POLITECNICO DI TORINO

DOCTORATE SCHOOL
Ph.D. in Metrology: Measuring science and Technique – XXVI doctoral cycle

PhD Thesis

Process Intensification Vs. Reliability

Gabriele Baldissone

Tutor
Prof. Micaela Demichela

PhD course coordinator
Prof. Franco Ferraris

February 2014

3 Methods of risk assessment

A risk assessment can be carried on performing the following steps:

1. Identification of hazards;
2. Estimation of the probability of occurrence;
3. Estimation of the consequences;
4. Risk assessment.

There are several possible methods for the risks assessment; typically they are classified on the basis of the type of result that they produce:

- Qualitative: These methods allow to identify the possible accidents and their causes and consequences, but don't give information on the entity of the risk;
- Semi-quantitative: These methods result in a risk index which, although in an indicative way, allows the estimation of the risk;
- Quantitative: These methods are more complex but allow to calculate the probability of occurrence and damage of undesired events, producing a numerical estimation of the risk.

The methodologies for the identification of hazards are the most diverse and depend on the desired result. The methods most commonly used in the process industry are:

- Checklists;
- What If? Analysis;
- Failure Modes, Effects and Criticality Analysis (FMECA);
- Hazard and Operability Studies (HAZOP);

The methodologies more used for the quantification of the probability, are:

- Event Trees (ET);
- Fault Trees (FT);

The estimation of the consequences is a very complex activity and depends on the type of accident hypothesized (release of energy or matter, loss of functioning of the system, ...), the type of desired result (map the extent of the damage, the economic cost of the malfunction, ...), the type of hypothesized consequences (damage to persons, damage to structures, economic damage, ...).

Finally, there is the last stage of the risk assessment: it is the comparison of the value of risk with the tolerable limits provided by law or chosen by the company (if more restrictive than legal limits). In case the reference limits are exceeded, immediate measures must be taken to reduce the risk. But even if risk is below the limit of tolerance is necessary to manage the residual risk.

3.1 Hazard identification

Some of the most common methods for "hazard identification" are descripted in the following chapter; they allow to identify all possible hazards and all possible ways that can bring to the accident hypothesized. The different methods can be adopted depending on the degree of knowledge about the process and the type of desired results.

3.1.1 Checklist

The checklist methodology is one of the most commonly used. The checklists are a set of questions with answer like yes or no. On the basis of the number and order of the affirmative or negative answers, it is possible to identify the various hazards. The type of the result obtained by this method depends on the degree of knowledge about the process.
The quality of the checklist depends on the experience of the person who prepares and applies it; however, literature provides several examples that could be adapted to the specific use.
The use of the checklist alone is suitable only for very simple cases or with well-known technology. In other cases, the use of this methodology can be useful for collecting information, useful to employ more complex techniques.
(Mannan, 2005) wrote:
“One of the most simplistic tools of hazard identification is the checklist. Like a standard or a code of practice, a checklist is a means of passing on lessons learned from experience. It is impossible to envisage high standards in hazard control unless this experience is effectively utilized. The checklist is one of the main tools available to assist in this.
Checklists are applicable to management systems in general and to a project throughout all its stages. Obviously the checklist must be appropriate to the stage of the project, starting with checklists of basic materials properties and process features, continuing on to checklists for detailed design and terminating with operations audit checklists.
There are a large number of checklists given in the literature - indeed a paper on practical engineering is quite likely to include a checklist.

A checklist should be used for just one purpose only - as a final check that nothing has been neglected. Also, it is more effective if the questions cannot be answered by a simple ‘yes’ or ‘no’ but require some thought in formulating an answer.
Checklists are effective only if they are used. There is often a tendency for them to be left to gather dust on the shelf. This is perhaps part of the reason for the development of other techniques such as HAZOP studies, as described below.”

3.1.2 What if? Analysis

The methodology of "What if?" consists in asking what happens if a certain event happens. This type of analysis is very creative, but, if it is not well organized, it can bring to overestimate or underestimate some aspects, or the analyst could not identify all possible hazards In order to perform this type of analysis, a team work is required to respond in appropriate way to the questions.. The best result can be achieved by coupling this methodology with the checklist: the "what if?" allows to investigate the behavior of the system and, at the same time, the checklist gives a reasonable assurance of a complete analysis.
(Mannan, 2005) wrote:
“TheWhat If ? method involves asking a series of questions beginning with this phrase as a means of identifying hazards. Apart from checklists, What If ? analysis is possibly the oldest method of hazard identification. The method is to ask questions such as,
What if the pump stops?
What if the temperature sensor fails?
The questions posed need not necessarily all start with What If ?; other phrases may be used.
The method involves review of the whole design by a team using questions of this type, often using a list of pre-determined questions. …
Figure 31 gives an example of the results from a What If ? study reported by (Kavianian, Rao and Brown, 1992).

What if	Consequence/hazard	Recommendations
Coolant pump to reactor fails	Runaway condition in reactor explosion/fatality	• Provide accurate temperature monitoring in reactor • Employ backup pump/high temperature alarm • Relieve reactor pressure in reactor through • automatic control to stop reactions • Provide automatic shut off of ethylene flow
Coolant temperature to racket is high	Eventual runaway condition in reactor	• Provide adequate temperature control on coolant line • Use heat exchanger flow control to adjust inlet temperature
Runaway condition in reactor	Explosion; fire/fatality	• Provide adequate temperature control on coolant line • Use heat exchanger flow control to adjust inlet temperature • Install rupture disk/relief valve to relieve pressure to stop reactions • Emergency shut-down procedure
Recycle gas compressor 1 or 2 fails	None likely	• Provide spare compressor or shut-down procedure
Melt pump fails	High level in reactor causing more polymerization: runaway reaction eventually exceeds design pressure	• Provide level and flow control schemes to activate spare pump or shut the flow of monomer • Shut down procedure if no spare pump
Leak at suction or discharge of compressors	Fire; explosion	• Use monitoring devices to ensure no flammable gas is released
Ethylene leaks out of process lines	Fire; explosion	• Provide adequate flammable gas monitoring devices
Monomer/initiator ratio out of control	Eventual runaway reaction causing fire and explosion	• Provide flow control on the initiator and monomer lines

Figure 31: What if? method: results for a high pressure/low density polyethylene plant (Kavianian, Rao and Brown, 1992)

3.1.3 Failure Modes Effects and Criticality Analysis

The Failure Modes Effects Analysis (FMEA), with its variant Failure Modes Effects and Criticality Analysis (FMECA), are two of the most used techniques for the identification of hazards in the complex activities. In literature, there are many books that treat this methodology, in example: (Recht, 1966; Taylor, 1973; Taylor, 1974; Taylor, 1975; Himmelblau, 1978; Lambert, 1978; Green, 1983; Flothmann and Mjaavatten, 1986; Moubray, 1991; Kavianian, Rao and Brown, 1992; Scott and Crawley, 1992; Goyal, 1993).

The methodology consists in identifying all the possible failure modes of the equipments with a consequent investigation of the possible causes and effects on the system. All this is formalized with the compilation of a table (see Figure 32).

The performance of this type of analysis is encoded in an appropriate technical standard in BS EN 60812:2006, summarized by (Mannan, 2005):

"Guidance on FMEA is given in BS 5760[1] *Reliability of Systems, Equipment and Components*, Part 5:1991 *Guide to Failure Modes, Effects and Criticality Analysis* (FMEA and FMECA). BS 5760: Part 5: 1991 deals with the purposes, principles, procedure and applications of FMEA, with its limitations and its relationship to other methods of hazard identification, and gives examples.

FMECA is an enhancement of FMEA in which a criticality analysis is performed. Criticality is a function of the severity of the effect and the frequency with which it is expected to occur. The criticality analysis involves assigning to each failure mode a frequency and to each failure effect a severity.

The purpose of an FMEA is to identify the failures which have undesired effects on system operation. Its objectives include: (1) identification of each failure mode, of the sequence of events associated with it and of its causes and effects; (2) a classification of each failure mode by relevant characteristics, including detectability, diagnosability, testability, item replaceability, compensating and operating provisions; and, for FMECA, (3) an assessment of the criticality of each failure mode.

The standard lists the information required for an FMEA, under the headings: (1) system structure; (2) system initiation, operation, control and maintenance; (3) system environment; (4) system modeling; (5) system software; (6) system boundary; (7) system functional structure; (8) system functional structure representation; (9) block diagrams and (10) failure significance and compensating provisions.

[1] The drafting of the text that referenced in the standard the object was still valid but now has been replaced by BS EN 60812:2006, that does not change what descripted here.

Component	*Failure or error mode*	*Effects on other system components*	*Effects on whole system*	*Hazard class*[a]				*Failure frequency*	*Detection methods*	*Compensating provisions and remarks*
				1	*2*	*3*	*4*			
Pressure relief valve	Jammed open	Increased operation of temperature sensing controller, and gas flow, due to hot water loss	Loss of hot water; greater cold water input, and greater gas consumption	1				Reasonably probable	Observe at pressure-relief valve	Shut off water supply, reseat or replace relief valve
	Jammed closed	None	None	1				Probable	Manual testing	Unless combined with other component failure, this failure has no consequence
Gas valve	Jammed open	Burner continues to operate. Pressure-relief valve opens	Water temperature and pressure increase. Water → steam			3		Reasonably probable	Water at faucet too hot. Pressure-relief valve open (observation)	Open hot water faucet to relieve pressure. Shut off gas supply. Pressure-relief valve compensates
	Jammed closed	Burner ceases to operate	System fails to produce hot water	1				Remote	Observe at output (water temperature too low)	
Temperature measuring and comparing device	Fails to react to temperature rise above preset level	Controller gas valve, burner continue to function 'on'. Pressure-relief valve opens	Water temperature too high. Water → steam			3		Remote	Observe at output (faucet)	Pressure-relief valve compensates. Open hot water faucet to relieve pressure. Shut off gas supply
Temperature measuring and comparing device	Fails to react to temperature drop below preset level	Controller, gas valve, burner continue to function off	Water temperature too low	1				Remote	Observe at output faucets	
Flue	Blocked	Incomplete combustion at burner	Inefficiency. Production of toxic gases				4	Remote	Possibly smell products of incomplete combustion	No compensation built in. Shut-down system
Pressure-relief valve and gas valve	Jammed closed	Burner continues to operate, pressure increases	Increased pressure cannot bleed at relief valve.				4	Probable	Manual testing of relief valve	Open hot water faucet. Shut off gas supply.
	Jammed open		Water → steam If pressure cannot back up cold water inlet, system may rupture violently					Reasonably probable	Observe water output	Pressure might be able to back into cold water supply, providing pressure in supply is not greater than failure pressure of system
								Reasonably probable	Temperature too high	

[a] 1, Negligible effects; 2, marginal effects; 3, critical effects; 4, catastrophic effects.

Figure 32: Failure modes and effects analysis: result for a process plant (after (Recht, 1966; Himmelblau, 1978))

Core information on the items studied is the (1) name, (2) function, (3) identification, (4) failure modes, (5) failure causes, (6) failure effects on system, (7) failure detection methods, (8) compensating provisions, (9) severity of effects and (10) comments.

The main documentation used in an FMEA is the functional diagram. Use may also be made of reliability block diagrams.

The identification of the failure modes, causes and effects is assisted by the preparation of a list of the expected failure modes in the light of (1) the use of the system, (2) the element involved, (3) the mode of operation, (4) the operation specification, (5) the time constraints and (6) the environment.

The failure modes may be described at two levels: generic failure modes and specific failure modes. The standard gives as an example a set of generic failure modes: (1) failure during operation, (2) failure to operate at a prescribed time, (3) failure to cease operation at a prescribed time and (4) premature operation. As examples of specific failure modes, the standard gives: (1) cracked/fractured, (2) distorted, (3) undersized, and so on.

The failure causes associated with each mode should be identified. BS 5760 gives a checklist of potential failure causes under the headings: (1) specification, (2) design, (3) manufacture, (4) installation, (5) operation, (6) maintenance, (7) environment and (8) uncontrollable forces.

The failure effects involve changes in the operation, function or status of the system and these should be identified by the analysis. Failure effects can be classified as local or as end effects. Local effects refer to the consequences at the level of the element under consideration and end effects to those at the highest level of the system.

Where FMEA is to be applied within a hierarchical structure, it is preferable to restrict it to two levels only and to perform separate analyses at the different levels. The failure effects identified at one level may be used as the failure modes of the next level up, and so on.

FMEA is an efficient method of analyzing elements which can cause failure of the whole, or of a large part, of a system. It works best where the failure logic is essentially a series one. It is much less suitable where complex logic is required to describe system failure.

