FED, P/C/G'
       OPT='GCP'
       ET(I)=ANSWER(OPT)
C      ------------------------------------------------------------------------------
C      For a positive displacement pumped flow determine if reverse flow is
C      allowed.  Set RF(port number) to 1 for no.
C      All other port types allow reverse flow.
C      ------------------------------------------------------------------------------
       IF (ET(I).EQ.2) THEN
        PRINT2,'IS REVERSE FLOW TO BE ALLOWED IN THIS CONNECTION, Y/N'
        OPT='YN'
        RF(I)=ANSWER(OPT)
       END IF
C      ------------------------------------------------------------------------------
C      If the inlet flow is centrifugally pumped determine if it is
C      high pressure which implies no pressure effects.
C      ------------------------------------------------------------------------------
       IF (I.EQ.J.AND.ET(I).EQ.1) THEN
        PRINT2,'IS THE PUMP A HIGH PRESSURE PUMP, Y/N'
        OPT='NY'
        ET(I)=ANSWER(OPT)
       END IF
C      ------------------------------------------------------------------------------
C      If flow is positive displacement pumped set ET(port number)
C      to 1 for no pressure effects.
C      ------------------------------------------------------------------------------
50     IF (ET(I).EQ.2) ET(I)=1
       END IF
C      ------------------------------------------------------------------------------
C      For any relief/drain port(s), ET and RF were zeroed at the
C      start of the program.  Set RF to 1 for the vessel ports.
C      ------------------------------------------------------------------------------
```

```
        DO 60,I=FLAG(7)+1,FLAG(8)
60      RF(I)=1
C       -------------------------------------------------------------------------------
C       Generate the assumptions and description array for the heat exchanger.
C       -------------------------------------------------------------------------------
        AD(1)='SHELL AND TUBE'
        IF (FLAG(10).EQ.1) AD(1)='FRAME AND PLATE'
        L1=LENGTH(AD(1))
        AD(1)=AD(1)(1:L1)//' HEAT EXCHANGER WITH'
        L1=LENGTH(AD(1))
        IF (GRAD.EQ.0) THEN
         AD(1)=AD(1)(1:L1)//' COLD'
        ELSE
         AD(1)=AD(1)(1:L1)//' HOT'
        END IF
        L1=LENGTH(AD(1))
        AD(1)=AD(1)(1:L1)//' FLUID IN THE'//STG2//'SIDE.'
        AD(2)='ANY PHASE CHANGE OR PRESSURE EFFECTS IN THE'//STG1//'SIDE IS IGNORED.'
        IF (PHASE.EQ.0) THEN
         AD(3)='NO'
        ELSE IF (PHASE.EQ.1) THEN
         AD(3)='A COMPLETE'
        ELSE
         AD(3)='A PARTIAL'
        END IF
        L1=LENGTH(AD(3))
        AD(3)=AD(3)(1:L1)//' PHASE CHANGE TAKES PLACE IN THE'//STG2//'SIDE.'
        AD(4)='THE'//STG1//'SIDE IS AT'
        IF (NF(10).EQ.0) THEN
         AD(4)=AD(4)(1:20)//' THE SAME PRESSURE AS'
        ELSE IF (NF(10).EQ.1) THEN
         AD(4)=AD(4)(1:20)//' A HIGHER PRESSURE THAN'
        ELSE
         AD(4)=AD(4)(1:20)//' A LOWER PRESSURE THAN'
        END IF
        L1=LENGTH(AD(4))
        AD(4)=AD(4)(1:L1)//' THE'//STG2//'SIDE.'
C       -------------------------------------------------------------------------------
C       Generate the first side characteristics.
C       -------------------------------------------------------------------------------
```

```fortran
      AD(5)=STG1//'SIDE.'
      AD(5)=AD(5)(2:)
      STRING='REVERSE FLOW ALLOWED.'
      IF (RF(1).EQ.1) STRING='NO '//STRING
      AD(6)='PORT 1: INLET STREAM, '//STRING
      STRING='REVERSE FLOW ALLOWED.'
      IF (RF(2).EQ.1) STRING='NO '//STRING
      AD(7)='PORT 2: OUTLET STREAM, '//STRING
      IL=7
      IF (FLAG(3).NE.0) THEN
       IL=IL+1
       IF (FLAG(3).EQ.2.OR.FLAG(4).EQ.1) THEN
        AD(IL)='PORT 3: RELIEF'
       ELSE
        AD(IL)='PORT 3: DRAIN'
       END IF
       L1=LENGTH(AD(IL))
       AD(IL)=AD(IL)(1:L1)//' PORT, REVERSE FLOW ALLOWED.'
       IF (FLAG(3).EQ.2) THEN
        IL=IL+1
        AD(IL)='PORT 4: DRAIN PORT, REVERSE FLOW ALLOWED.'
       END IF
      END IF
C     --------------------------------------------------------------------------------
C     Generate the second side characteristics.
C     --------------------------------------------------------------------------------
      IL=IL+1
      AD(IL)=STG2//'SIDE.'
      AD(IL)=AD(IL)(2:)
      IL=IL+1
      WRITE(POCH(1),'(I1)')FLAG(3)+3
      AD(IL)='PORT '//POCH(1)//': INLET STREAM,'
      IF (ET(FLAG(3)+3).EQ.1) AD(IL)=AD(IL)(1:21)//' NO'
      L1=LENGTH(AD(IL))
      AD(IL)=AD(IL)(1:L1)//' PRESSURE EFFECTS,'
      L1=LENGTH(AD(IL))
      IF (RF(FLAG(3)+3).EQ.1) AD(IL)=AD(IL)(1:L1)//' NO'
      L1=LENGTH(AD(IL))
      AD(IL)=AD(IL)(1:L1)//' REVERSE FLOW.'
      IL=IL+1
      WRITE(POCH(2),'(I1)')FLAG(3)+4
```

D-28

```
IF (PHASE.NE.2) THEN
 AD(IL)='PORT '//POCH(2)//': OUTLET STREAM,'
ELSE
 AD(IL)='PORT '//POCH(2)//': LIQUID OUTLET STREAM,'
END IF
L1=LENGTH(AD(IL))
IF (ET(FLAG(3)+4).EQ.1) AD(IL)=AD(IL)(1:L1)//' NO'
L1=LENGTH(AD(IL))
AD(IL)=AD(IL)(1:L1)//' PRESSURE EFFECTS,'
L1=LENGTH(AD(IL))
IF (RF(FLAG(3)+4).EQ.1) AD(IL)=AD(IL)(1:L1)//' NO'
L1=LENGTH(AD(IL))
AD(IL)=AD(IL)(1:L1)//' REVERSE FLOW.'
IF (PHASE.EQ.2) THEN
 IL=IL+1
 WRITE(POCH(3),'(I1)')FLAG(3)+5
 AD(IL)='PORT '//POCH(3)//': VAPOUR OUTLET STREAM,'
 IF (ET(FLAG(3)+5).EQ.1) AD(IL)=AD(IL)(1:29)//' NO'
 L1=LENGTH(AD(IL))
 AD(IL)=AD(IL)(1:L1)//' PRESSURE EFFECTS,'
 L1=LENGTH(AD(IL))
 IF (RF(FLAG(3)+5).EQ.1) AD(IL)=AD(IL)(1:L1)//' NO'
 L1=LENGTH(AD(IL))
 AD(IL)=AD(IL)(1:L1)//' REVERSE FLOW.'
END IF
IF (FLAG(5).NE.0) THEN
 IL=IL+1
 IF (FLAG(5).EQ.1) THEN
  WRITE(POCH(4),'(I1)')FLAG(7)
 ELSE
  WRITE(POCH(4),'(I1)')FLAG(7)-1
 END IF
 WRITE(POCH(5),'(I1)')FLAG(7)
 IF (FLAG(5).EQ.2.OR.FLAG(6).EQ.1) THEN
  AD(IL)='PORT '//POCH(4)//': RELIEF'
 ELSE
  AD(IL)='PORT '//POCH(5)//': DRAIN'
 END IF
 L1=LENGTH(AD(IL))
 AD(IL)=AD(IL)(1:L1)//' PORT, PRESSURE EFFECTS, REVERSE FLOW.'
 IF (FLAG(5).EQ.2) THEN
```

```fortran
      IL=IL+1
      AD(IL)='PORT '//POCH(5)//': DRAIN PORT, PRESSURE EFFECTS, REVERSE FLOW.'
      END IF
      END IF
C     ----------------------------------------------------------------------
C     Generate the vessel port characteristics.
C     ----------------------------------------------------------------------
      IF (PHASE.EQ.1) THEN
      IL=IL+1
      WRITE(VESCH(1),'(I1)')FLAG(7)+1
      WRITE(VESCH(2),'(I1)')FLAG(7)+1
      AD(IL)='PORT '//VESCH(1)//': VESSEL PORT RELATING TO THE'//STG2//'SIDE.'
      ELSE IF (PHASE.EQ.2) THEN
      IL=IL+1
      WRITE(VESCH(1),'(I1)')FLAG(7)+1
      AD(IL)='PORT '//VESCH(1)//': VESSEL PORT FOR LIQUID IN THE'//STG2//'SIDE.'
      IL=IL+1
      WRITE(VESCH(2),'(I1)')FLAG(7)+2
      AD(IL)='PORT '//VESCH(2)//': VESSEL PORT FOR VAPOUR IN THE'//STG2//'SIDE.'
      END IF
      IF (PHASE.NE.0) THEN
      IL=IL+1
      WRITE(VESCH(3),'(I1)')FLAG(8)
      AD(IL)='PORT '//VESCH(3)//': VESSEL PORT FOR'
      IF (GRAD.EQ.0) THEN
       AD(IL)=AD(IL)(1:23)//' LIQUID TO VAPOUR'
      ELSE
       AD(IL)=AD(IL)(1:23)//' VAPOUR TO LIQUID'
      END IF
      AD(IL)=AD(IL)(1:40)//' FLOW IN THE'//STG2//'SIDE.'
      END IF
C     ----------------------------------------------------------------------
C     Display the model description for the user.
C     ----------------------------------------------------------------------
      PRINT3,'THE MODEL HAS BEEN DEFINED AS;'
      PRINT*,' '
      DO 70,I=1,IL
70    PRINT*,AD(I)
C     ----------------------------------------------------------------------
C     Generate the propagation equation.
C     ----------------------------------------------------------------------
```