FMEA is an inductive method. A complementary deductive method is provided by fault tree analysis, which is the more suitable where analysis of complex failure logic is required.

BS 5760: Part 5: 1991 states that FMEA can be a laborious and inefficient process unless judiciously applied. The uses to which the results are to be put should be defined.

FMEA should not be included in specifications indiscriminately."

3.1.4 Hazard and Operability Studies

Another method of hazard identification is the Hazard and Operability Analysis (HAZOP). This type of analysis is carried out by a multi-disciplinary group with the purpose of reviewing the entire production process, in order to identify possible dangers and potential problems using an organized approach based on guide words. This approach is based on the analysis of criticality of the system.
The methodology is largely reported in the literature: (Crawley, Preston and Tyler, 2000; Klertz, 1983; Klertz, 1986; Kletz, 1992; CENTER FOR CHEMICAL PROCESS SAFETY, 1992; Knowlton, 1992; Gibson, 1976; Lawley, 1974; Kavianian, Rao and Brown, 1992).
This type of analysis can be applied to the flow diagram (PFD) and to detailed piping and instrument diagrams (P&ID); logically, the accuracy of the results depends on the knowledge available about the process available.
The completeness of the analysis strongly depends on the choice of the level of project development to which the technique has to be applied: if the HAZOP is performed at a too early stage, during the analysis some information may still be missing and, at more advanced steps, any corrective actions would be very expensive.
This methodology was developed in the sixties by ICI as a method for the analysis of critical situations (Binsted, 1960; Elliott and Owen, 1968).
The basic concept on which the HAZOP analysis is based is to ask for each part of the process which types of deviations from design conditions are possible, and what can be the possible causes and consequences. This approach is made systematic by using guide words (Mannan, 2005):
"The basic concept of the HAZOP study is to take a full description of the process and to question every part of it to discover what deviations from the intention of the design can occur and what the causes and consequences of these deviations may be. This is done systematically by applying suitable guidewords."
The possible deviations can regard all aspects and parameters involved in the process such as:

- material;
- activity;
- equipment;
- source;
- destination;
- time;
- space.

To the process parameters described above, the guide words have to be applied, this are a coded system to help the analyst to find and analyze the deviations that may be present in the system. The guide words are:

- NO or NOT, Negation of intention;
- MORE, Quantitative increase;
- LESS, Quantitative decrease;
- AS WELL AS, Qualitative increase;
- PART OF, Qualitative decrease;
- REVERSE, Logical opposite of intention;
- OTHER THAN, Complete substitution.

Depending on the different variables, variations of the guide words may be applied allows maintaining the sense, for example: the MORE or LESS can be applied in the case of durations or frequencies, but in the case of absolute times is more applicable SOONER and LATER. The same things can be done for other variables.

Figure 33 shows a non-exhaustive list of possible guide words.

The performance of HAZOP analysis is structured according to the following procedure (Mannan, 2005):

"(1) Definition of objectives and study scope;

(2) Selection of multi-disciplined team;

(3) Preparation;

(4) Conduct/facilitate;

(5) Record/document study;

(6) Preparation of HAZOP study report."

The first step of the analysis is related to the definition of the objectives.

After this, it is possible to proceed with the choice of the team members: this step is one of the most delicate moments of the whole analysis, because the choice of the people influences the points on which the attention will be focused. Therefore, the choice of the group depends on the purposes the analysis. However, in generally is better to choose a multi-disciplinary team to have a complete picture. (Mannan, 2005):

"A multi-disciplinary team is used to conduct the HAZOP study. The team should contain people from design and from operations who can cover the main relevant disciplines, who are senior

enough to make on-the-spot decisions and who personally attend all the meetings, but the team size should be kept fairly small.

Process variables	Flow	High; low; reverse; two-phase; leak
	Level	High; low
	Temperature	High; low
	Pressure	High; low; vacuun
	Load	High; low
	Viscosity	High; low
	Quality	Concentration/proportion; impurities; cross-contamination, side reactions; particle size; viscosity; water content; inspection and testing
Plant states	Commissioning and start-up	Statutory approvals; compliance checking; sequence of steps; supervision, training
	Shut-down	Isolation; purging; cleaning
	Breakdown	Fail-safe response; loss of utilities; emergency procedures
Production	Throughput	Sources of unreliability/unavaliability; bottlenecks
	Efficiency	
Materials of construction		Corrosion; erosion; wear; chilling; compatibility; sparking
Plant layout	Access	Operation; maintenance; escape; emergency response
	Space	(For housekeeping, work in progress, escape) Cramped; wasted
	Electrical safety	Hazardous area classification; electrostatic discharge and earthing: lightning protection
Utilities		Air; nitrogen; steam; electrical power; water: process, hot/cold, demineralized, drinking: drainage
Machinery	Machinery	Overload; malfunction; foreign body; rotation: fast, slow; jamming/seizing; frictional overheating; mechanical failure/fracture; impact; valve blockage: mechanical, low temperature; interlocks
	Assembly	Component missing, extra, or wrong; assembly sequence wrong; screwed wrong: (too tight, too loose): crossed/contaminated thread; shielding
	Incompatibility	Tools and equipment material: foreign objects
Materials handling	Speed	Fast; slow; unbalanced
	Packaging	Filling (over, under); damage (external, internal); poor sealing: legal requirements; labelling
	Physical damage	Impact; dropping; vibration
	Stoppages	Breakage; blockage; jamming; loss of feed; loss of packaging; advance warning; rectification
	Direction	Upwards; downwards: to one side; reverse
	Spillage	Spillage: into product, into other materials. Spillage: outside equipment, outside plant
	Location	Wrong: vertical, horizontal; orientation
Stockholding	Stockholding	Failed stock rotation: poor storage (water, vermin); storage of consumables (spares, tools, paints/solvents); storage of raw materials
Hazards	Fire and explosion	Prevention; detection; separation; protection; control
	Toxicity	Acute; long-term; ventilation; control
	Ignition	Friction: impact: static electricity; degraded electricals; failed earthing; mechanical sparling: misalignment; high temperature; dropped objects
	Environment	Housekeeping; dust, spillage, scrap/residues: humidity high, low; ventilation failure
	Environmental control	Effluents: gaseous, liquid, solid; noise; monitoring
Targets	Severity	Quantity exposed; protection and adequacy; venting; personnel escape: unplanned exposure: dust, maintenance, protection malfunction, propagation via duct
	Operator injury	Heavy lifting; repetitive motion; exposure: dust, fume, heat; falling, slipping, tripping
Reactions	Reaction rate	Fast; slow
Control and protection	Control	Sensor and display location; response speed; interlocks
	Protection	Response speed; element common with control loop: sensor, valve, operator; remote actuation; venting; testing
	Safety equipment	Personal equipment; showers
Sequences	Timing	Duration/dwell; rate of approach; sequence; start too early, late: stop too early, late
Testing	Testing	Raw materials; products; equipment; protective instrumentation: alarms, interlocks, trips; protective equipment

Figure 33: Hazard and operability studies: some additional parameters and guidewords (Mannan, 2005)

A typical team might comprise the study leader (facilitator), the project engineer, a process engineer, an instrument engineer and the commissioning manager. Other personnel who are often included, depending on the nature of the project, are a chemist, a civil engineer, an electrical engineer, a materials technologist, an operations supervisor, an equipment supplier's representative, and so on.

The technical team members provide the technical input in response to the guidewords. They are also able to amplify the information about the plant design given in the plant diagrams, operating instructions, etc.

The role of the study leader is to act as a facilitator to bring to bear the expert knowledge of the technical team members in a structured interaction. It is not his role to identify hazards and operability problems, but rather to ensure that such identification takes place.

The study leader should be someone not directly involved in the design, but with skills as a HAZOP leader. The effectiveness of a HAZOP study is highly dependent on the skill of the study leader. Trained and experienced leaders are crucial.

The study leader is responsible for the definition of the project HAZOP; for the preparation of the meetings to be held; for arranging the schedules of meetings, and hence their timing and pacing; for assembling an appropriate team and ensuring that they understand their role, and receive training if necessary; for the provision to each study meeting of the necessary documents and other information; for the conduct and recording of the meetings; and for follow-up of matters raised during the meetings.

At the definition stage, the study leader should ensure that there is a satisfactory liaison with the client such that the latter will follow-up the results emerging from the study.

In the preparation stage, the study leader should review the extent to which the plant under consideration is similar to one already studied and how this should affect the study to be conducted. In many cases, the design is not a completely new one but constitutes a modification of an existing design. The study leader will then define the features, which are novel - raw materials, plant equipment, materials of construction, environmental conditions, etc. – and decide whether examination of certain parts can be omitted as unproductive. He should also ensure as part of the preparation that the necessary information is available and is correct.

The most elusive skill of the study leader lies in the conduct of the HAZOP meeting itself. One essential requirement is to ensure that the examination neither becomes too superficial nor gets bogged down in detail so that it identifies all the hazards but within a reasonable time-scale. Another is to manage the personal interactions between the team members, to obtain balanced

contributions and to minimize the effect on individuals when the design is subject to criticism. These requirements are easily stated but constitute a significant skill."

Then the information need to every on the analysis have to be collected, e.g.: (Mannan, 2005)

"Closely associated with these are checks on (1) information still lacking, (2) particular equipment, (3) supplier information, (4) plant phases (start-up, shut-down) and (5) maintenance procedures, and on entities to be protected such as (1) persons working on the unit, (2) others on the site, (3) the public, (4) the plant and (5) the environment."

Then, the analysis can be carried on according to the diagram shown in Figure 34.

The analysis begins with the choice of the equipment and the definition of its function and incoming and outgoing flows. For each stream, all the words guide must be applied: each guide word tells what kind of deviation is being analyzed, and what is its level. In general, it is better to assume a big deviation, that the control system is not able to afford. For each deviation possible causes, consequences and dangers has to be identified. The analysis proceed in this way for all the lines and auxiliary devices in the part of plant under review. After finishing, it is possible to move to the next part of plant.

"The study uses a formal, even mechanistic, approach and the questions raised may in some cases appear unrealistic or trivial. It is important to emphasize, however, that the approach is intended as an aid to the imagination of the team in visualizing deviations and their causes and consequences. The effectiveness of the technique depends very much on the spirit in which it is done."

The final result of the analysis is a series of tables (an example is shown in Figure 35) that encodes the results obtained, puts in evidence the possible deviations with the possible causes and consequences.

"Limitations of the analysis are of two kinds. The first type arises from the assumptions underlying the method and is an intended limitation of scope. In its original form, the method assumes that the design has been carried out in accordance with the appropriate codes. Thus, for example, it is assumed that the design caters for the pressures at normal operating conditions and intended relief conditions.

It is then the function of HAZOP to identify pressure deviations which may not have been foreseen.

The other type of limitation is that which is not intended, or desirable, but is simply inherent in the method. HAZOP is not, for example, particularly well suited to deal with spatial features associated with plant layout and their resultant effects."