```fortran
      CALL HEXCHPROP(VESCH,POCH)
C     ----------------------------------------------------------------------
C     Generate the event statements.
C     ----------------------------------------------------------------------
      PRINT3,'GENERATING THE EVENT STATEMENTS - PLEASE WAIT.'
      CALL HEXCHSTATE(VESCH,POCH)
C     ----------------------------------------------------------------------
C     Generate the decision tables.
C     ----------------------------------------------------------------------
      CALL HEXCHTABLE(VESCH,POCH)
C     ----------------------------------------------------------------------
C     Determine the supplimentary information.
C     The normal state lists the ports where flow normally exists.
C     ----------------------------------------------------------------------
      NS='FLOW IS NORMAL THROUGH PORTS'
      DO 100,I=1,FLAG(7)
      IF (NF(I).EQ.0) THEN
      WRITE(PCH,'(I1)')I
      L1=LENGTH(NS)
      NS=NS(1:L1)//' '//PCH//','
      END IF
100   CONTINUE
      NS=NS(1:L1-1)//' AND '//PCH//'.'
C     ----------------------------------------------------------------------
C     The generation of the model is complete.
C     ----------------------------------------------------------------------
      RETURN
      END
```

```fortran
      SUBROUTINE HEXCHPROP(VESCH,POCH)

C     ------------------------------------------------------------------------
C     Subroutine to generate prop. equations for the heat exchanger model.
C     ------------------------------------------------------------------------
      CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
     *  FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
      INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),PHASE,
     *  GRAD,NF(10),ET(10),RF(10),FLAG(10)

      COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
      COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
      COMMON/TWELVE/LG,PHASE,GRAD,NF,ET,RF,FLAG

      CHARACTER VESCH(3),POCH(5),PCH,TEMPFE(45)*80,STRING*6
      INTEGER L1

3     FORMAT(/,' ',A,I2,A)
C     ------------------------------------------------------------------------
C     Initialize the temporary storage strings.
C     ------------------------------------------------------------------------
      DO 50,I=1,45
50    TEMPFE(I)=FE(I)
C     ------------------------------------------------------------------------
C     Generate the first side inlet port flow, standard flow equation.
C     ------------------------------------------------------------------------
      TEMPFE(1)='G1IN=F(Q1IN,Q2OUT)'
C     ------------------------------------------------------------------------
C     Generate the first side inlet port relief.
C     ------------------------------------------------------------------------
      TEMPFE(2)='R1IN=F(R2OUT)'
      FL=2
C     ------------------------------------------------------------------------
C     Generate the first side inlet port temperature. The equation depends on
C     the relative first side temperature and if vessel ports are present. The
C     liquid vessel port is used for a hot first side and the vapour one for a
C     cold first side.
C     ------------------------------------------------------------------------
```

D-32

```
      IF (RF(1).EQ.0) THEN
       IF (RF(2).EQ.0) THEN
        TEMPFE(3)='U1IN=F(U2OUT,'
       ELSE
        TEMPFE(3)='U1IN=F('
       END IF
       STRING='G'//POCH(1)//'IN,'
       IF (GRAD.EQ.0) STRING='-G'//POCH(1)//'IN,'
       L1=LENGTH(TEMPFE(3))
       TEMPFE(3)=TEMPFE(3)(1:L1)//STRING
       STRING='T'//POCH(1)//'IN)'
       IF (PHASE.NE.0) THEN
        STRING='T'//VESCH(1)//'VES)'
        IF (PHASE.EQ.2.AND.GRAD.EQ.1) STRING(2:2)=VESCH(2)
       END IF
       L1=LENGTH(TEMPFE(3))
       TEMPFE(3)=TEMPFE(3)(1:L1)//STRING
       FL=3
```
C   --
C   Generate the first side inlet port composition use the multi-component
C   term 'C' if necessary.
C   --
```
       IF (RF(2).EQ.0) THEN
        TEMPFE(4)='Y1IN=F(Y2OUT)'
        IF (MC.EQ.'Y') TEMPFE(4)='YC1IN=F(YC2OUT)'
        FL=4
       END IF
      END IF
```
C   --
C   Generate the first side outlet port flow, standard flow equation.
C   --
```
      TEMPFE(FL+1)='Q2OUT=F(G1IN,G2OUT)'
```
C   --
C   Generate the first side outlet port pressure.
C   --
```
      TEMPFE(FL+2)='P2OUT=F(P1IN)'
```
C   --
C   Generate the first side outlet port temperature. The equation depends on
C   the relative first side temperature and if vessel ports are present. The
```

```
C     liquid vessel port is used for a hot first side and the vapour one for a
C     cold first side.
C     ------------------------------------------------------------------------
      STRING='G1IN,'
      IF (GRAD.EQ.1) STRING='-G1IN,'
      TEMPFE(FL+3)='T2OUT=F('//STRING
      STRING='G'//POCH(1)//'IN,'
      IF (GRAD.EQ.0) STRING='-G'//POCH(1)//'IN,'
      L1=LENGTH(TEMPFE(FL+3))
      TEMPFE(FL+3)=TEMPFE(FL+3)(1:L1)//'T1IN,'//STRING
      STRING='T'//POCH(1)//'IN)'
      IF (PHASE.NE.0) THEN
       STRING='T'//VESCH(1)//'VES)'
       IF (PHASE.EQ.2.AND.GRAD.EQ.1) STRING(2:2)=VESCH(2)
      END IF
      L1=LENGTH(TEMPFE(FL+3))
      TEMPFE(FL+3)=TEMPFE(FL+3)(1:L1)//STRING
C     ------------------------------------------------------------------------
C     Generate the first side outlet port composition use the multi-component
C     term 'C' if necessary.
C     ------------------------------------------------------------------------
      TEMPFE(FL+4)='X2OUT=F(X1IN)'
      IF (MC.EQ.'Y') TEMPFE(FL+4)='XC2OUT=F(XC1IN)'
      FL=FL+4
C     ------------------------------------------------------------------------
C     If any exist, generate the first side relif/drain port(s).  Use the
C     inlet pressure in the flow and pressure equations.  Use the same
C     temperature and composition equations as the outlet port.
C     ------------------------------------------------------------------------
      DO 70,I=1,FLAG(3)
       WRITE(PCH,'(I1)')I+2
       TEMPFE(FL+1)='Q'//PCH//'OUT=F(G'//PCH//'OUT,P1IN)'
       TEMPFE(FL+2)='P'//PCH//'OUT=F(P1IN)'
       TEMPFE(FL+3)=TEMPFE(FL-1)
       TEMPFE(FL+3)(2:2)=PCH
       TEMPFE(FL+4)=TEMPFE(FL)
       IF (MC.EQ.'N') THEN
        TEMPFE(FL+4)(2:2)=PCH
       ELSE
```

```fortran
          TEMPFE(FL+4)(3:3)=PCH
          END IF
70        FL=FL+4
C         -----------------------------------------------------------------------------------
C         Generate the second side inlet port flow.  For no phase change use
C         the standard flow equation, else use a pressure term, if necessary.
C         -----------------------------------------------------------------------------------
          TEMPFE(FL+1)='G'//POCH(1)//'IN=F(Q'//POCH(1)//'IN'
          IF (PHASE.EQ.0) THEN
           TEMPFE(FL+1)=TEMPFE(FL+1)(1:11)//',Q'//POCH(2)//'OUT)'
          ELSE IF (ET(FLAG(3)+3).EQ.0) THEN
           TEMPFE(FL+1)=TEMPFE(FL+1)(1:11)//',-P'//VESCH(2)//'VES)'
          ELSE
           TEMPFE(FL+1)=TEMPFE(FL+1)(1:11)//')'
          END IF
C         -----------------------------------------------------------------------------------
C         Generate the second side inlet port relief.
C         -----------------------------------------------------------------------------------
          TEMPFE(FL+2)='R'//POCH(1)//'IN=F(R'//POCH(2)//'OUT)'
          IF (PHASE.NE.0) THEN
           TEMPFE(FL+2)='R'//POCH(1)//'IN=F(-P'//VESCH(2)//'VES)'
          END IF
          FL=FL+2
C         -----------------------------------------------------------------------------------
C         Generate the second side inlet port temperature, use flow and
C         temperature terms or the vessel temperature which may be taken
C         at the liquid or vapour vessel port.
C         -----------------------------------------------------------------------------------
          IF (RF(FLAG(3)+3).EQ.0) THEN
           FL=FL+1
           IF (PHASE.EQ.0) THEN
            IF (GRAD.EQ.0) THEN
             TEMPFE(FL)='U'//POCH(1)//'IN=F(G1IN,T1IN)'
             L1=16
            ELSE
             TEMPFE(FL)='U'//POCH(1)//'IN=F(-G1IN,T1IN)'
             L1=17
            END IF
            IF (RF(FLAG(3)+4).EQ.0) TEMPFE(FL)=TEMPFE(FL)(1:L1)//',U'//POCH(2)//'OUT)'
           ELSE
```

D-35

```
      TEMPFE(FL)='U'//POCH(1)//'IN=F(T'//VESCH(1)//'VES)'
      IF (GRAD.EQ.1) TEMPFE(FL)(9:9)=VESCH(2)
      END IF
```