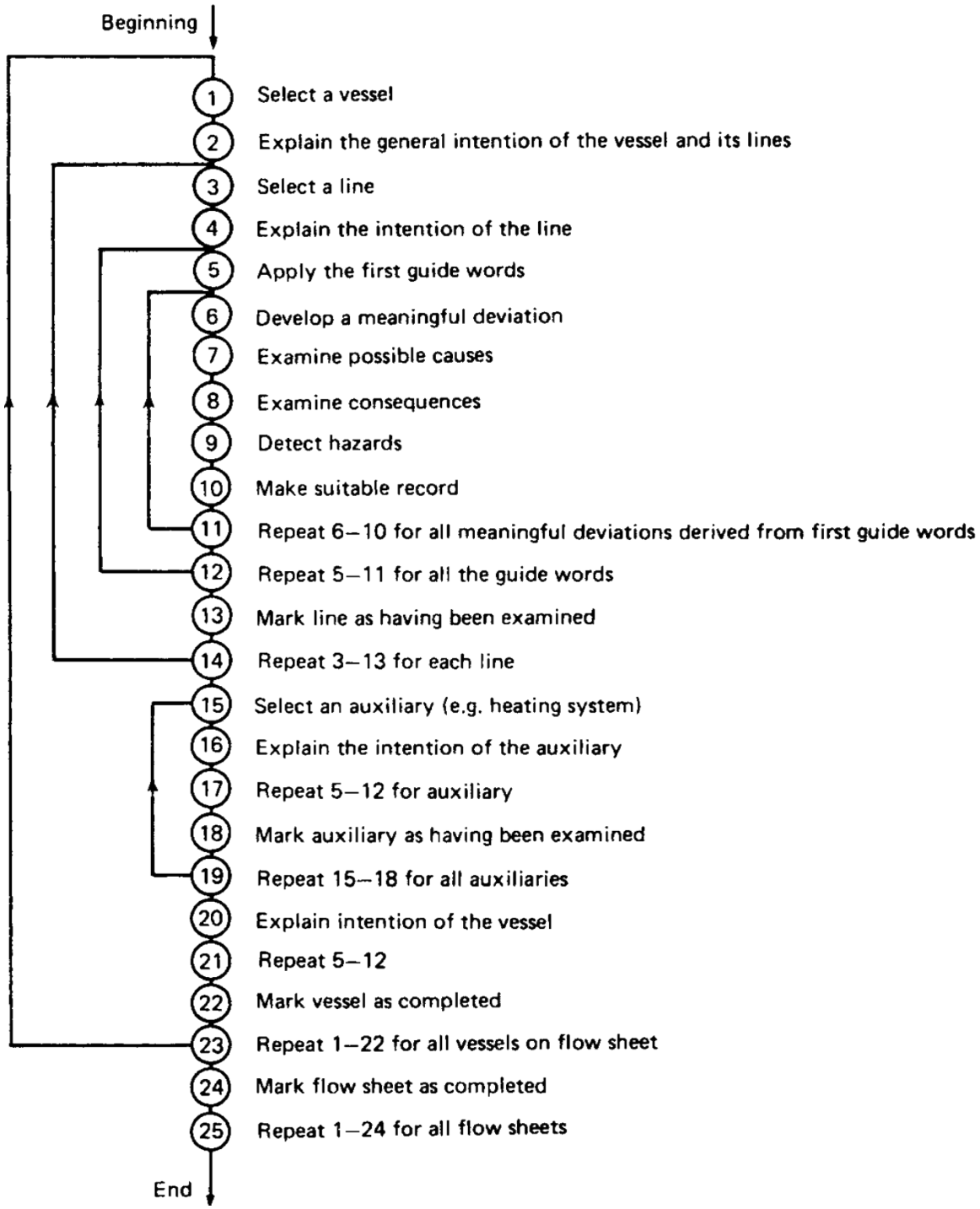


Figure 34: Hazard and operability studies: detailed sequence of examination
(Chemical Industry Safety and Health Council, 1977)

During the years, different evolutions of this type of analysis have been developed to adapt it to specific situations (batch system), human factor analysis, and so on. They will be discussed in the following paragraph .

Guide word	*Deviation*	*Possible causes*	*Consequences*	*Action required*
NONE	No flow	(1) No hydrocarbon available at intermediate storage	Loss of feed to reaction section and reduced output. Polymer formed in heat exchanger under no flow conditions	(a) Ensure good communications with intermediate storage operator. (b) Install low level alarm on settling tank LIC
		(2) J1 pump fails (motor fault, loss of drive, impeller corroded away, etc.)	As for (1)	Covered by (b)
		(3) Line blockage, isolation closed in error, or LCV fails shut	As for (1) J1 pump overheats	Covered by (b) (c) Install kick-back on J1 pumps (d) Check design of J1 pump strainers
		(4) Line fracture	As for (1) Hydrocarbon discharged into area adjacent to public highway	Covered by (b) (e) Institute regular patrolling and inspection of transfer line
MORE OF	More flow	(5) LCV fails open or LCV bypass open in error	Settling tank overfills	(f) Install high level alarm on LIC and check sizing relief opposite liquid overfilling (g) Institute locking off procedure for LCV bypass when not in use
			Incomplete separation of water phase in tank, leading to problems on reaction section	(h) Extend J2 pump suction line to 12 in above tank base
	More pressure	(6) Isolation valve closed in error or LCV closes, with J1 pump running	Transfer line subjected to full pump delivery or surge pressure	(j) Covered by (c) except when kick-back blocked or isolated. Check line, FQ and flange ratings, and reduce stroking speed of LCV if necessary. Install a PG upstream of LCV and an independent PG on settling tank
		(7) Thermal expansion in an isolated valved section due to fire or strong sunlight	Line fracture or flange leak	(k) Install thermal expansion relief on valved section (relief discharge route to be decided later in study)
	More temperature	(8) High intermediate storage temperature	Higher pressure in transfer line and settling tank	(l) Check whether there is adequate warning of high temperature at intermediate storage. If not, install

Figure 35: Hazard and operability studies: results for feed section of proposed alkene dimerization plant from intermediate storage to buffer/settling tank (Lawley, 1974)

Guideword	*Deviation*	*Possible causes*	*Consequences*	*Action required*
LESS OF	Less flow	(9) Leaking flange or valved stub not blanked and leaking	Material loss adjacent to public highway	Covered by (e) and the checks in (j)
	Less temperature	(10) Winter conditions	Water sump and drain line freeze up	(m) Lag water sump down to drain valve, and steam trace drain valve and drain line downstream
PART OF	High water concentration in stream	(11) High water level in intermediate storage tank	Water sump fills up more quickly. Increased chance of water phase passing to reaction section	(n) Arrange for frequent draining off of water from intermediate storage tank. Install high interface level alarm on sump
	High concentration of lower alkanes or alkenes in stream	(12) Disturbance on distillation columns upstream of intermediate storage	Higher system pressure	(o) Check that design of settling tank and associated pipework, including relief valve sizing, will cope with sudden ingress of more volatile hydrocarbons
MORE THAN	Organic acids present	(13) As for (12)	Increased rate of corrosion of tank base, sump and drain line	(p) Check suitability of materials of construction
OTHER	Maintenance	(14) Equipment failure, flange leak, etc.	Line cannot be completely drained or purged	(q) Install low point drain and N_2 purge point downstream of LCV. Also N_2 vent on settling tank

Figure 35: (continued)

3.1.4.1 Recursive operability analysis

In last years, different types of HAZOP were developed to solve different situations and to simplify certain steps of analysis. One of the most interesting variations is that proposed by Piccinini and Ciarambino (1997): it was developed to facilitate the subsequent development of the logic trees. The recursive operability analysis (ROA) differs from the traditional one for:

1) The way to define the relevant deviations, not through the use of guide words, but through the expect judgment;
2) The two recursive mechanism, that allows the consequences developed till the TOP EVENTs and the causes till the primary causes, not just in one row, but following their evolution in time and along the process;
3) The way of recording the outcomings of the analysis, that allows to build almost automatically the fault tree (Piccinini and Ciarambino, 1997).

This type of analysis proceeds in the way presented in Figure 37.

The analysis begins with the preliminary activity, that consists of the subdivision of the system into subsystems (A, B, C, W). The subdivision consists of dividing the system into homogeneous parts. Some considerations about the subdivision can be founded in Figure 36.

1. A plant should be analyzed in accordance with its 'flow lines' and divided in accordance with its process functions. These are distributed to the components, or sets of components that will actually form the sub-systems. Exceptions, however, may be necessary, for example when the main purpose of an analysis is to study the TE that will arise through the occurrence of deviations that do not propagate along the usual flow lines.
2. Each sub-system must comprise elements operation on the prevalent process function.
3. A given component can only form part of one subsystem. This rule must always be applied to ensure that fully distinct sub-systems are obtained.
4. Apart from the protection systems, all the instrumentation relating to an apparatus must be included in its specific sub-system.
5. It is a good idea to regard the automatic shut-down devices as distinct sub-systems.

Figure 36: Recommended ways of dividing a plant into subsystems

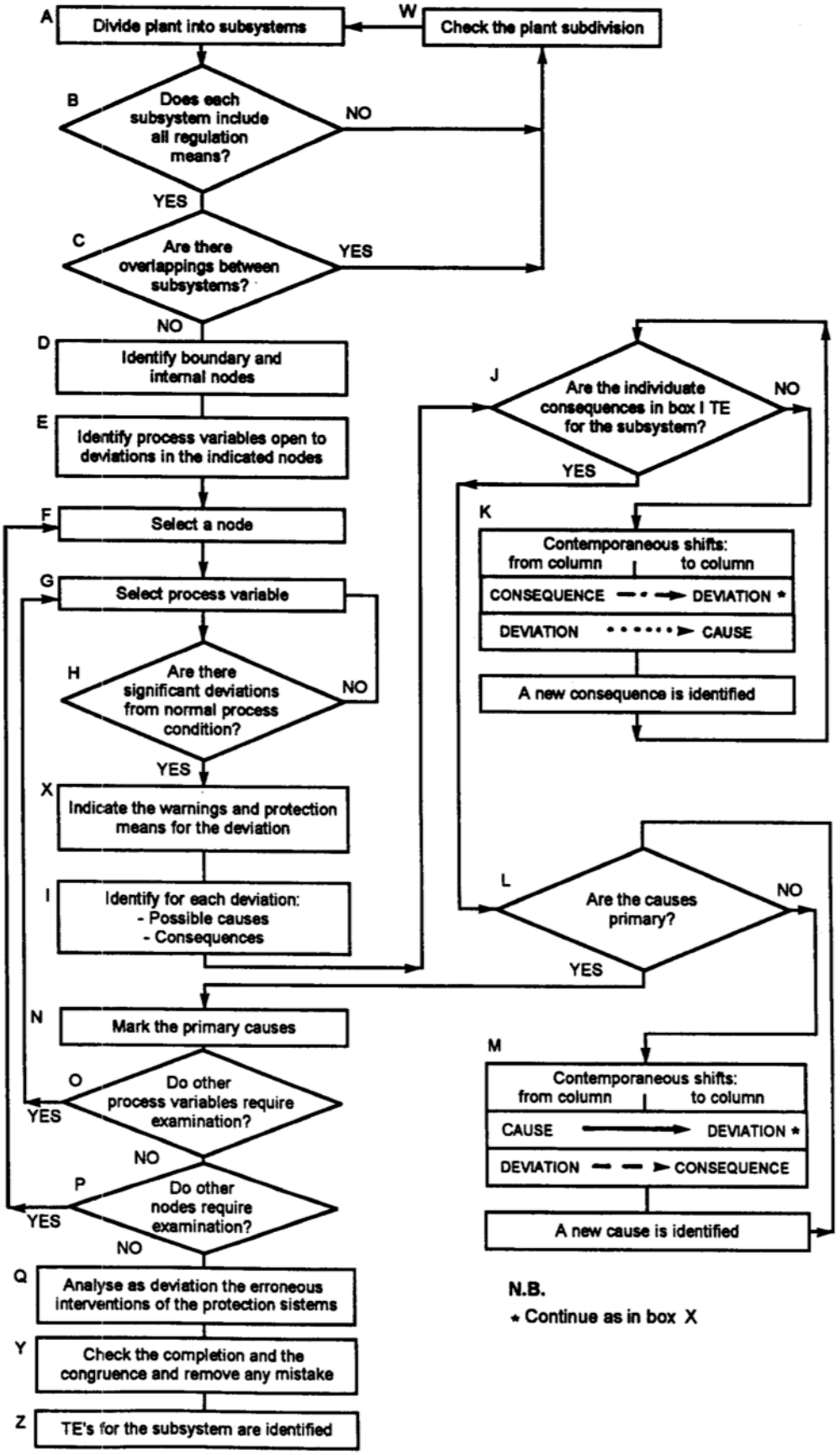


Figure 37: Flow sheet of recursive operability analysis

The next stage is the identification of the nodes (D), defined by (Piccinini and Ciarambino, 1997) as:

"Another preliminary operation is the identification of the boundary nodes and those internal to each ub-system--|Box D|, i.e. the points where deviations of a process variable (temperature, pressure, etc.) may develop or propagate."

Nodes are points in which are assumed the deviations from the design conditions and from which is observed the propagation of the deviations in the system.

Nodes are divided into two types, the boundary nodes and the internal nodes:

- Boundary nodes are the points of contact between more subsystems along process lines;
- Internal nodes, instead, are points placed where the analyst want to observe with special attention how the system behaves following deviations from design conditions.

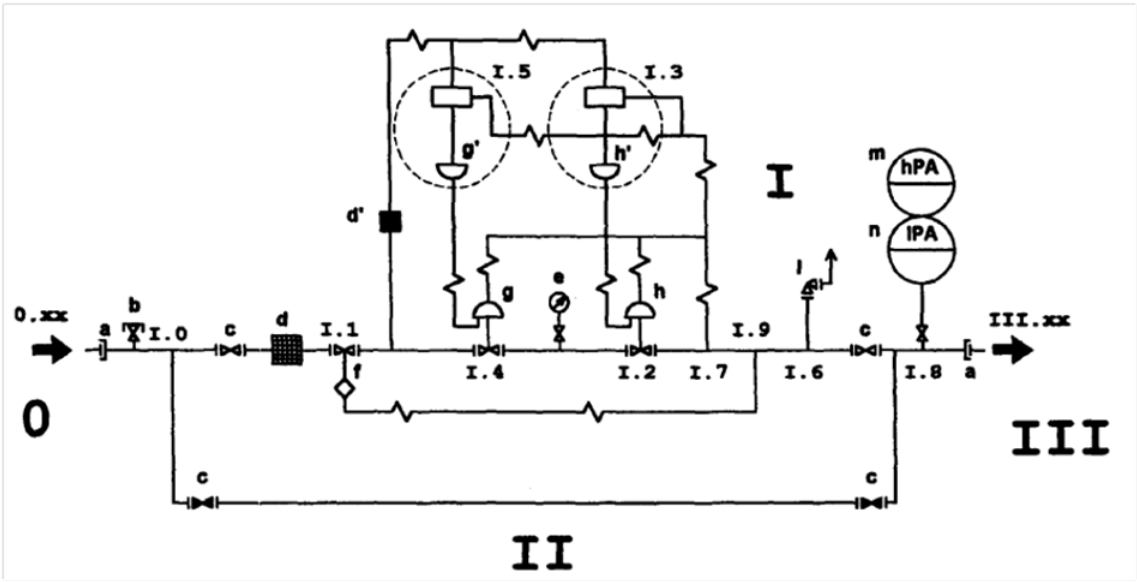
The various nodes are identified and a numbered consecutively.

For each node, the process variables that may be subject to deviations have to be identified (E). Then the analysis can be started by completing a table like the one shown in Figure 38. The table columns represent:

1. The deviation hypothesized;
2. The possible causes;
3. The possible consequences;
4. Optical or acoustic alarms that intervene as a result of the deviation;
5. The automatic protection systems;
6. Notes;
7. The Indication if the traced consequence is a Top Event.

A consequence can be defined as Top Event if it is a condition that has some relevance for the system or it is the maximum possible consequence of the subsystem.

The analysis proceeds by examining the nodes previously identified (F), one at a time. To the node analyze, a deviation from the design parameters (H) is applied for each variable (G).



Boundary notes: I.0; III.0 xx Node not defined along a line
Internal nodes: I.1; I.2; I.3; I.4; I.5; I.6; I.7; I.8; I.9;
Nodes already examined:-
* Primary event. One not open to further analysis and hence a terminal point

Deviation	Possible causes	Consequences	Warning (optical or acustical)	Protection means (automatic)	Note	TE
1.0 - High P from medium pressure line	0.xx - High P from medium pressure line	Not examined				
1.9 - P + ΔP	1.2 - Regulator fails in open position	1.9 - High P		1.9 - Monitor		
1.9 - High P	1.9 - P + ΔP	1.9 - High P + ΔP	1.8 - High P alarm			
	1.2/1.4 - Regulator & monitor opening					
1.9 - High P + ΔP	1.9 - High P	1.9 - Very high P		1.6 - Vent value		
		1.6 - Vent valve stuck open				
1.6 - Vent valve stuck open	1.9 - High P + ΔP & 1.6 - Intervention of vent valve *	1.6 - Discharge of gas to the atmosphere			¥	3
1.9 - Very high P	1.9 - High P + ΔP	III.xx - Very high P to users		I.1 - Slam shut value		1
1.2 - Regulator fails in open position	1.2 - Malfunctioning (opening) of regulator *	1.6 - Discharge of gas to the atmosphere				
	1.3 - Malfunctioning (opening) of regulator pilot *					
1.2/1.4 - Regulator & monitor opening	1.7 - Breakage of pilots low-pressure pulse inlet *	1.9 - High P				

Figure 38: Example of Recursive operability analysis

For any deviation identified, the protection systems or alarms (X), the causes and the consequences (I) have to be identified. In case of multiple causes or consequences, the logical link that correlates them has to be identified: an example is shown in Figure 39.

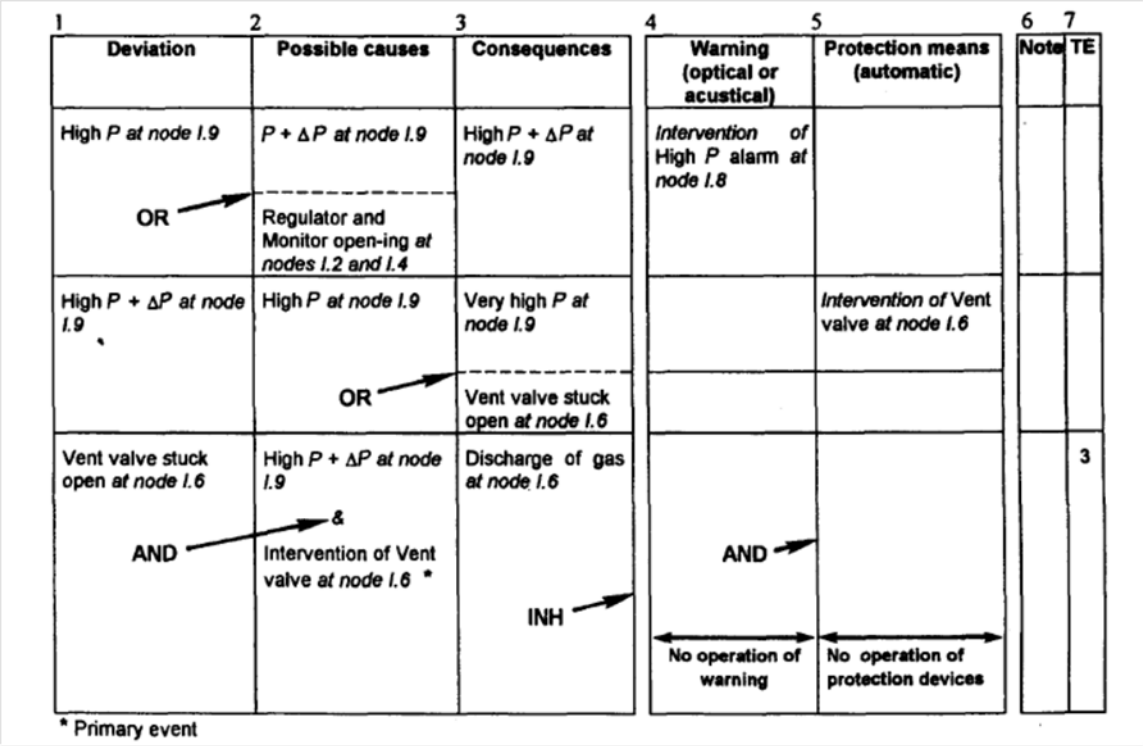


Figure 39: Graphical sign used for represents logical bonds

Once the consequences have been defined, they have to be classified as TOP EVENT or not (J). In case of an affirmative answer, a sequential number in the last column has to be inserted. Otherwise, it is necessary to report in the following line the contents of the column *Consequences* in the *Deviation* one and the contents of the column *Deviation* in the *Possible causes* one (K). At this point, possible warning or automatic protection system are considered as fault.

Once a TOP EVENT has been identified, the analysis of the deviation proceeds whit the analysis of the causes, to identified the primary causes (L). One cause is called primary if it cannot be divided further, maintaining the level of detail desired. In example the failure of a control system can be divided in the failures of its components: the sensor, the logic and the actuator; generally this level of detail may be sufficient.

If the cause is not a primary cause, it must be further analyzed shifting it in the *Deviation* column and the *Deviation* in the column of the *Consequences* (M). Now IT is necessary to identify in depth the possible causes, iterating the process until all the causes are brought to the level of primary causes. The primary causes are indicated by a distinctive sign, typically an asterisk.

The method described above has to be followed for all the hypothesized deviations (O), for all the variables on all the nodes (P).
The final check on the completeness of the analysis concludes the ROA.

3.2 Estimation of the probability of occurrence

This section describes the techniques used to estimate the probability of occurrence of unwanted events, as identified in previous step.
These techniques are the Event Tree (ET) and the Fault Tree (FT).
The ET is a graphical structure that represents the possible consequences that may occur from a single initial event.
The FT is a graphical representation of all the possible causes for a given event and the logical links between these causes.

3.2.1 Event tree

This technique can be both qualitative and quantitative. The qualitative use is as graphical representation that put in evidence the possible consequences from an initial event. The quantitative part allows the evaluation of the probability (or frequency) of the various possible consequences.
The ET consists of a graphical structure like the one in Figure 40, with the starting event on the left,

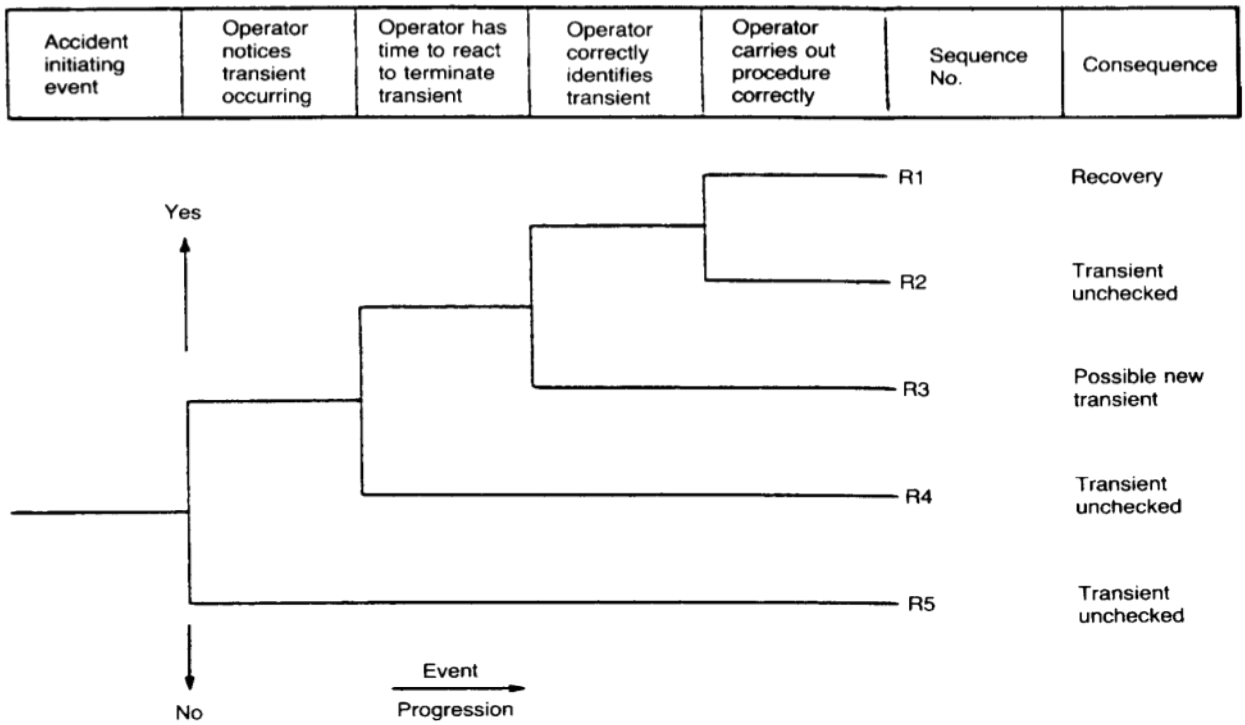


Figure 40: Example of qualitative ET (Mannan, 2005)

from which the different branches branch off. The upper branch represent the occurrence of the event described at the top of the tree, the lower branch represent the non-occurrence.