C   ---
C   Generate second side inlet port composition. This may be the exit or
C   the vessel composition. For a partial phase change multi-components may
C   be necessary and either the liquid or vapour vessel port may be used.
C   ---

```
      FL=FL+1
      IF (PHASE.EQ.0) THEN
       IF (RF(FLAG(3)+4).EQ.0) THEN
        TEMPFE(FL)='Y'//POCH(1)//'IN=F(Y'//POCH(2)//'OUT)'
       END IF
      ELSE IF (PHASE.EQ.1) THEN
       TEMPFE(FL)='Y'//POCH(1)//'IN=F(X'//VESCH(1)//'VES)'
      ELSE
       IF (MC.EQ.'N') THEN
        TEMPFE(FL)='Y'//POCH(1)//'IN=F(X'//VESCH(1)//'VES)'
        IF (GRAD.EQ.1) TEMPFE(FL)(9:9)=VESCH(2)
       ELSE
        IF (GRAD.EQ.0) THEN
         TEMPFE(FL)='YA'//POCH(1)//'IN=F(-XB'//VESCH(1)//'VES)'
         TEMPFE(FL+1)='YB'//POCH(1)//'IN=F(XB'//VESCH(1)//'VES)'
        ELSE
         TEMPFE(FL)='YA'//POCH(1)//'IN=F(XA'//VESCH(2)//'VES)'
         TEMPFE(FL+1)='YB'//POCH(1)//'IN=F(-XA'//VESCH(2)//'VES)'
        END IF
        FL=FL+1
       END IF
      END IF
      END IF
```

C   ---
C   Generate the second side outlet port(s) flow. This may be the single
C   outlet or either the liquid or vapour outlet port for a partial phase
C   change. For no phase change use the standard flow equation, else use a
C   pressure term, if necessary.
C   ---

```
      J=1
      IF (PHASE.EQ.2) J=2
      DO 100,I=1,J
       IF (PHASE.EQ.0) THEN
```

D-36

```
        TEMPFE(FL+1)='Q'//POCH(2)//'OUT=F(G'//POCH(1)//'IN,G'//POCH(2)//'OUT)'
      ELSE
        TEMPFE(FL+1)='Q'//POCH(I+1)//'OUT=F(G'//POCH(I+1)//'OUT'
        IF (ET(FLAG(3)+I+3).EQ.0) THEN
          TEMPFE(FL+1)=TEMPFE(FL+1)(1:13)//',P'//VESCH(2)//'VES)'
        ELSE
          TEMPFE(FL+1)=TEMPFE(FL+1)(1:13)//')'
        END IF
      END IF
C       --------------------------------------------------------------------------------
C       Generate the second side outlet port relief.  This may be the
C       inlet pressure or the vessel pressure.
C       --------------------------------------------------------------------------------
        TEMPFE(FL+2)='P'//POCH(I+1)//'OUT=F(P'//POCH(1)//'IN)'
        IF (PHASE.NE.0) THEN
          TEMPFE(FL+2)='P'//POCH(I+1)//'OUT=F(P'//VESCH(2)//'VES)'
        END IF
C       --------------------------------------------------------------------------------
C       Generate the second side outlet port temperature, use flow and
C       temperature terms or the vessel temperature which may be taken
C       at the liquid or vapour vessel port.
C       --------------------------------------------------------------------------------
        IF (PHASE.EQ.0) THEN
          IF (GRAD.EQ.0) THEN
            TEMPFE(FL+3)='T'//POCH(2)//'OUT=F(G1IN,T1IN,-G'//POCH(1)//'IN,T'//POCH(1)//'IN)'
          ELSE
            TEMPFE(FL+3)='T'//POCH(2)//'OUT=F(-G1IN,T1IN,G'//POCH(1)//'IN,T'//POCH(1)//'IN)'
          END IF
        ELSE
          TEMPFE(FL+3)='T'//POCH(I+1)//'OUT=F(T'//VESCH(I)//'VES)'
        END IF
C       --------------------------------------------------------------------------------
C       Generate second side outlet port composition.  This may be the exit or
C       the vessel composition.  For a partial phase change multi-components may
C       be necessary and either the liquid or vapour vessel port may be used.
C       --------------------------------------------------------------------------------
        FL=FL+4
        IF (PHASE.EQ.0) THEN
          TEMPFE(FL)='X'//POCH(2)//'OUT=F(X'//POCH(1)//'IN)'
        ELSE IF (MC.EQ.'N') THEN
          TEMPFE(FL)='X'//POCH(I+1)//'OUT=F(X'//VESCH(I)//'VES)'
```

```
        ELSE
          IF (I.EQ.1) THEN
            TEMPFE(FL)='XA'//POCH(2)//'OUT=F(-XB'//VESCH(1)//'VES)'
            TEMPFE(FL+1)='XB'//POCH(2)//'OUT=F(XB'//VESCH(1)//'VES)'
          ELSE
            TEMPFE(FL)='XA'//POCH(3)//'OUT=F(XA'//VESCH(2)//'VES)'
            TEMPFE(FL+1)='XB'//POCH(3)//'OUT=F(-XA'//VESCH(2)//'VES)'
          END IF
          FL=FL+1
        END IF
100     CONTINUE
C       -----------------------------------------------------------------------------------
C       If there is a second side relief and/or drain port, generate the
C       necessary equations. Use the inlet or vessel pressure
C       in the flow equation.
C       -----------------------------------------------------------------------------------
        DO 130,I=1,FLAG(5)
        J=5
        IF (I.EQ.1.AND.(FLAG(5).EQ.2.OR.FLAG(6).EQ.1)) J=4
        TEMPFE(FL+1)='Q'//POCH(J)//'OUT=F(G'//POCH(J)//'OUT,P'
        IF (PHASE.EQ.0) THEN
          TEMPFE(FL+1)=TEMPFE(FL+1)(1:15)//POCH(1)//'IN)'
        ELSE
          TEMPFE(FL+1)=TEMPFE(FL+1)(1:15)//VESCH(2)//'VES)'
        END IF
C       -----------------------------------------------------------------------------------
C       Use the inlet or vessel pressure in the pressure equation.
C       -----------------------------------------------------------------------------------
        TEMPFE(FL+2)='P'//POCH(J)//'OUT=F(P'//POCH(1)//'IN)'
        IF (PHASE.NE.0) THEN
          TEMPFE(FL+2)='P'//POCH(J)//'OUT=F(P'//VESCH(2)//'VES)'
        END IF
C       -----------------------------------------------------------------------------------
C       Use the same outlet port temperature equation.
C       -----------------------------------------------------------------------------------
        IF (PHASE.EQ.0) THEN
        IF (GRAD.EQ.0) THEN
          TEMPFE(FL+3)='T'//POCH(J)//'OUT=F(G1IN,T1IN,-G'//POCH(1)//'IN,T'//POCH(1)//'IN)'
        ELSE
          TEMPFE(FL+3)='T'//POCH(J)//'OUT=F(-G1IN,T1IN,G'//POCH(1)//'IN,T'//POCH(1)//'IN)'
        END IF
```

D-38

```
      ELSE
       TEMPFE(FL+3)='T'//POCH(J)//'OUT=F(T'//VESCH(1)//'VES)'
       IF (J.EQ.4) TEMPFE(FL+3)(10:10)=VESCH(2)
      END IF
C     ------------------------------------------------------------------
C     Use the same outlet port composition equation(s).
C     ------------------------------------------------------------------
      FL=FL+4
      IF (PHASE.EQ.0) THEN
       TEMPFE(FL)='X'//POCH(J)//'OUT=F(X'//POCH(1)//'IN)'
      ELSE IF (PHASE.EQ.1) THEN
       TEMPFE(FL)='X'//POCH(J)//'OUT=F(X'//VESCH(1)//'VES)'
      ELSE
       IF (MC.EQ.'N') THEN
        TEMPFE(FL)='XA'//POCH(J)//'OUT=F(-XB'//VESCH(1)//'VES)'
        IF (J.EQ.4) TEMPFE(FL)(10:10)=VESCH(2)
       ELSE
        IF (J.EQ.4) THEN
         TEMPFE(FL)='XA'//POCH(4)//'OUT=F(XA'//VESCH(2)//'VES)'
         TEMPFE(FL+1)='XB'//POCH(4)//'OUT=F(-XA'//VESCH(2)//'VES)'
        ELSE
         TEMPFE(FL)='XA'//POCH(5)//'OUT=F(-XB'//VESCH(1)//'VES)'
         TEMPFE(FL+1)='XB'//POCH(5)//'OUT=F(XB'//VESCH(1)//'VES)'
        END IF
        FL=FL+1
       END IF
      END IF
130   CONTINUE
C     ------------------------------------------------------------------
C     Vessel ports are only necessary if phase changes occur.  Generate
C     the level term, as a balance of the inlet and outlet liquid flows
C     and the flow from liquid to vapour or vice versa.
C     ------------------------------------------------------------------
      IF (PHASE.NE.0) THEN
       FL=FL+1
       IF (GRAD.EQ.0) THEN
        TEMPFE(FL)='L'//VESCH(1)//'VES=F(G'//POCH(1)//'IN,'
        IF (PHASE.EQ.2) TEMPFE(FL)=TEMPFE(FL)(1:13)//'-Q'//POCH(2)//'OUT,'
        L1=LENGTH(TEMPFE(FL))
        TEMPFE(FL)=TEMPFE(FL)(1:L1)//'-Q'//VESCH(3)//'VES)'
```

D-39

```
        ELSE
        TEMPFE(FL)='L'//VESCH(1)//'VES=F(-Q'//POCH(2)//'OUT,Q'//VESCH(3)//'VES)'
        END IF
C       ---------------------------------------------------------------------------------------
C       Generate the pressure term if there is a complete phase change,
C       this is a balance of the inlet and outlet vapour flows and the
C       flow from liquid to vapour of vice versa.
C       ---------------------------------------------------------------------------------------
        IF (PHASE.EQ.1) THEN
        FL=FL+1
        IF (GRAD.EQ.0) THEN
         TEMPFE(FL)='P'//VESCH(1)//'VES=F(-Q'//POCH(2)//'OUT,'
        ELSE
         TEMPFE(FL)='P'//VESCH(1)//'VES=F(G'//POCH(1)//'IN,'
        END IF
        L1=LENGTH(TEMPFE(FL))
        TEMPFE(FL)=TEMPFE(FL)(1:L1)//'L'//VESCH(1)//'VES,'
        L1=L1+6
        IF (GRAD.EQ.0) THEN
         TEMPFE(FL)=TEMPFE(FL)(1:L1)//'Q'//VESCH(3)//'VES)'
        ELSE
         TEMPFE(FL)=TEMPFE(FL)(1:L1)//'-Q'//VESCH(3)//'VES)'
        END IF
        END IF
C       ---------------------------------------------------------------------------------------
C       Generate the temperature term.
C       ---------------------------------------------------------------------------------------
        FL=FL+1
        IF (MC.EQ.'N') THEN
        I=PHASE
        TEMPFE(FL)='T'//VESCH(1)//'VES=F(P'//VESCH(I)//'VES,-Q'//VESCH(3)//'VES)'
        ELSE
        TEMPFE(FL)='T'//VESCH(1)//'VES=F(P'//VESCH(2)//'VES,XB'//VESCH(1)//'VES)'
        END IF
C       ---------------------------------------------------------------------------------------
C       Generate the composition term.
C       ---------------------------------------------------------------------------------------
        FL=FL+1
        IF (MC.EQ.'N') THEN
         TEMPFE(FL)='X'//VESCH(1)//'VES=F(X'//POCH(1)//'IN)'
        ELSE
```

```
      IF (GRAD.EQ.0) THEN
       TEMPFE(FL)='XB'//VESCH(1)//'VES=F(XB'//POCH(1)//'IN,Q'//VESCH(3)//'VES)'
      ELSE
       TEMPFE(FL)='XB'//VESCH(1)//'VES=F(-XA'//VESCH(2)//'VES,-Q'//VESCH(3)//'VES)'
      END IF
      END IF
```

C   --
C   For a partial phase changes, an additional vessel port is required.
C   Generate the pressure term.
C   --

```
      IF (PHASE.EQ.2) THEN
      FL=FL+1
      IF (GRAD.EQ.0) THEN
       TEMPFE(FL)='P'//VESCH(2)//'VES=F(-Q'//POCH(3)//'OUT,'
      ELSE
       TEMPFE(FL)='P'//VESCH(2)//'VES=F(G'//POCH(1)//'IN,-Q'//POCH(3)//'OUT,'
      END IF
      L1=LENGTH(TEMPFE(FL))
      TEMPFE(FL)=TEMPFE(FL)(1:L1)//'L'//VESCH(1)//'VES,'
      L1=L1+6
      IF (GRAD.EQ.0) THEN
       TEMPFE(FL)=TEMPFE(FL)(1:L1)//'Q'//VESCH(3)//'VES)'
      ELSE
       TEMPFE(FL)=TEMPFE(FL)(1:L1)//'-Q'//VESCH(3)//'VES)'
      END IF
```

C   --
C   Generate the temperature term, for single components use the
C   pressure and flow else use the pressure and composition.
C   --

```
      FL=FL+1
      IF (MC.EQ.'N') THEN
       TEMPFE(FL)='T'//VESCH(2)//'VES=F(P'//VESCH(2)//'VES,-Q'//VESCH(3)//'VES)'
      ELSE
       TEMPFE(FL)='T'//VESCH(2)//'VES=F(P'//VESCH(2)//'VES,-XA'//VESCH(2)//'VES)'
      END IF
```