Figure 41 is an example of an ET used for quantitative purposes: the initiator event is associated to its probability or frequency (events / year), as well as each other event represented. At this point you can get the probabilities of the various possible consequences. The probability of the

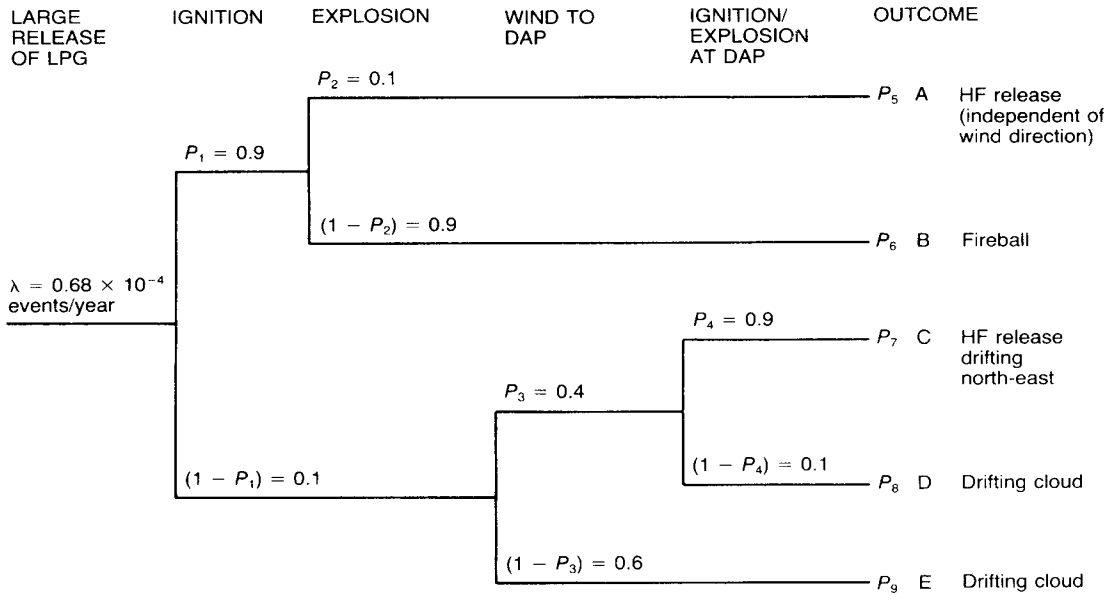


Figure 41: Example of quantitative ET

consequences is obtained starting from the probability of the event initiator and multiplying it with the probability of each event met along the branches.

In the literature many examples of the use of ET are available, such as: (Von Alven, 1964; Rasmussen, 1974; HEALTH AND SAFETY EXECUTIVE, 1978; HEALTH AND SAFETY EXECUTIVE, 1981; Rijnmond Public Authority, 1982).

3.2.2 Fault tree

The fault trees are graphical and represent a logical structure that shows all the possible causes for a given event, in a top-down diagram. Some literature examples are present in: (Henley and Kumamoto, 1992; Dhillon and Singh, 1981; Vesey, 1981; Barlow, Fussell and Singpurwalla, 1975).

The fault tree methodology can be used both in a qualitative way, describing all the ways that may lead to a certain event, and in a quantitative way, estimating the probability of occurrence of the Top Event. In complex not standardized systems, such as process plants, the fault trees are developed starting from an HAZOP analysis, which identifies what are the critical events and the logical links throughout the causes.

"A general account of fault tree methods has been given by (Fussell, 1976). He sees fault tree analysis as being of major value in

(1) directing the analyst to ferret out failures deductively;

(2) pointing out the aspects of the system important in respect of the failure of interest;

(3) providing a graphical aid giving visibility to those in system management who are removed from system design changes;

(4) providing options for qualitative or quantitative system reliability analysis;

(5) allowing the analyst to concentrate on one particular system failure at a time;

(6) providing the analyst with genuine insight into system behavior."

In this type of analysis, a device can only assume two states: the state of function or the failure. This seems to be a simplifying assumption, because actually the operation status of an equipment can have many intermediate conditions. The use of this approximation, however, allows to hypothesize a binary behavior and to apply Boolean logic.

Among the items listed in Figure 42 are relevant:

1. The Top Event, already introduced in the previous section: in this case it is the event (usually unwanted) that is on top of the FT.
2. Primary events: the primary events are the basic events that are not investigated to search more detailed causes.
3. Intermediate events, are events located between the primary events and the Top Event.

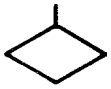
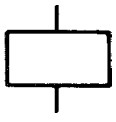
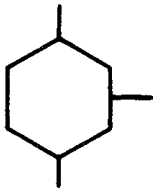
A Events	
Symbol	
	Primary, or base, event – basic fault event requiring no further development
	Undeveloped, or diamond, event – fault event which has not been further developed
	Intermediate event – fault event which occurs due to antecedent causes acting through a logic gate
	Conditioning event – specific condition which applies to a logic gate (used mainly with PRIORITY, AND and INHIBIT gates)
	External, or house, event – event which is normally expected to occur[a]

Figure 42: Fault tree event and logic symbols (Mannan, 2005)

Symbol	*Alternative symbol*	
		AND gate – output exists only if all inputs exist
		OR gate – output exists if one or more inputs exists
		INHIBIT gate – output exists if input occurs in presence of the specific enabling condition (specified by conditioning event to right of gate)
		PRIORITY AND gate – output exists if all inputs occur in a specific sequence (specified by conditioning event to right of gate)
		EXCLUSIVE OR gate – output exists if one, and only one, input exists
		VOTING gate – output exists if there exist *r*-out-of-n inputs[b]
		TRANSFER IN – symbol indicating that the tree is developed further at the corresponding TRANSFER OUT symbol
		TRANSFER OUT – symbol indicating that the portion of the tree below the symbol is to be attached to the main tree at the corresponding TRANSER IN symbol

[a] This is the definition given by Vesely *et al.* (1981). Other authors such as Henley and Kumamoto (1981) use this symbol for an event which is expected to occur or not to occur.
[b] See Chapter 13.

Figure 42 : (continued)

4. Logic gates are the gates that indicate the relationship between the input and output data
   - OR: Indicates that the output event occurs if even only one of the input events actualizes;
   - AND: is the condition in which the output event occurs only if all the input events actualize;

- INHIBIT: is the condition in which the output event occurs if the input event occurs under certain conditions. This type of gate is used to represent alarm system, or protection devices: in this case the output event occurs if the input event actualizes, and, only on demand, the protection systems does not intervene.

"The construction of a fault tree appears a relatively simple exercise, but it is not always as straightforward as it seems and there are a number of pitfalls. Guidance on good practice in fault tree construction is given in the Fault Tree Handbook. Other accounts include that in the CCPS QRA Guidelines (CENTER FOR CHEMICAL PROCESS SAFETY, 1989), and those by (Lawley, 1974; Lawley, 1980; Fussell, 1976; Doelp et al., 1984).

An essential preliminary to construction of the fault tree is definition and understanding of the system. Both the system itself and its bounds need to be clearly defined. Information on the system is generally available in the form of functional diagrams such as piping and instrument diagrams and more detailed instrumentation and electrical diagrams. There will also be other information required on the equipment and its operation, and on the environment. The quality of the final tree depends crucially on a good understanding of the system, and time spent on this stage is well repaid.

It is emphasized by (Fussell, 1976)that the system boundary conditions should not be confused with the physical bounds of the system. The system boundary conditions define the situation for which the fault tree is to be constructed. An important system boundary condition is the top event. The initial system configuration constitutes additional boundary conditions. This configuration should represent the system in the unfailed state. Where a component has more than one operational state, an initial condition needs to be specified for that component. Furthermore, there may be fault events declared to exist and other fault events not to be considered, these being termed by Fussell the 'existing system boundary conditions' and the 'not allowed system boundary conditions', respectively. Fault trees for process plants fall into two main groups, distinguished by the top event considered. The first group comprises those trees where the top event is a fault within the plant, including faults that can result in a release or an internal explosion. In the second group, the top event is a hazardous event outside the plant, essentially fires and explosions.

If the top event of the fault tree is an equipment failure, it is necessary to decide whether it is the reliability, availability, or both, which is of interest. Closely related to this is the extent to which the components in the system are to be treated as non-repairable or repairable.

As already described, the principal elements in fault trees are the top event, primary events and intermediate events, and the AND and OR gates. The Handbook gives five basic rules for fault tree construction:

Ground Rule 1	Write the statements that are entered in the event boxes as faults; state precisely what the fault is and when it occurs.
Ground Rule 2	If the answer to the question, 'Can this fault consist of a component failure?' is 'Yes', classify the event as a 'state-of component fault'. If the answer is 'No', classify the event as a 'state of-system fault'.
No Miracles Rule	If the normal functioning of a component propagates a fault sequence, then it is assumed that the component functions normally.
Complete-the-Gate Rule	All inputs to a particular gate should be completely defined before further analysis of any one of them is undertaken.
No Gate-to-Gate Rule	Gate inputs should be properly defined fault events, and gates should not be directly connected to other gates.

Each event in the tree, whether a top, intermediate or primary event, should be carefully defined. Failure to observe a proper discipline in the definition of events can lead to confusion and an incorrect tree.

The identifiers assigned to events are also important. If a single event is given two identifiers, the fault tree itself may be correct, if slightly confusing, but in the minimum cut sets the event will appear as two separate events, which is incorrect.

For a process system, the top event will normally be a failure mode of an equipment. The immediate causes will be the failure mechanisms for that particular failure. These in turn constitute the failure modes of the contributing subsystems, and so on.

The procedure followed in constructing the fault tree needs to ensure that the tree is consistent. Two types of consistency may be distinguished: series consistency within one branch and parallel consistency between two or more branches. Account needs also to be taken of events that are certain to occur and those that are impossible.

The development of a fault tree is a creative process. It involves identification of failure effects, modes and mechanisms. Although it is often regarded primarily as a means of quantifying hazardous events, which it is, the fault tree is of equal importance as a means of hazard identification. It follows also that fault trees created by different analysts will tend to differ. The differences may be due to style, judgment and/or omissions and errors.

It is generally desirable that a fault tree has a well-defined structure. In many cases such a structure arises naturally. It is common to create a 'demand tree', which shows the propagation of the faults in the absence of protective systems, and then to add branches, representing protection by instrumentation and by the process operator, which are connected by AND gates at points in the demand tree." (Mannan, 2005).

3.2.2.1 Quantification the FT

The first analysis on a FT is a logical one, to identify the minimal cut set (MCS). The cut sets are sets of events that together are sufficient to bring to the Top Event. A MCS is a cut set that does not contain other cut set.

According to the example in Figure 43, rewriting the FT in Boolean form brings to the following expression:

$$T = (A + B + C + D)(B^* + F)(G + H + I) \qquad [\ 3.2.2.1]$$

If needed, the equation can be reduced, using the rules of Boolean algebra (Figure 44).

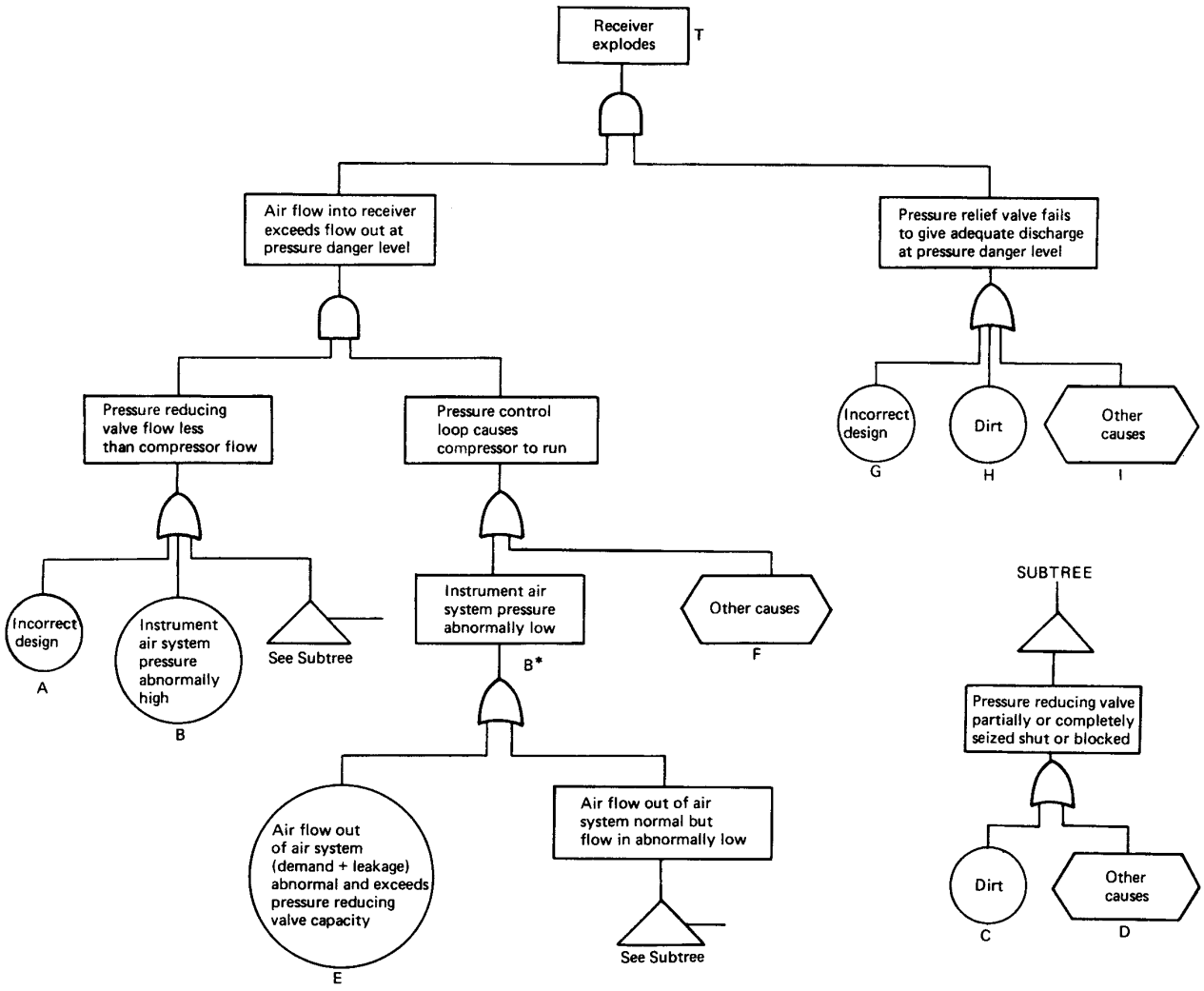


Figure 43: Instrument air receiver system: flow diagram and fault trees for the explosion of an air receiver: (a) instrument air receiver system; (b) fault tree for top event 'Receiver explodes'; (c) equivalent but simplified fault tree for top event 'Receiver explodes'