C   --
C   Generate the composition term.
C   --

```
      FL=FL+1
      IF (MC.EQ.'N') THEN
       TEMPFE(FL)='X'//VESCH(2)//'VES=F(X'//POCH(1)//'IN)'
      ELSE
       IF (GRAD.EQ.0) THEN
        TEMPFE(FL)='XA'//VESCH(2)//'VES=F(-XB'//VESCH(1)//'VES,-Q'//VESCH(3)//'VES)'
       ELSE
        TEMPFE(FL)='XA'//VESCH(2)//'VES=F(XA'//POCH(1)//'IN,Q'//VESCH(3)//'VES)'
       END IF
      END IF
      END IF
C     --------------------------------------------------------------------------------
C     Generate the vessel port equation used to model the flow of liquid
C     to vapour or vice verse.
C     --------------------------------------------------------------------------------
      FL=FL+1
      IF (GRAD.EQ.0) THEN
       TEMPFE(FL)='Q'//VESCH(3)//'VES=F(G1IN,T1IN,'
       IF (MC.EQ.'Y') TEMPFE(FL)=TEMPFE(FL)(1:18)//'-XB'//VESCH(1)//'VES,'
       L1=LENGTH(TEMPFE(FL))
       TEMPFE(FL)=TEMPFE(FL)(1:L1)//'-P'//VESCH(2)//'VES)'
      ELSE
       TEMPFE(FL)='Q'//VESCH(3)//'VES=F(G1IN,-T1IN,'
       IF (MC.EQ.'Y') TEMPFE(FL)=TEMPFE(FL)(1:19)//'-XA'//VESCH(2)//'VES,'
       L1=LENGTH(TEMPFE(FL))
       TEMPFE(FL)=TEMPFE(FL)(1:L1)//'P'//VESCH(2)//'VES)'
      END IF
      END IF
C     --------------------------------------------------------------------------------
C     Completion of propagation equations, now check.
C     --------------------------------------------------------------------------------
      PRINT3,'THE FOLLOWING ',FL,' PROPAGATION EQUATIONS HAVE BEEN CREATED:'
      CALL CHECK(TEMPFE,FL)
C     --------------------------------------------------------------------------------
C     Take prop. eqns. from the temporary array and place in the FE array.
C     --------------------------------------------------------------------------------
      K=FL
      FL=0
      DO 200,I=1,K
       IF (TEMPFE(I)(1:1).NE.' ') THEN
        FL=FL+1
```

```
             L1=INDEX(TEMPFE(I),' ')
             IF (L1.EQ.0) L1=80
             FE(FL)=TEMPFE(I)(1:L1)
             END IF
200      CONTINUE


         RETURN
         END




         SUBROUTINE HEXCHSTATE(VESCH,POCH)


C        --------------------------------------------------------------------------------------------
C        Subrouitine to generate event statements for the heat exchanger model.
C        --------------------------------------------------------------------------------------------
         CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
     *   FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
         INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),PHASE,
     *   GRAD,NF(10),ET(10),RF(10),FLAG(10)

         COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
         COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
         COMMON/TWELVE/LG,PHASE,GRAD,NF,ET,RF,FLAG

         CHARACTER VESCH(3),POCH(5),PCH,TEMPFI(50)*80,BASICF*10,TEMPEV(25)*12,S1*3,S2*3
         INTEGER L1,FAUPOR(50,12)

3        FORMAT(/,' ',A,I2,A)
C        --------------------------------------------------------------------------------------------
C        Initialize the temporary storage strings.
C        --------------------------------------------------------------------------------------------
         DO 50,I=1,50
          TEMPFI(I)=FI(I)
50        FAUPOR(I,1)=0
         JL=0
C        --------------------------------------------------------------------------------------------
C        Set up S1 and S2 character arrays for heating or cooling of second side.
C        --------------------------------------------------------------------------------------------
```

```fortran
        IF (GRAD.EQ.0) THEN
        S1=' HI'
        S2=' LO'
        ELSE
        S1=' LO'
        S2=' HI'
        END IF
C       ------------------------------------------------------------------------
C       For the second side liquid outlet port, enter the effect of HIgh flow on
C       the level if there is a phase change.
C       ------------------------------------------------------------------------
        IF (PHASE.EQ.2.OR.(PHASE.EQ.1.AND.GRAD.EQ.1)) THEN
        JL=JL+1
        TEMPFI(JL)='V Q'//POCH(2)//'OUT HI:L'//VESCH(1)//'VES NONE'
        END IF
C       ------------------------------------------------------------------------
C       For each of the relief and/or drain port(s) generate the effects of SOME
C       flow through the port.  These are HIgh inlet flows and relief, LOw, NONE
C       and possibly REVerse outlet flows and pressure.  If there is a phase
C       change only LOw vessel pressure is considered for relief outlets but also
C       LOw and NO level for drain outlets.
C       ------------------------------------------------------------------------
        DO 70,I=1,FLAG(3)
        JL=JL+1
        WRITE(PCH,'(I1)')I+2
        TEMPFI(JL)='V Q'//PCH//'OUT SOME:G1IN HI,Q2OUT LO,Q2OUT NONE'
        IF (RF(2).EQ.0) THEN
         TEMPFI(JL)=TEMPFI(JL)(1:40)//',Q2OUT REV,R1IN HI,P2OUT LO'
         JL=JL+1
         TEMPFI(JL)=TEMPFI(JL-1)(1:13)//'P2OUT NONE,P2OUT REV'
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:40)//',R1IN HI,P2OUT LO,P2OUT NONE'
        END IF
70      CONTINUE

        DO 75,I=1,FLAG(5)
        JL=JL+1
        IF (FLAG(5).EQ.1) THEN
        J=FLAG(7)
        ELSE
         J=FLAG(7)-2+I
```

```fortran
      END IF
      WRITE(PCH,'(I1)')J
      TEMPFI(JL)='V Q'//PCH//' OUT SOME:'
      L1=13
      IF (PHASE.EQ.0) THEN
       TEMPFI(JL)=TEMPFI(JL)(1:13)//'G'//POCH(1)//' IN HI,Q'//POCH(2)//' OUT LO,Q'
   *     //POCH(2)//' OUT NONE'
       IF (RF(4+FLAG(3)).EQ.0) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:40)//',Q'//POCH(2)//' OUT REV,R'//POCH(1)//' IN HI,P'
   *     //POCH(2)//' OUT LO'
        JL=JL+1
        TEMPFI(JL)=TEMPFI(JL-1)(1:13)//' P'//POCH(2)//' OUT NONE,P'//POCH(2)//' OUT REV'
       ELSE
        TEMPFI(JL)=TEMPFI(JL)(1:40)//',R'//POCH(1)//' IN HI,P'//POCH(2)//' OUT LO,P'
   *     //POCH(2)//' OUT NONE'
       END IF
      ELSE
       IF ((FLAG(5).EQ.2.AND.I.EQ.2).OR.(FLAG(5).EQ.1.AND.FLAG(6).EQ.2)) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:13)//'L'//VESCH(1)//' VES LO,L'//VESCH(1)//' VES NONE,'
        L1=33
       END IF
       TEMPFI(JL)=TEMPFI(JL)(1:L1)//'P'//VESCH(2)//' VES LO'
      END IF
75    CONTINUE
C     ------------------------------------------------------------------------
C     Consider the effect of REVerse flow through the inlet flow port of a
C     phase change situation.  The others are modelled by the propagation
C     equations.  These are LOw vessel pressure and for a cold side LOw and
C     No level.
C     ------------------------------------------------------------------------
      IF (PHASE.NE.0) THEN
       IF (RF(FLAG(3)+3).EQ.0) THEN
        JL=JL+1
        TEMPFI(JL)='V G'//POCH(1)//' IN REV:'
        L1=11
        IF (GRAD.EQ.0) THEN
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//'L'//VESCH(1)//' VES LO,L'//VESCH(1)//' VES NONE,'
         L1=31
        END IF
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//'P'//VESCH(1)//' VES LO'
       END IF
```

```fortran
C      -----------------------------------------------------------------
C      Consider the effect of REVerse flow through the outlet flow port of a
C      phase change situation. The others are modelled by the propagation
C      equations. These are High vessel pressure and for a hot side High level.
C      -----------------------------------------------------------------
       IF (RF(FLAG(3)+4).EQ.0) THEN
       JL=JL+1
       TEMPFI(JL)='V Q'//POCH(2)//'OUT REV:'
       L1=12
       IF (GRAD.EQ.1.OR.PHASE.EQ.2) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//'L'//VESCH(1)//'VES HI,'
        L1=21
       END IF
       TEMPFI(JL)=TEMPFI(JL)(1:L1)//'P'//VESCH(1)//'VES HI'
       END IF
       IF (RF(FLAG(3)+5).EQ.0.AND.PHASE.EQ.2) THEN
       JL=JL+1
       TEMPFI(JL)='V Q'//POCH(3)//'OUT REV:P'//VESCH(2)//'VES HI'
       END IF
       END IF
C      -----------------------------------------------------------------
C      For each of the relief and/or drain port(s) generate the effects of
C      REVerse flow through the port. These are lower inlet and higher outlet
C      flows and pressure.
C      -----------------------------------------------------------------
       DO 80,I=1,FLAG(3)
       JL=JL+1
       WRITE(PCH,'(I1)')I+2
       TEMPFI(JL)='V Q'//PCH//'OUT REV:G1IN LO,G1IN NONE,'
       IF (RF(1).EQ.0) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:30)//'G1IN REV,Q2OUT HI,R1IN LO'
        JL=JL+1
        TEMPFI(JL)='V Q'//PCH//'OUT REV:R1IN NONE,R1IN REV,P2OUT HI,'
        IF (MC.EQ.'N') THEN
         TEMPFI(JL)=TEMPFI(JL)(1:40)//'Y1IN HI,X2OUT HI'
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:40)//'YD1IN HI,XD2OUT HI'
        END IF
       ELSE
        TEMPFI(JL)=TEMPFI(JL)(1:30)//'Q2OUT HI,R1IN LO,R1IN NONE,P2OUT HI'
        JL=JL+1
```

```fortran
      IF (MC.EQ.'N') THEN
       TEMPFI(JL)='V Q'//PCH//'OUT REV:X2OUT HI'
      ELSE
       TEMPFI(JL)='V Q'//PCH//'OUT REV:XD2OUT HI'
      END IF
      END IF
80    CONTINUE

      DO 85,I=1,FLAG(5)
      JL=JL+1
      IF (FLAG(5).EQ.1) THEN
       J=FLAG(7)
      ELSE
       J=FLAG(7)-2+I
      END IF
      WRITE(PCH,'(I1)')J
      TEMPFI(JL)='V Q'//PCH//'OUT REV:'
      IF (PHASE.EQ.0) THEN
       TEMPFI(JL)=TEMPFI(JL)(1:12)//'G'//POCH(1)//'IN LO,G'//POCH(1)//'IN NONE,'
       IF (RF(3+FLAG(3)).EQ.0) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:30)//'G'//POCH(1)//'IN REV,Q'//POCH(2)//'OUT HI'
        JL=JL+1
        TEMPFI(JL)=TEMPFI(JL-1)(1:12)//'R'//POCH(1)//'IN LO,R'//POCH(1)//'IN NONE,R'
     *    //POCH(1)//'IN REV,Q'//POCH(2)//'OUT HI'
        JL=JL+1
        TEMPFI(JL)='V Q'//PCH//'OUT REV:Y'//POCH(1)//'IN HI,X'//POCH(2)//'OUT HI'
       ELSE
        TEMPFI(JL)=TEMPFI(JL)(1:30)//'Q'//POCH(2)//'OUT HI,R'//POCH(1)//'IN LO,R'
     *    //POCH(1)//'IN NONE,P'//POCH(2)//'OUT HI'
        JL=JL+1
        TEMPFI(JL)='V Q'//PCH//'OUT REV:X'//POCH(2)//'OUT HI'
       END IF
      ELSE
       IF ((FLAG(5).EQ.2.AND.I.EQ.2).OR.(FLAG(5).EQ.1.AND.FLAG(6).EQ.2)) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:12)//'L'//VESCH(1)//'VES HI,P'//VESCH(2)//'VES HI,'
        IF (MC.EQ.'N') THEN
         TEMPFI(JL)=TEMPFI(JL)(1:30)//'X'//VESCH(1)//'VES HI'
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:30)//'XD'//VESCH(1)//'VES HI'
        END IF
       ELSE
```

```
        TEMPFI(JL)=TEMPFI(JL)(1:12)//'P'//VESCH(2)//'VES HI,'
        IF (MC.EQ.'N') THEN
         TEMPFI(JL)=TEMPFI(JL)(1:21)//'X'//VESCH(2)//'VES HI'
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:21)//'XD'//VESCH(2)//'VES HI'
        END IF
        END IF
        END IF
85      CONTINUE
C       -----------------------------------------------------------------------
C       If there is a phase change generate effects of HIgh, LOw and NO level.
C       -----------------------------------------------------------------------
        IF (PHASE.NE.0) THEN
        JL=JL+1
        TEMPFI(JL)='V L'//VESCH(1)//'VES HI:T2OUT'//S2//',Q'//VESCH(3)//'VES'//S1
        JL=JL+1
        TEMPFI(JL)='V L'//VESCH(1)//'VES LO:T2OUT'//S1//',Q'//VESCH(3)//'VES'//S2
        JL=JL+1
        TEMPFI(JL)='V L'//VESCH(1)//'VES NONE:T2OUT'//S1
        L1=21
        IF (GRAD.EQ.0) THEN
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//VESCH(2)//'VES HI'
         L1=L1+9
        END IF
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//',P'//VESCH(2)//'VES LO,Q'
        L1=L1+1
        IF (GRAD.EQ.0) THEN
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//VESCH(3)//'VES NONE'
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//VESCH(3)//'VES HI'
        END IF
C       -----------------------------------------------------------------------
C       If there is a phase change then generate the effects of HIgh and LOw
C       pressure.  Use the EVSTAT routine.
C       -----------------------------------------------------------------------
        BASICF='V P'//VEXCH(2)//'VES HI'
        NDEV=0
        TEMPEV(NDEV+1)='G'//POCH(1)//'IN NONE'
        TEMPEV(NDEV+2)='R'//POCH(1)//'IN NONE'
        NDEV=NDEV+2
        IF (RF(FLAG(3)+3).EQ.0) THEN
```

```
          TEMPEV(NDEV+1)='G'//POCH(1)//'IN REV'
          TEMPEV(NDEV+2)='R'//POCH(1)//'IN REV'
          NDEV=NDEV+2
       END IF
       CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)

       BASICF='P'//VEXCH(2)//'VES NONE'
       NDEV=0
       IF (RF(FLAG(3)+4).EQ.0) THEN
         TEMPEV(NDEV+1)='Q'//POCH(2)//'OUT REV'
         TEMPEV(NDEV+2)='P'//POCH(2)//'OUT REV'
         NDEV=NDEV+2
       END IF
       IF (PHASE.EQ.2.AND.RF(FLAG(3)+5).EQ.0) THEN