Definitions:	
Null set $\emptyset$	$\emptyset$ is a set with no elements
Sample space set S	S is the set containing all the elements in the sample space
Elements	
$a \in A$	a is an element of A
Subsets	
$A \subset B$	A is a subset of B
$B \supset A$	B contains A
$A \subseteq B$	A is a subset of B or is equal to B
Equality	
$A = B$	A has the same elements as B
$A = \emptyset$	A has no elements
Operations:	
Union of sets	
$C = A \cup B$	C contains all the elements of A and B
also written	
$C = A + B$	
Intersection of sets	
$C = A \cap B$	C contains only the elements common to A and B
also written	
$C = AB$ or $A \cdot B$	
$C = A$ and B	
Disjoint sets (mutually exclusive sets)	
$C = A \cap B = \emptyset$	
Difference of sets	
$C = A - B$	C contains all the elements of A which are not elements of B
Complement of a set[a]	
$A' = S - A$	A' contains all the elements of S which are not elements of A
Laws:	
Commutative laws	
$A + B = B + A$	
$AB = BA$	
Associative laws	
$(A + B) + C = A + (B + C) = A + B + C$	
$(AB)C = A(BC) = ABC$	
Distributive laws	
$A(B + C) = AB + AC$	
$A + BC = (A + B)(A + C)$	
Absorption laws	
$A + A = A$	
$AA = A$	
Dualization (de Morgan's) laws	
$(A + B)' = A'B'$	
$(AB)' = A' + B'$	

[a] Use may also be made of the notation $\bar{A}$ to signify 'not A'.

Figure 44: Some Boolean algebra law

Where:

$$B^* = C + D + E \quad [3.2.2.2]$$

From Boolean's algebra:

$$BB' = 0 \quad [3.2.2.3]$$

$$CC = C \; and \; DD = D \quad [3.2.2.4]$$

$$AC, CD, CE, CF \subset C \quad [3.2.2.5]$$

$$AD, DC, DE, DF \subset D \qquad [\ 3.2.2.6]$$

Thus the expression becomes:

$$T = (AE + AF + C + D) \cdot (G + H + I) = [A \cdot (A + F) + BF + C + D] \cdot (G + H + I) \qquad [\ 3.2.2.7]$$

Solving the above reported equation, the following minimum cut set are obtained:

AEG, AEH, AEI, AFG, AFH, AFI, BEG, BFH, BFI, CG, CH, CI, DG, DH, DI. [3.2.2.8]

"Quantitative evaluation of a fault tree requires that numbers be put to the frequency or probability of the primary events. Given these quantitative data, there are several options for the evaluation of the frequency or probability of the top event.

Three methods of evaluation are considered here. These are the use of (1) the minimum cut sets, (2) the gate-by-gate method and (3) Monte Carlo simulation.

In the first of these methods, the probability of the top event may be evaluated from the probabilities of the minimum cut sets C_i

$$P(T) = P\left(\bigcup_{i=1}^{n} C_i\right) \qquad [\ 3.2.2.9]$$

The events are not mutually exclusive and, therefore, strictly

$$\begin{aligned} P(T) = {} & P(C_1) + P(C_2) + \cdots + P(C_n) - P(C_1)P(C_2) - \\ & -P(C_1)P(C_3) - \cdots - P(C_{n-1})P(C_n) + \\ & P(C_1)P(C_2)P(C_3) + P(C_1)P(C_2)P(C_4) + \cdots + P(C_{n-2})P(C_{n-1})P(C_n) \\ & + \cdots + (-1)^{n-1} \prod_{i=1}^{n} P(C_i) \end{aligned} \qquad [\ 3.2.2.10]$$

But usually it is sufficient to use the low probability, or rare event, approximation

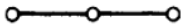
$$P(T) = \sum_{i=1}^{n} P(C_i) \qquad [\ 3.2.2.11]$$

Equation [3.2.2.11] always gives a higher probability than Equation [3.2.2.10] and thus for failure oriented logic it is conservative.

The second method is to work up the tree gate-by-gate from the bottom calculating the frequency or probability of the output event of each gate from those of the input events. The procedure is straightforward except where there occur at the gate, some input faults which are expressed in terms of frequency rather than probability.

The output-input relations which are permitted and those that are not are given for a two-input gate in Figure 45, Section B. The main problem arises where the two inputs to an AND gate both have

A Basic probability relations[a]

Logic symbol	*Reliability graph*	*Boolean algebra relation*	*Probability relations*
		A = BC	$P(A) = P(B)P(C)$
		A = B + C	$P(A) = P(B) + P(C) - P(B)P(C)$

B Relations involving frequencies and/or probabilities[a]

Gate	*Inputs*	*Outputs*
OR	P_B OR P_C	$P_A = P_B + P_C - P_B P_C \approx P_B + P_C$
	F_B OR F_C	$F_A = F_B + F_C$
	F_B OR P_C	Not permitted
AND	P_B AND P_C	$P_A = p_B p_C$
	F_B AND F_C	Not permitted; reformulate
	F_B AND P_C	$F_A = F_B P_C$

[a] F, frequency; P, probability.

Figure 45: Probability and frequency relations for fault tree logic gates (output A; inputs B and C)

the dimensions of frequency. The output from the gate must also have the dimension of frequency. It is not permissible to multiply the two frequencies together.

There are advantages and disadvantages to both the gate-by-gate and minimum cut set methods. The former is the method traditionally used in manual evaluation of trees. It gives an evaluation of all the intermediate events as well as of the top event, which can be valuable. On the other hand, it tends to be error prone and becomes tedious for large trees.

The third method is the use of Monte Carlo simulation. ... Its application to fault tree evaluation involves a series of trials. In a given trial each primary event either does or does not occur, the occurrence being determined by sampling, where the values returned by the sampling are, on average, in accordance with the frequency or probability data supplied. The outcome of each trial is the occurrence or non-occurrence of the top event. Provided a sufficient number of trials are used, the frequency or probability of the top event is then given by the proportion of trials in which it occurs. An account of the use of this method for fault tree evaluation has been given by (Hauptmanns and Yllera, 1983).

The use of Monte Carlo simulation is virtually unavoidable if the primary events are characterized by a range of values of probability, or frequency. The probability of a failure may be given not by a point probability value but by a probability density function. Obviously if the probabilities of the other events are expressed as density functions, then the probability of the top event must be expressed as a density function also.

A given Monte Carlo trial generates a set of probabilities for the primary failure events. The probability of the top event may then be evaluated by an analytical method, such as the minimum cut set method." (Mannan, 2005)

Over the years, however, several softwares were developed to allow a direct quantitative and qualitative assessment of the FT. One of this is ASTRA, developed by Joint Research Centre (Contini, 1995; Contini et al., 1998), that has been used in this work.

3.2.3 Construction of logic tree starting from a ROA

This section report the logic trees that can be extracted from a recursive analysis of operability (Piccinini and Ciarambino, 1997).

The first graphic representation that can be developed by a ROA is the Incidental Sequences Diagram (ISD). This graph is a simple transcription in graphic form of the data contained in the analysis.

The procedure to obtain the ISD from the recursive HAZOP is shown in Figure 46.

The development of IDS starts with the selection of the Top Event (A). In the same row of the event, the cause and the deviations (B) can be found: if the causes are more than one, they are clustered under the appropriate logic gate (D) OR or AND.

If the specific deviation (E) has a protective systems or alarm (F) INHIBIT gate must be introduced (G).

Concerning the causes: if the cause is not a primary one, it is necessary repeat the construction of the tree, looking at the line where the cause is reported in the column of the deviations (N) and repeating the procedure from step E, until there only primary causes remain. The whole operation is repeated for all causes (P) and all deviations (Q) that lead to the Top Event.

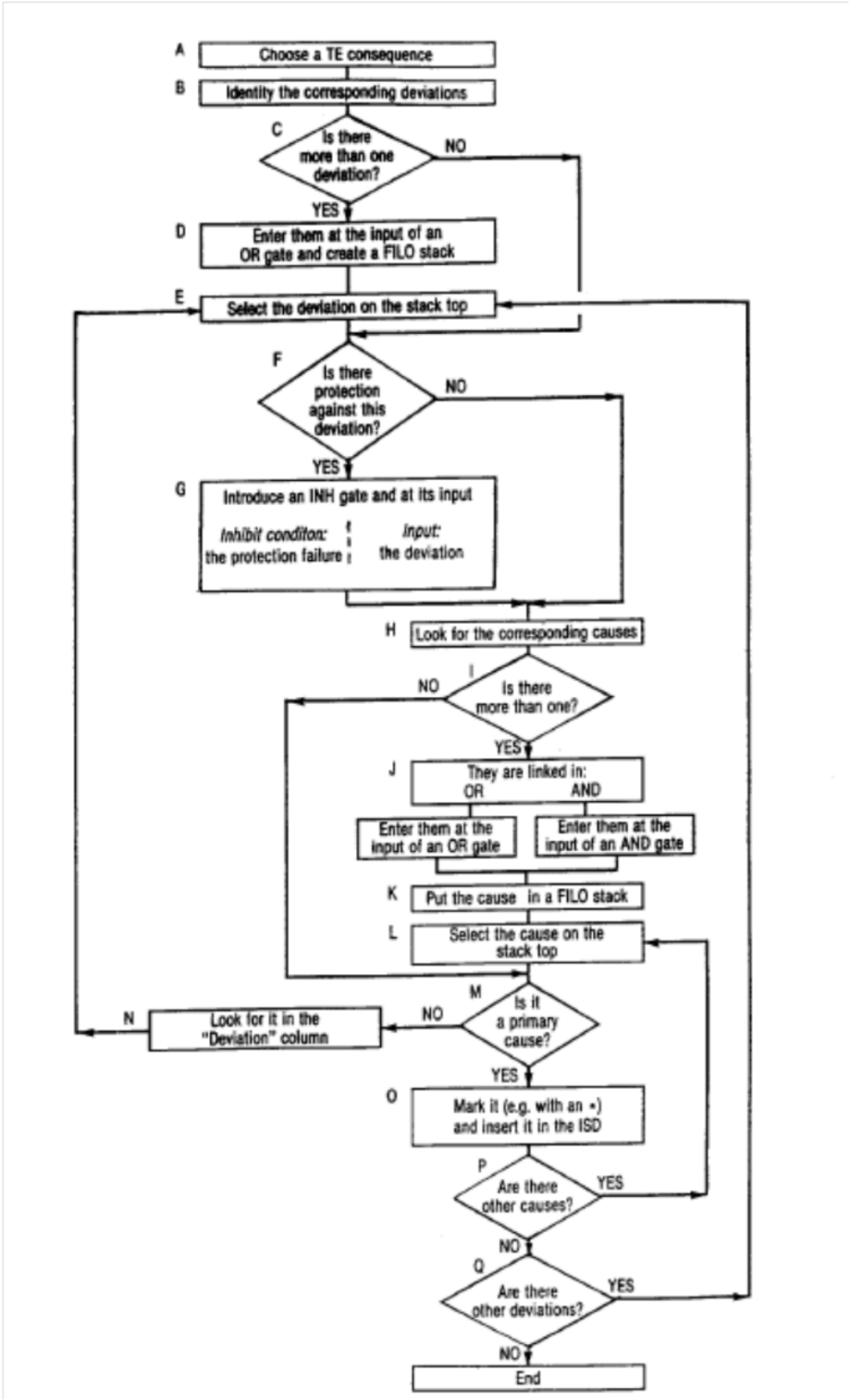


Figure 46: Construction of an Incidental Sequence Diagram from an Operability Analysis

As shown in Figure 47, ISD is very similar to a FT. Once the ISD has been developed, to obtain a FT it is sufficient to graphical develop the branches of the protection systems till their primary cause.

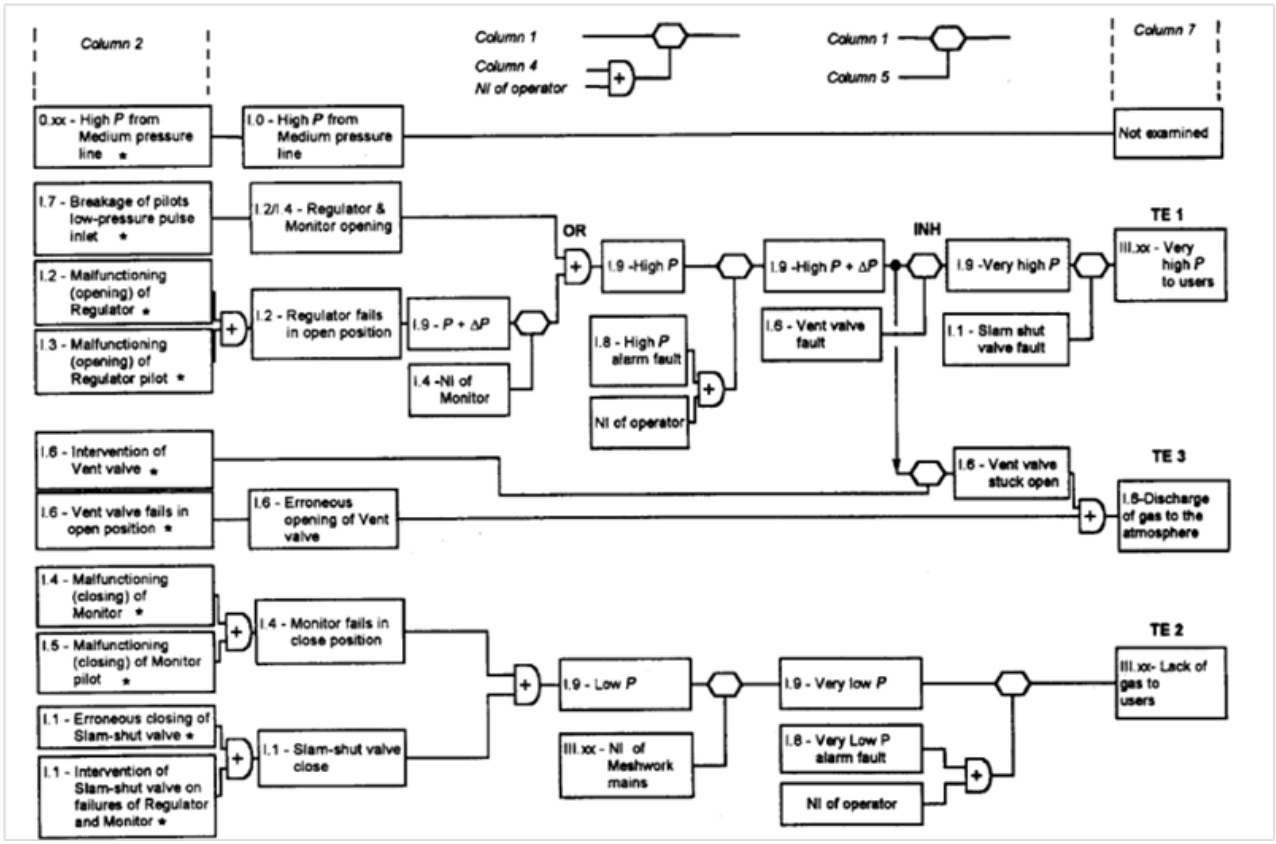


Figure 47: Example of Incidental Sequences Diagram. (HI: no intervention, *: primary event.) directly down from the HAZOP recursive table in Figure 38 (Piccinini and Ciarambino, 1997)

3.2.4 Bases of reliability

The reliability theory assumes that the components can have a binary behavior: operating or failed. In this case, reliability (*R(t)*) is defined as the probability that the component arrives at the observation time (t) still working, without having faults.

$$R(t) = \Pr_{failure}\{\tau > t\} \qquad [\ 3.2.4.1]$$

In this case we can derive the following properties for this function:

- That the component at time 0 is certainly operating $R(0) = 1$
- And after a very long time the component will certainly fall $\lim_{t\to\infty} R(t) = 0$.

Similarly we can define the unreliability (*F(t)*) such as the probability that at the observation time the component is already failed. In this way $R(t) + F(t) = 1$

Intuitively, an estimator of unreliability can be:

$$F(t) = \frac{n_g(t)}{n} \qquad [\ 3.2.4.2]$$

Where: n_g is the number of components that have failed after a certain time (*t*), and *n* is set of identical components operating from the time 0.

The probability that a particular component fails at time [t, t+dt], the density of the probability of failure (*f (t)*) has to be introduced:

$$f(t) = \lim_{\Delta t \to 0} \frac{1}{n} \cdot \frac{n_g(t + \Delta t) - n_g(t)}{\Delta t} = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \qquad [\ 3.2.4.3]$$

At this point, the failure rate (*h(t)*) can be introduced as the probability that a component fails between t and t + dt, provided that it is arrived at time t in a operating mode.

$$h(t) = \frac{f(t)}{R(t)} = -\frac{1}{R(t)} \cdot \frac{dR(t)}{dt} = -\frac{d[lnR(t)]}{dt} \qquad [\ 3.2.4.4]$$

Whereby:

$$R(t) = exp\left(-\int_0^t h(t)\, dt\right) \qquad [\ 3.2.4.5]$$

$$F(t) = 1 - exp\left(-\int_0^t h(t)\, dt\right) \qquad [\ 3.2.4.6]$$

$$f(t) = h(t) \cdot R(t) = h(t) \cdot exp\left(-\int_0^t h(t)\, dt\right) \qquad [\ 3.2.4.7]$$

"In general, the failure behavior of an equipment exhibits three stages: initially during commissioning the rate is high, then it declines during normal operation, and finally it rises again as deterioration sets in. For many equipments, particularly electronic equipments, the rate has been found to form a bathtub curve, as shown in Figure 48 (a) (e.g., (Carhart, 1953)). This curve has three regimes: (1) early failure, (2) constant failure and (3) wear-out failure.

Early failure, or infant mortality, is usually due to such factors as defective equipment, incorrect installation, etc. It also tends to reflect the learning curve of the equipment user. Constant failure, or so-called' random failure', is often caused by random fluctuations of load which exceed the design strength of the equipment. A constant failure characteristic is also shown by an equipment which has a number of components that individually exhibit different failure distributions. Wear-out failure is self-explanatory. The corresponding curve for the failure density function is shown in Figure 48(b).The bathtub curve is widely quoted in the reliability literature, but it should be emphasized that its applicability to all types of equipment, particularly mechanical equipment, is

not established. There are, in fact, good theoretical reasons for treating it with reserve (Carter, 1973)This aspect is considered in more detail below.

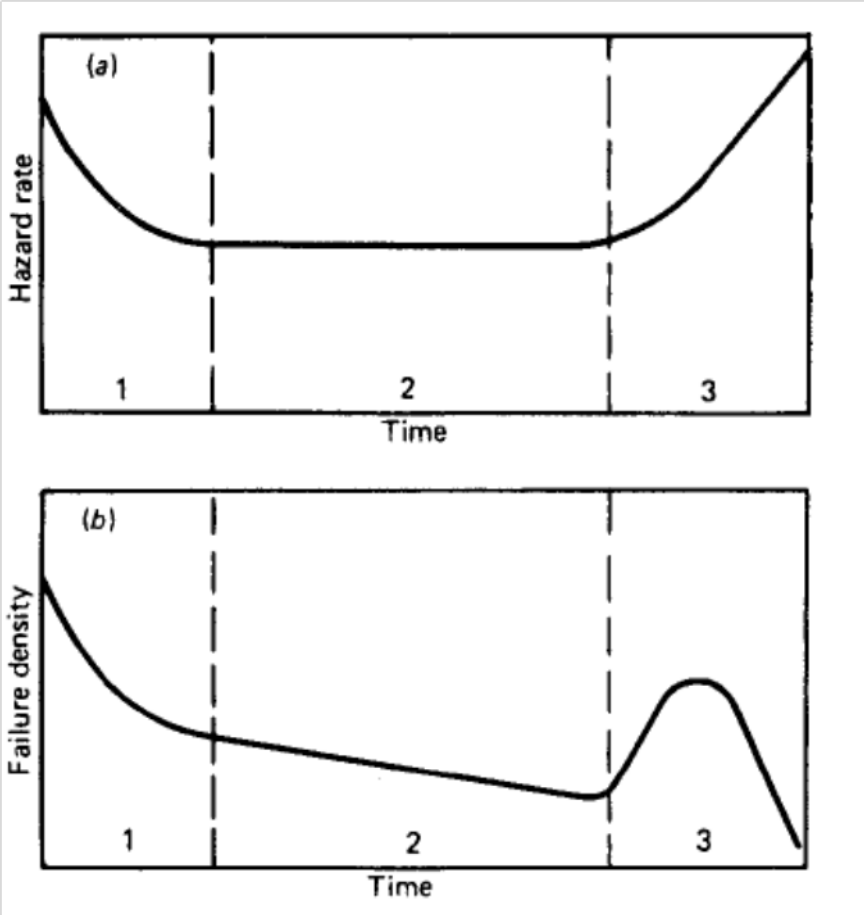


Figure 48: The bathtub curve: (a) hazard rate; (b) failure density

It is often said that human mortality follows a bathtub curve. This implies that there are higher death rates in infancy and old age with a lower, nearly constant, death rate in between. However, (Aird, 1978) has shown that for humans the infant mortality period is followed immediately by the onset of the wear-out period, so that there is effectively no constant failure period." (Mannan, 2005).

In general, it is assumed that the component is in condition of random fault, so that the failure rate can be considered constant and equal to λ.

$$R(t) = exp(-\lambda t) \qquad [\ 3.2.4.8]$$

$$F(t) = 1 - exp(-\lambda t) \qquad [\ 3.2.4.9]$$

$$f(t) = \lambda \cdot exp(-\lambda t) \qquad [\ 3.2.4.10]$$

In case of equipments that can be repaired, the availability (*A(t)*) is defined as the probability that at the observation time the component is working. This is because the behavior of this type of component is characterized by alternating periods in which it works and periods in which it does not work.

$$A(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} exp[-(\lambda + \mu)t] \qquad [\ 3.2.4.11]$$

After a long time we can assume that the system arrives to a steady state, so the probability of finding the component working assumes a constant value equal to:

$$A(\infty) = \frac{\mu}{\lambda + \mu} \qquad [\ 3.2.4.12]$$

In this formulation appears μ that represents the repair rate, that is the probability that the component is repaired between t and t+dt, if the component at t is failed. Also in this case it can be assumed as constant.

The assessment of this value is complex and depends on many factors such as the availability of spare parts, the size of the maintenance teams, etc.. This value changes from company to company.

3.3 Estimation of the consequences

The next step in the risk assessment requires an estimation of the consequences for the cases analyzed with the logic trees.

The method used to estimate the consequences depends on the type of accident that is analyzed and on the type of expected consequence. The consequences can be economic (e.g. lost revenue and damage) or injury to persons (e.g. number of casualties or injuries) or to environment.

In case of economic damage, it is necessary to estimate the extent of the damages caused by the accident, such as damages to the equipment, costs related to the impact on the production and any other structural damage.

The consequences on people are divided on the basis of the type of accident hypothesized. If a toxic release is hypothesized, it is necessary to estimate the size of the area affected by the hazardous substance and the number of possible people involved.

On the other hand, if a release of flammable substances is hypothesized, it should be estimated the extension of the area involved by the thermal effects and the fumes, also considering the number of possible targets involved and the consequences on them.

The description of the model or the techniques used for the estimation of the consequences of an event are out of the scope of this work.

The only thing that has to be noticed is that in traditionally risk assessment the consequence analysis is carried on in a separated step from the probabilistic analysis.