         TEMPEV(NDEV+1)='Q'//POCH(3)//'OUT REV'
         TEMPEV(NDEV+2)='P'//POCH(3)//'OUT REV'
         NDEV=NDEV+2
       END IF
       IF (FLAG(5).NE.0) THEN
         TEMPEV(NDEV+1)='Q'//POCH(4)//'OUT REV'
         TEMPEV(NDEV+2)='P'//POCH(4)//'OUT REV'
         NDEV=NDEV+2
       END IF
       IF (FLAG(5).EQ.2) THEN
         TEMPEV(NDEV+1)='Q'//POCH(5)//'OUT REV'
         TEMPEV(NDEV+2)='P'//POCH(5)//'OUT REV'
         NDEV=NDEV+2
       END IF
       IF (NDEV.NE.0) THEN
       J=JL+1
       CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
       DO 90,I=J,JL
90       TEMPFI(I)='V '//TEMPFI(I)
       END IF
C      ------------------------------------------------------------------------
C      Generate the effects of NO vapour-liquid flow.
C      ------------------------------------------------------------------------
       JL=JL+1
       TEMPFI(JL)='V Q'//VESCH(3)//'VES NONE:L'//VESCH(1)//'VES'//S1//',P'//VESCH(2)//'VES'//S2
       END IF
```

```
C       --------------------------------------------------------------------------------
C       Initialize the FAUPOR array for the basic faults EXT-HEAT, EXT-COLD
C       and LK-LP-EN.  Use GENEV for these.
C       For EXT-HEAT and EXT-COLD define the first side inlet and outlet ports
C       for all cases.  For no phase change add the second side inlet and outlet
C       ports, for a complete phase change add the vessel port and for a partial
C       phase change add the liquid and vapour vessel ports.
C       --------------------------------------------------------------------------------
        DO 100,I=1,2
        FAUPOR(I,1)=I+4
        J=0
        IF (RF(1).EQ.0) THEN
         J=J+1
         FAUPOR(I,2+J)=1
        END IF
        J=J+1
        FAUPOR(I,2+J)=2
        IF (PHASE.EQ.0) THEN
         IF (RF(FLAG(3)+3).EQ.0) THEN
          J=J+1
          FAUPOR(I,2+J)=FLAG(3)+3
         END IF
         J=J+1
         FAUPOR(I,2+J)=FLAG(3)+4
        ELSE
         J=J+1
         FAUPOR(I,2+J)=FLAG(7)+1
         IF (PHASE.EQ.2) THEN
          J=J+1
          FAUPOR(I,2+J)=FLAG(7)+2
         END IF
        END IF
        FAUPOR(I,2)=J
100     CONTINUE
        CALL GENEV(FAUPOR,TEMPFI)
C       --------------------------------------------------------------------------------
C       For LK-LP-EN use FLAG(9) to determine which side has the leak.
C       Define the inlet and outlet ports for no phase change or the vessel
C       port(s) for a phase change.
C       --------------------------------------------------------------------------------
```

```
       FAUPOR(2,1)=0
       FAUPOR(1,1)=1
       IF (FLAG(9).EQ.0) THEN
        FAUPOR(1,2)=2
        IF (PHASE.EQ.0) THEN
         FAUPOR(1,3)=FLAG(3)+3
         FAUPOR(1,4)=FLAG(3)+4
        ELSE
         FAUPOR(1,3)=FLAG(7)+1
         FAUPOR(1,4)=FLAG(7)+2
         IF (PHASE.EQ.1) FAUPOR(1,2)=1
        END IF
       ELSE
        FAUPOR(1,2)=2
        FAUPOR(1,3)=1
        FAUPOR(1,4)=2
       END IF
       CALL GENEV(FAUPOR,TEMPFI)
C      --------------------------------------------------------------------------------
C      Generate the INT-LK fault, the result depends on the pressure difference
C      between the first and second side.  Use the GENEV routine but the
C      composition and temperature effects have to be generated separately.
C      --------------------------------------------------------------------------------
       IF (NF(10).NE.0) THEN
C      --------------------------------------------------------------------------------
C      Deal with the first side first.
C      --------------------------------------------------------------------------------
       FAUPOR(1,1)=9
       IF (NF(10).EQ.2) FAUPOR(1,1)=10
       FAUPOR(1,2)=2
       FAUPOR(1,3)=1
       FAUPOR(1,4)=2
C      --------------------------------------------------------------------------------
C      Now deal with the second side.
C      --------------------------------------------------------------------------------
       FAUPOR(2,1)=10
       IF (NF(10).EQ.2) FAUPOR(2,1)=9
C      --------------------------------------------------------------------------------
C      For no phase change use the inlet and outlet ports, otherwise use the
C      vessel ports.
C      --------------------------------------------------------------------------------
```

```
        IF (PHASE.EQ.0) THEN
         FAUPOR(2,2)=2
         FAUPOR(2,3)=FLAG(3)+3
         FAUPOR(2,4)=FLAG(3)+4
        ELSE
         FAUPOR(2,2)=2
         FAUPOR(2,3)=FLAG(7)+1
         FAUPOR(2,4)=FLAG(7)+2
        END IF
        CALL GENEV(FAUPOR,TEMPFI)
        FAUPOR(1,1)=0
        FAUPOR(2,1)=0
C     ------------------------------------------------------------------------
C     Now generate the temperature and composition effects.
C     ------------------------------------------------------------------------
        JL=JL+1
        IF (NF(10).EQ.1) THEN
         TEMPFI(JL)='F INT-LK:T2OUT'//S2
         IF (PHASE.EQ.0) THEN
         L1=17
         IF (RF(FLAG(3)+3).EQ.0) THEN
          TEMPFI(JL)=TEMPFI(JL)(1:17)//',U'//POCH(1)//'IN'//S1
          L1=25
         END IF
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//POCH(2)//'OUT'//S1
         L1=L1+9
         IF (RF(FLAG(3)+3).EQ.0) THEN
          TEMPFI(JL)=TEMPFI(JL)(1:L1)//',Y'//POCH(1)//'IN HI'
          L1=L1+8
         END IF
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',X'//POCH(2)//'OUT HI'
         ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:17)//',T'//VESCH(1)//'VES'//S1
         IF (MC.EQ.'N') THEN
          TEMPFI(JL)=TEMPFI(JL)(1:26)//',X'//VESCH(1)//'VES HI'
          IF (PHASE.EQ.2) TEMPFI(JL)=TEMPFI(JL)(1:35)//',X'//VESCH(2)//'VES HI'
         ELSE
          TEMPFI(JL)=TEMPFI(JL)(1:26)//',T'//VESCH(2)//'VES'//S1//',XC'//VESCH(1)
     *     //'VES HI'//',XC'//VESCH(2)//'VES HI'
         END IF
         END IF
        END IF
```

```
      ELSE
       IF (RF(1).EQ.0) THEN
        TEMPFI(JL)='F INT-LK:U1IN'//S2//',T2OUT'//S2
        L1=25
       ELSE
        TEMPFI(JL)='F INT-LK:T2OUT'//S2
        L1=17
       END IF
       IF (PHASE.EQ.0) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//POCH(2)//'OUT'//S1
        L1=L1+9
       ELSE
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//VESCH(1)//'VES'//S1
        L1=L1+9
        IF (PHASE.EQ.2) THEN
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//VESCH(2)//'VES'//S1
         L1=L1+9
        END IF
       END IF
       IF (MC.EQ.'N') THEN
        IF (RF(1).EQ.0) THEN
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',Y1IN HI'
         L1=L1+8
        END IF
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//',X2OUT HI'
       ELSE
        IF (RF(1).EQ.0) THEN
         JL=JL+1
         TEMPFI(JL)='YA1IN HI,YB1IN HI,XA2OUT HI,XB2OUT HI'
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',XA2OUT HI,XB2OUT HI'
        END IF
       END IF
      END IF
     END IF
```

```
C     -------------------------------------------------------------------
C     Generate the PART-BLK fault using the GENEV routine.  This is concerned
C     with the tubes for a shell and tube exchanger or both sides for a frame
C     and plate but the temperature effects have to be generated separately.
C     -------------------------------------------------------------------
```

```fortran
      FAUPOR(1,1)=3
      IF (FLAG(10).EQ.0) THEN
       FAUPOR(1,2)=2
       IF (FLAG(9).EQ.0) THEN
        FAUPOR(1,3)=1
        FAUPOR(1,4)=2
       ELSE
        FAUPOR(1,3)=FLAG(3)+3
        FAUPOR(1,4)=FLAG(3)+4
       END IF
      ELSE
       FAUPOR(1,2)=4
       FAUPOR(1,3)=1
       FAUPOR(1,4)=2
       FAUPOR(1,5)=FLAG(3)+3
       FAUPOR(1,6)=FLAG(3)+4
       IF (PHASE.EQ.2) THEN
        FAUPOR(1,2)=5
        FAUPOR(1,7)=FLAG(3)+5
       END IF
      END IF
      CALL GENEV(FAUPOR,TEMPFI)
      JL=JL+1
      IF (RF(1).EQ.0) THEN
       TEMPFI(JL)='F PART-BLK:U1IN'//S2//',T2OUT'//S2
       L1=27
      ELSE
       TEMPFI(JL)='F PART-BLK:T2OUT'//S2
       L1=19
      END IF
      IF (PHASE.EQ.0) THEN
       IF (RF(FLAG(3)+3).EQ.0) THEN
        TEMPFI(JL)=TEMPFI(JL)(1:L1)//',U'//POCH(1)//'IN'//S2
        L1=L1+8
       END IF
       TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//POCH(2)//'OUT'//S2
      ELSE
       TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//VESCH(1)//'VES'//S2
       IF (PHASE.EQ.2) TEMPFI(JL)=TEMPFI(JL)(1:L1+9)//',T'//VESCH(2)//'VES'//S2
      END IF
```

```
C       -----------------------------------------------------------------------------
C       Generate the basic faults that effect the heat transfer rate,
C       FOULING, FROTHING and VAP-BLKT.
C       -----------------------------------------------------------------------------
        JL=JL+1
        IF (RF(1).EQ.0) THEN
         TEMPFI(JL)='F FOULING:U1IN'//S1//',T2OUT'//S1
         L1=26
        ELSE
         TEMPFI(JL)='F FOULING:T2OUT'//S1
         L1=18
        END IF
        IF (PHASE.EQ.0) THEN
         IF (RF(FLAG(3)+3).EQ.0) THEN
          TEMPFI(JL)=TEMPFI(JL)(1:L1)//',U'//POCH(1)//'IN'//S2
          L1=L1+8
         END IF
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//POCH(2)//'OUT'//S2
        ELSE
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//VESCH(1)//'VES'//S2
         L1=L1+9
         IF (PHASE.EQ.2) THEN
          TEMPFI(JL)=TEMPFI(JL)(1:L1)//',T'//VESCH(2)//'VES'//S2
          L1=L1+9
         END IF
         TEMPFI(JL)=TEMPFI(JL)(1:L1)//',Q'//VESCH(3)//'VES LO,Q'//VESCH(3)//'VES NONE'
        END IF
        L1=L1+9
        IF (PHASE.NE.0) THEN
         JL=JL+1
         TEMPFI(JL)='F FROTHING:'//TEMPFI(JL-1)(11:L1)
         JL=JL+1
         TEMPFI(JL)='F VAP-BLKT:'//TEMPFI(JL-1)(12:)
        END IF
C       -----------------------------------------------------------------------------
C       Create the NORMAL state event statement.
C       -----------------------------------------------------------------------------
        BASICF='S NORMAL'
        NDEV=0
        DO 150,I=3,FLAG(7)
         IF (PHASE.EQ.0.AND.(I.EQ.FLAG(3)+3.OR.I.EQ.FLAG(3)+4)) GOTO 150
```

```
        NDEV=NDEV+1
        WRITE(PCH,'(I1)')I
        IF (I.EQ.FLAG(3)+3) THEN
         TEMPEV(NDEV)='G'//PCH//' IN SOME'
        ELSE
         IF (NF(I).EQ.0) THEN
          TEMPEV(NDEV)='Q'//PCH//' OUT SOME'
         ELSE
          TEMPEV(NDEV)='Q'//PCH//' OUT NONE'
         END IF
        END IF
150     CONTINUE
        DO 170,I=3,FLAG(7)
        IF (PHASE.EQ.0.AND.(I.EQ.FLAG(3)+3.OR.I.EQ.FLAG(3)+4)) GOTO 170
        WRITE(PCH,'(I1)')I
        IF (I.EQ.FLAG(3)+3) THEN
         NDEV=NDEV+1
         TEMPEV(NDEV)='R'//PCH//' IN SOME'
        ELSE
         IF (NF(I).EQ.0) THEN
          NDEV=NDEV+1
          TEMPEV(NDEV)='P'//PCH//' OUT SOME'
         END IF
        END IF
170     CONTINUE
        IF (NDEV.NE.0) CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
C       -----------------------------------------------------------------------
C       Create the IMPOSS state event statement.
C       -----------------------------------------------------------------------
        BASICF='S IMPOSS'
        NDEV=0
        DO 200,I=1,FLAG(7)
         IF (RF(I).EQ.1) THEN
          WRITE(PCH,'(I1)')I
          IF (I.EQ.1.OR.I.EQ.FLAG(3)+3) THEN
           TEMPEV(NDEV+1)='G'//PCH//' IN REV'
           TEMPEV(NDEV+2)='R'//PCH//' IN REV'
          ELSE
           TEMPEV(NDEV+1)='Q'//PCH//' OUT REV'
           TEMPEV(NDEV+2)='P'//PCH//' OUT REV'
          END IF
```

```fortran
            NDEV=NDEV+2
            END IF
200     CONTINUE
        IF (PHASE.NE.0) THEN
          TEMPEV(NDEV+1)='Q'//VESCH(3)//'VES REV'
          NDEV=NDEV+1
        END IF
        IF (NDEV.NE.0) CALL EVSTAT(TEMPEV,BASICF,NDEV,TEMPFI,JL)
```
```
C       ---------------------------------------------------------------------------
C       Completion of event statements, now check.
C       ---------------------------------------------------------------------------
```
```fortran
        PRINT3,'THE FOLLOWING ',JL,' EVENT STATEMENTS HAVE BEEN CREATED:'
        CALL CHECK(TEMPFI,JL)
```
```
C       ---------------------------------------------------------------------------
C       Take event statements from the temporary array and place in the FI array.
C       ---------------------------------------------------------------------------
```
```fortran
        K=JL
        JL=0
        DO 250,I=1,K
         IF (TEMPFI(I)(1:1).NE.' ') THEN
          JL=JL+1
          L1=INDEX(TEMPFI(I),' ')
          IF (L1.EQ.0) L1=80
          FI(JL)=TEMPFI(I)(1:L1)
         END IF
250     CONTINUE

        RETURN
        END
```

D-57

```
        SUBROUTINE HEXCHTABLE(VESCH,POCH)

C       --------------------------------------------------------------------------------
C       Subroutine to generate the decision tables for the heat exchanger model.
C       --------------------------------------------------------------------------------
        CHARACTER FU(500)*8,DV(20)*4,VR(20),PT(10)*3,NA*50,AD(25)*80,
     *   FE(45)*80,FI(50)*80,NS*60,MC,FA(5)*80,DT(25)*80,TE(200)*12
        INTEGER IL,FL,DL,JL,PV(10,20),PC(10,20),PM,CM,KL,LG(10),PHASE,
     *   GRAD,NF(10),ET(10),RF(10),FLAG(10)

        COMMON/ONE/FU,DV,VR,PT,NA,AD,FE,FI,NS,MC,FA,DT,TE
        COMMON/TWO/IL,FL,DL,JL,PV,PC,PM,CM,KL
        COMMON/TWELVE/LG,PHASE,GRAD,NF,ET,RF,FLAG

        CHARACTER VESCH(3),POCH(5),PCH,TEMPDT(25)*80
        INTEGER L1


3       FORMAT(/,' ',A,I2,A)
C       --------------------------------------------------------------------------------
C       Initialize the temporary storage strings.
C       --------------------------------------------------------------------------------
        DO 50,I=1,25
50      TEMPDT(I)=DT(I)
        DL=0
C       --------------------------------------------------------------------------------
C       Generate the causes of vessel temperature deviation as reverse flow
C       through the second side outlet port.
C       --------------------------------------------------------------------------------
        IF (PHASE.NE.0) THEN
        IF (RF(FLAG(3)+4).EQ.0) THEN
        TEMPDT(DL+1)='V U'//POCH(2)//' OUT HI V Q'//POCH(2)//' OUT REV T T'//VESCH(1)//' VES HI'
        TEMPDT(DL+2)='V U'//POCH(2)//' OUT LO V Q'//POCH(2)//' OUT REV T T'//VESCH(1)//' VES LO'
        DL=DL+2
        END IF
        IF (PHASE.EQ.2.AND.RF(FLAG(3)+5).EQ.0) THEN
        TEMPDT(DL+1)='V U'//POCH(3)//' OUT HI V Q'//POCH(3)//' OUT REV T T'//VESCH(2)//' VES HI'
        TEMPDT(DL+2)='V U'//POCH(3)//' OUT LO V Q'//POCH(3)//' OUT REV T T'//VESCH(2)//' VES LO'
        DL=DL+2
        END IF
```

```
C       -----------------------------------------------------------------------
C       Generate the causes of vessel temperature deviation as reverse flow
C       through the second side additional port(s).
C       -----------------------------------------------------------------------
        DO 70,I=1,FLAG(5)
        J=5
        IF (.EQ.1.AND.(FLAG(5).EQ.2.ORFLAG(6).EQ.1)) J=4
        TEMPDT(DL+1)='V U'//POCH(J)//'OUT HI V Q'//POCH(J)//'OUT REV T T'//VESCH(2)//'VES HI'
        IF (J.EQ.5) TEMPDT(DL+1)(27:27)=VESCH(1)
        TEMPDT(DL+2)='V U'//POCH(J)//'OUT LO V Q'//POCH(J)//'OUT REV T T'//VESCH(2)//'VES LO'
        IF (J.EQ.5) TEMPDT(DL+2)(27:27)=VESCH(1)
70      DL=DL+2
C       -----------------------------------------------------------------------
C       Generate the causes of vessel composdition deviation as reverse flow
C       through the second side outlet port.
C       -----------------------------------------------------------------------
        IF (RF(FLAG(3)+4).EQ.0) THEN
        IF (MC.EQ.'N')THEN
         TEMPDT(DL+1)='V Y'//POCH(2)//'OUT HI V Q'//POCH(2)//'OUT REV T X'//VESCH(1)//'VES HI'
         TEMPDT(DL+2)='V Y'//POCH(2)//'OUT LO V Q'//POCH(2)//'OUT REV T X'//VESCH(1)//'VES LO'
         DL=DL+2
        ELSE
         TEMPDT(DL+1)='V YA'//POCH(2)//'OUT HI V Q'//POCH(2)//'OUT REV T XB'//VESCH(1)//'VES LO'
         TEMPDT(DL+2)='V YA'//POCH(2)//'OUT LO V Q'//POCH(2)//'OUT REV T XB'//VESCH(1)//'VES HI'
         TEMPDT(DL+3)='V YB'//POCH(2)//'OUT HI V Q'//POCH(2)//'OUT REV T XB'//VESCH(1)//'VES HI'
         TEMPDT(DL+4)='V YB'//POCH(2)//'OUT LO V Q'//POCH(2)//'OUT REV T XB'//VESCH(1)//'VES LO'
         DL=DL+4
        END IF
        END IF
        IF (PHASE.EQ.2.AND.RF(FLAG(3)+5).EQ.0) THEN
        IF (MC.EQ.'N')THEN
         TEMPDT(DL+1)='V Y'//POCH(3)//'OUT HI V Q'//POCH(3)//'OUT REV T X'//VESCH(2)//'VES HI'
         TEMPDT(DL+2)='V Y'//POCH(3)//'OUT LO V Q'//POCH(3)//'OUT REV T X'//VESCH(2)//'VES LO'
         DL=DL+2
        ELSE
         TEMPDT(DL+1)='V YA'//POCH(3)//'OUT HI V Q'//POCH(3)//'OUT REV T XA'//VESCH(1)//'VES HI'
         TEMPDT(DL+2)='V YA'//POCH(3)//'OUT LO V Q'//POCH(3)//'OUT REV T XA'//VESCH(1)//'VES LO'
         TEMPDT(DL+3)='V YB'//POCH(3)//'OUT HI V Q'//POCH(3)//'OUT REV T XA'//VESCH(1)//'VES LO'
         TEMPDT(DL+4)='V YB'//POCH(3)//'OUT LO V Q'//POCH(3)//'OUT REV T XA'//VESCH(1)//'VES HI'
```

```
          DL=DL+4
        END IF
      END IF
C     ----------------------------------------------------------------
C     Generate the causes of vessel composition deviation as reverse flow
C     through the second side additional port(s).
C     ----------------------------------------------------------------
      DO 80,I=1,FLAG(5)
        J=5
        IF (I.EQ.1.AND.(FLAG(5).EQ.2.ORFLAG(6).EQ.1)) J=4
        IF (MC.EQ.'N')THEN
        TEMPDT(DL+1)='V Y'//POCH(J)//'OUT HI V Q'//POCH(J)//'OUT REV T X'//VESCH(2)//'VES HI'
          IF (J.EQ.5) TEMPDT(DL+1)(27:27)=VESCH(1)
        TEMPDT(DL+2)='V Y'//POCH(J)//'OUT LO V Q'//POCH(J)//'OUT REV T X'//VESCH(2)//'VES LO'
          IF (J.EQ.5) TEMPDT(DL+2)(27:27)=VESCH(1)
        ELSE
        TEMPDT(DL+1)='V YD'//POCH(J)//'OUT HI V Q'//POCH(J)//'OUT REV T XD'//VESCH(2)//'VES HI'
          IF (J.EQ.5) TEMPDT(DL+1)(29:29)=VESCH(1)
        TEMPDT(DL+2)='V YD'//POCH(J)//'OUT LO V Q'//POCH(J)//'OUT REV T XD'//VESCH(2)//'VES LO'
          IF (J.EQ.5) TEMPDT(DL+1)(29:29)=VESCH(1)
        END IF
80      DL=DL+2
      END IF
C     ----------------------------------------------------------------
C     Completion of decision tables, now check.
C     ----------------------------------------------------------------
      IF (DL.EQ.0) THEN
        PRINT*,' '
        PRINT*,'NO DECISION TABLES ARE REQUIRED FOR THE MODEL.'
      ELSE
        PRINT3,'THE FOLLOWING ',DL,' DECISION TABLES HAVE BEEN CREATED:'
        CALL CHECK(TEMPDT,DL)
C     ----------------------------------------------------------------
C     Take decison tables from the temporary array and place in the DL array.
C     ----------------------------------------------------------------
      K=DL
      DL=0
      DO 150,I=1,K
        IF (TEMPDT(I)(1:1).NE.' ') THEN
          DL=DL+1
          L1=INDEX(TEMPDT(I),' ')
```

```fortran
          IF (L1.EQ.0) L1=80
          DT(DL)=TEMPDT(I)(1:L1)
         END IF
150     CONTINUE
        END IF


        RETURN
        END
```

Appendix E

Divider, Header And Closed Valve Models

E.1 Divider Model

```
1)       MODEL   NUMBER    NAME
          116                DIVIDER (2 OUTPUTS)


NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 3
NO. OF PROPAGATION EQUATIONS: 12
NO. OF EVENT STATEMENTS: 14
NO. OF DECISION TABLES: 13
NO. OF FAILURE MODES: 1


2)       ENGINEERING ASSUMPTIONS AND DESCRIPTIONS


A UNIT TO SPLIT A STREAM INTO 2 OUTPUT STREAMS.
PORTS 2 & 3 ARE THE OUTPUT STREAM PORTS.
NO PIPE TYPE FAULTS.


3)       PROPAGATION EQUATIONS


G1IN= F( Q1IN, Q2OUT, Q3OUT)
R1IN= F( R2OUT, R3OUT)
U1IN= F( U2OUT, U3OUT)
Y1IN= F( Y2OUT, Y3OUT)
Q2OUT= F( G2OUT, G1IN/ Q3OUT)
T2OUT= F( T1IN)
X2OUT= F( X1IN)
P2OUT= F( P1IN, -R3OUT)
Q3OUT= F( G3OUT, G1IN/ Q2OUT)
T3OUT= F( T1IN)
X3OUT= F( X1IN)
P3OUT= F( P1IN, -R2OUT)


4)       EVENT STATEMENTS


V  G1IN HI : Q2OUT SOME, Q3OUT SOME
V  R1IN HI : P2OUT SOME, P3OUT SOME
V  Q2OUT HI : Q3OUT REV, Q3OUT NONE
V  Q2OUT NONE : G1IN LO, Q3OUT HI
V  Q2OUT REV : G1IN LO, G1IN NONE, Q3OUT HI, Q3OUT SOME
V  Q3OUT HI : Q2OUT REV, Q2OUT NONE
V  Q3OUT NONE : G1IN LO, Q2OUT HI
```

V Q3OUT REV : G1IN LO, G1IN NONE, Q2OUT HI, Q3OUT SOME
V R2OUT HI : P3OUT NONE, P3OUT REV
V R2OUT NONE : R1IN LO, P3OUT HI
V R2OUT REV : R1IN LO, R1IN NONE, P3OUT HI, P3OUT SOME ·
V R3OUT HI : P2OUT NONE, P2OUT REV
V R3OUT NONE : R1IN LO, P2OUT HI
V R3OUT REV : R1IN LO, R1IN NONE, P2OUT HI, P2OUT SOME

5)   DECISION TABLES

V Q2OUT NONE V Q3OUT NONE T G1IN NONE
V R2OUT NONE V R3OUT NONE T R1IN NONE
V R2OUT NOP V R3OUT NOP T R1IN NOP
V P1IN NOR V R3OUT NONE T P2OUT NOR
V P1IN NOR V R2OUT NONE T P3OUT NOR
V U3OUT LO V Q3OUT REV T T2OUT LO
V U3OUT HI V Q3OUT REV T T2OUT HI
V Y3OUT LO V Q3OUT REV T X2OUT LO
V Y3OUT HI V Q3OUT REV T X2OUT HI
V U2OUT LO V Q2OUT REV T T3OUT LO
V U2OUT HI V Q2OUT REV T T3OUT HI
V Y2OUT LO V Q2OUT REV T X3OUT LO
V Y2OUT HI V Q2OUT REV T X3OUT HI

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: N/A
NO MULTI-COMPONENT FEATURES

E.2 Header Model

1)   MODEL   NUMBER   NAME
            117   HEADER (2 INPUTS)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 3
NO. OF PROPAGATION EQUATIONS: 12
NO. OF EVENT STATEMENTS: 16
NO. OF DECISION TABLES: 7
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

HEADER COMBINING TWO INPUT STREAMS TO GIVE ONE OUTPUT STREAM.
PORTS 1 AND 3 ARE THE INLETS, PORT 2 IS THE OUTLET.
THERE IS NO SPARE INLET CAPACITY. THERE ARE NO PIPE TYPE FAULTS.

3)   PROPAGATION EQUATIONS

G1IN= F(Q1IN, Q2OUT, -G3IN)
R1IN= F(R2OUT, -P3IN)
U1IN= F(U2OUT, T3IN)
Y1IN= F(Y2OUT, X3IN)
Q2OUT= F(G1IN, G2OUT, G3IN)
T2OUT= F(T1IN, T3IN)
X2OUT= F(X1IN, X3IN)
P2OUT= F(P1IN, P3IN)
G3IN= F(Q3IN, Q2OUT, -G1IN)
R3IN= F(R2OUT, -P1IN)
U3IN= F(U2OUT, T1IN)
Y3IN= F(Y2OUT, X1IN)

4)   EVENT STATEMENTS

V G1IN HI : G3IN NONE, G3IN REV
V G1IN NONE : G3IN HI, Q2OUT LO
V G1IN REV : G3IN HI, G3IN SOME, Q2OUT LO, Q2OUT NONE
V G3IN HI : G1IN NONE, G1IN REV
V G3IN NONE : G1IN HI, Q2OUT LO
V G3IN REV : G1IN HI, G1IN SOME, Q2OUT LO, Q2OUT NONE
V P1IN HI : R3IN NONE
V P1IN NONE : P2OUT LO
V P1IN REV : P2OUT LO, P2OUT NONE, R3IN HI, R3IN SOME
V P3IN HI : R1IN NONE
V P3IN NONE : P2OUT LO
V P3IN REV : R1IN HI, R1IN SOME, P2OUT LO, P2OUT NONE
V Q2OUT HI : G1IN SOME, G3IN SOME

V Q2OUT LO : G1IN REV, G3IN REV
V P2OUT HI : R1IN SOME, R3IN SOME
V P2OUT LO : R1IN REV, R3IN REV

5)   DECISION TABLES

V G1IN NONE V G3IN NONE T Q2OUT NONE
V P1IN NONE V P3IN NONE T P2OUT NONE
V P1IN NOR V P3IN NOR T P2OUT NOR
V R2OUT NONE V P3IN NOR T R1IN NONE
V R2OUT NONE V P1IN NOR T R3IN NONE
V R2OUT NOP V P3IN NONE T R1IN NOP
V R2OUT NOP V P1IN NONE T R3IN NOP

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: N/A
NO MULTI-COMPONENT FEATURES

E.3 Closed Valve Model

1)   MODEL   NUMBER   NAME
                  3   HAND VALVE (CLOSED)

NO. OF ENG. ASSUMPTIONS/DESCRIPTIONS: 2
NO. OF PROPAGATION EQUATIONS: 0
NO. OF EVENT STATEMENTS: 7
NO. OF DECISION TABLES: 16
NO. OF FAILURE MODES: 1

2)   ENGINEERING ASSUMPTIONS AND DESCRIPTIONS

A VALVE CLOSED DURING NORMAL OPERATION.
A(DUMMY) REPRESENTS THE VALVE BEING OPEN.

3)   PROPAGATION EQUATIONS

N/A

4)   EVENT STATEMENTS

F HV-F-OP : A(DUMMY)
O HV-D-OP : A(DUMMY)
S NORMAL : G1IN NONE, Q2OUT NONE, R1IN NONE, R1IN NOP, P2OUT NONE, P2OUT NOR
V Q1IN SOME : G1IN SOME
V Q1IN REV : G1IN REV
V G2OUT SOME : Q2OUT SOME
V G2OUT REV : Q2OUT REV

5)   DECISION TABLES

I A(DUMMY) V Q2OUT SOME T Q1IN SOME
I A(DUMMY) V Q2OUT REV T Q1IN REV
I A(DUMMY) V U2OUT LO T U1IN LO
I A(DUMMY) V U2OUT HI T U1IN HI
I A(DUMMY) V Y2OUT LO T Y1IN LO
I A(DUMMY) V Y2OUT HI T Y1IN HI
I A(DUMMY) V R2OUT SOME T R1IN SOME
I A(DUMMY) V R2OUT REV T R1IN REV
I A(DUMMY) V G1IN SOME T G2OUT SOME
I A(DUMMY) V G1IN REV T G2OUT REV
I A(DUMMY) V T1IN LO T T2OUT LO
I A(DUMMY) V T1IN HI T T2OUT HI
I A(DUMMY) V X1IN LO T X2OUT LO
I A(DUMMY) V X1IN HI T X2OUT HI
I A(DUMMY) V P1IN SOME T P2OUT SOME

I A(DUMMY) V P1IN REV T P2OUT REV

6)   SUPPLEMENTARY INFORMATION

NORMAL STATE: CLOSED - NO FLOW THROUGH UNIT.

NO MULTI-COMPONENT FEATURES

E-6