3.4 Integrated Dynamic Decision Analysis

Integrated Dynamic Decision Analysis (IDDA) is described in: (Demichela and Piccinini, 2008; Demichela and Turja, 2011; Demichela and Camuncoli, 2013; Demichela, 2014).
This type of approach is based on a logical-probabilistic modeling of the system and on an implement a phenomenological modeling.
The logical-probabilistic model is based on a syntactic system based on general logic, according to the following steps:

1. Identification of the events related to the operation of the system itself and construction of a list of levels, with questions and affirmations, which represents the elementary matter of the logical model and also the nodes in the event tree.
2. Construction of a 'reticulum' indicating the addresses (subsequent level) to be visited after each response in each level, and a comment string that allows the user to read the logical development of a sequence.
3. Association to each of the levels of a probability, which represents the expectation degree of the failure or unwanted event and of an uncertainty ratio, which represents the distribution of the probability.
4. Definition of all the constraints, logical and probabilistic, which can modify run time the model, fitting it to the current knowledge status.

The logical model is based on a tree built on levels. Each level is used to describe a specific event, in example the state of operation of a component. The event described may represent different scenarios, i.e.: the component works or it does not work and returns a value lower than the real one, it does not work and returns a value higher than the real one. These descriptions are represented by the outputs of the level. In this way the levels are described by "questions", that are then associated to different answers. To each answer a probability value has to be associated.
The various levels should be placed in a network, indicating what event is expected after a specific answer and if the latter can modify another event, conditioning it.
Following this type of description of the system, it is possible to obtain all the possible sequences of events that describe the behavior of the system: the sequences describe the possible configurations of the system with the respective probabilities.

Together with the logical modeling, it is necessary to prepare a phenomenological model, able to describe the physical behavior of the system.

The phenomenological model of the process must be configured according to the system conditions described by the logic analysis, that means in example to import from logical modeling what equipment are expected to be failed and in which way. In this way, the phenomenological model will be able to simulate the behavior of the system and, according to the results obtained, it will be possible to change the logical model adapting it to the reality; i.e. indicating if, after the failure of a particular equipment, the others are able to compensate it and the system can proceed or if cumulative effects appear and diverge the system from its normal behavior. In some cases, these effects can be hardly detected during the logical analysis and the model might result not completely coherent with the reality.

This approach allows the establishment of a circular pattern (Figure 49) which implies changes to the logical analysis derived by the results of phenomenological modeling, until it adheres to reality.

The phenomenological modeling can provide a direct estimation of the consequences for the various sequences in order to obtain a direct risk estimation, the evaluation of the overall risk of the system and the expected value of consequence. The latter is calculated as a weighted average of the consequences, according to their probability.

The logical modeling can be performed through a dedicated software (IDDA 2.2) which can import the consequences and get an estimation of the risk.

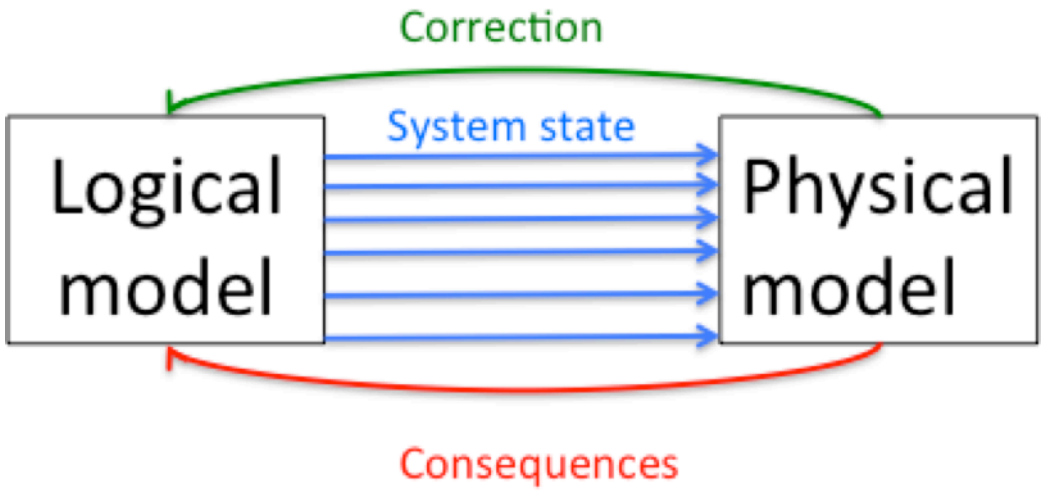


Figure 49: Diagram of the interaction between the logical model and the phenomenological in IDDA

3.4.1 Input file

The logical analysis is generated from an input file written with a special syntax. This section gives a quick overview of the syntax used in the preparation of the input file, that describes the system analyzed. (Galvagni, 1996)

Each level can be described as following:

nL m, P_1 ... P_m, r_1 ... r_m, i_0 ... i_m, 'cmL' 'cm0' ... 'cmm' Hide

nL: Level index; it is the level numerical label (handle) and identifies one (and only one) level. It has to be an Integer having value in the range 1...4095; it identifies the level for the addressing and the constraint definition.

m: Level order; it is the number of the possible level failure modes alternative to the default one (the number "0" or success mode); it has to be an INTEGER having value in the range 1...8. A binary level having just two exclusive possible choices, a success and a single failure, has order m=1.

P_1, P_2, ..., P_m: Failure modes probability; it represents the expectation degree of one of the possible level failure modes. It has to be a REAL number having value in the range: $0 \leq P_i \leq 1$. For multiple levels, having level order m>1, it must be also: $\sum_{i=1}^{m} P_i \leq 1$

r_1, r_2, ..., r_m: Failure modes probability uncertainty ratio. In the reliability problems the analyst often uses the failure rate (the inverse of the mean time between failures in a Poisson process) to provide the failure probability assessment in a given analysis period. Sometimes the statistically acquired data sheet does not report a failure rate value, but a failure rate probability distribution; we assume that distribution to be a LOG-NORMAL. In this case our probability description is based on a median probability value (it corresponds to the 50% of distribution area) and an uncertainty ratio that provides an estimate of the data dispersion around it. The uncertainty ratio is the ratio between the "maximum" value (95% of distribution area) and the "minimum" one (5% of distribution area) and must be a REAL number having value in the range $1 \leq r_i \leq 1/P_i$, where Pi is the related median probability. The program performs all the analysis assuming the "median" as the probability value but provides also, into the View Event Window and the Graphic Analysis tools, an estimate of the uncertainty data dispersion on the final probability distribution. If you are not using statistical data for the probability assessment, then the uncertainty ratio must be 1.

i_0, i_1, i_2, ..., i_m: Next level address; it points the next level depending on the output mode, hence its value has to be the index of a defined level; a next level address value i=0 means end of the sequence (terminal event tree branch). A level cannot points to itself.

'cmL': Level title; this characters string provides a level global description, common to all its output. It has to be enclosed between single apexes.

'cm0', 'cm1', 'cm2', ..., 'cmm': Level output string; it provides a comment depending on the current level output. They have to be enclosed between single apexes.

Hide: No print word flag; this optional parameter allows to hide some level output descriptions in the View Event sequence presentation procedure. A word flag allows to select just a subset of the possible level outputs; the word flag cannot contain any blank character, and the output identifiers have to be separated by pipe "|" characters. For example the word 0|2|4 hides the level success (0) and the level failure modes number 2 and number 4.

Comments are preceded by **!**.

After the definition of the various levels, it has to be defined how they interact and how they condition each other.

The first type of constraint is the address change constraint. It allows to change run time the logical path following a constrained level (called the target level), changing its "next level addresses" as the effect of a level (called the source level) given output achievement. It is a dynamic constraint because it becomes active only when the current sequence passes on a given source level output.

A nL flag, kL, i0 ... imk

A or **a:** This character identifies the constraint type.

nL: Source level index; identifies the level whose outputs activate the change, hence its value has to be the index of a defined level.

flag: Activation word flag; identifies the outputs that activate the change. It allows to select just a subset of the possible level outputs; the word flag cannot contain any blank character, and the output identifiers have to be separated by pipe "|" characters. For example the word 0|2|4 identifies the level success (0) and the level failure modes number 2 and number 4.

kL: Target level index; identifies the level whose addresses have to be changed, hence its value has to be the index of a defined level.

i0, i1, i2, ..., imk: New next level address (mk is the target level order); a new address value i=* means do not change, a new address value i=0 means end of the sequence, otherwise it has to be a level index.

Another type of constraint is logical constraint. It allows to compel run time the output of a constrained level (called the target level), as the effect of a level (called the source level) given output achievement. It is a dynamic constraint because it becomes active only when the current sequence passes on a given source level output.

L nL flag, kL, kv_flag, ki

L or **l:** This character identifies the constraint type.

nL: Source level index; identifies the level whose outputs activate the constraint, hence its value has to be the index of a defined level.

flag: Activation word flag; identifies the outputs that activate the change. It allows to select just a subset of the possible level outputs; the word flag cannot contain any blank character, and the output identifiers have to be separated by pipe "|" characters. For example the word 0|2|4 identifies the level success (0) and the level failure modes number 2 and number 4.

kL: Target level index; identifies the level whose output must be compelled, hence its value has to be the index of a defined level.

kv_flag: Forced target level output word flag; if the logical constraint force in any activation flag case the same status, it has to be a single integer having value in the range 0...mk (mk is the target level order). Else it must be a word flag containing the same output number of the activation one; each source level output in the activation flag will force the related output on the target level.

ki: Constraint intensity; this optional parameter allows to define a hierarchy on the logical and the probabilistic constraints. It has to be an integer having value in the range 1....255 (the default value is ki=1); a logical or probabilistic constraint is active if and only if the target level probability in the past was not be modified by a constraint having a greater intensity. To avoid to override a compelled logical status using a probabilistic constraint, you can set always an high value for this parameter.

Another type of constraint is probabilistic constraint. It allows to change run time the expectation degree spreading relevant to a constrained level (called the target level), changing its failure mode probability values as the effect of a level (called the source level) given output achievement. It is a dynamic constraint because it becomes active only when the current sequence passes on a given source level output.

P nL flag, kL, P1 ... Pmk, r1 ... rmk, ki

P or **p:** This character identifies the constraint type.

nL: Source level index; identifies the level whose outputs activate the change, hence its value has to be the index of a defined level.

flag: Activation word flag; identifies the outputs that activate the change. It allows to select just a subset of the possible level outputs; the word flag cannot contain any blank character, and the output identifiers have to be separated by pipe "|" characters. For example the word 0|2|4 identifies the level success (0) and the level failure modes number 2 and number 4.

kL: Target level index; identifies the level whose probabilities have to be changed, hence its value has to be the index of a defined level.

P1, P2, ..., Pmk: Target level failure modes new probability (mk is the target level order). It must satisfy the same conditions of the level instruction probability.

r1, r2, ..., rmk: Target level failure modes new probability uncertainty ratio (mk is the target level order). It must satisfy the same conditions of the level instruction probability uncertainty ratio. If you are not using failure rate statistical data for the probability assessment, it must be 1.

ki: Constraint intensity; this optional parameter allows to define a hierarchy on the logical and the probabilistic constraints. It has to be an integer having value in the range 1....255 (the default value is ki=1); a logical or probabilistic constraint is active if and only if the target level probability in the past was not be modified by a constraint having a greater intensity.

There are other types of constraints that were not used in this case, and that will not be described here.

3.4.2 Connection between logical and phenomenological modeling

The phenomenological modeling is usually operated by dedicated software that are interfaced with IDDA. The phenomenological modeling software must have some characteristic such as:

1. Allowing the automatic variation of the parameters used in the simulation , without intervention by the operator;
1. automatically saving of the results of the modeling;
2. Exporting a numeric value for the consequences, which will be imported from IDDA to estimate the risk.

The software IDDA produces a set of files containing the descriptions of the system analyzed. This description is constituted of an indexed structure in binary code, contained in a series of vectors. The indexes used are the following: in case of success (the first output of the level) Level = x + (m-1) * 4096 where m is the number of possible types of outputs except the first, this index is always upper than zero. Instead in case of failure (the outputs of the level excluding the first) x = - Level - (i – 1) * 4096, where i is the number of the output (excluding the first), in this case the index is always negative .

For binary events in case of success m = 1 and x = Level, in cases of failure i = 1 and x = -Level.

It can be convenient to interpose a special software written ad hoc between the indexes issued by IDDA and the phenomenological model: this program has to be able to read and interpret a vector of indexes, to change the parameters of modeling and after this, launch the phenomenological modeling. In example, if the level 10 asks if a certain valve is open, regarded as successful, or closed, regarded as a failure, the program shall verify each time if the index 10 or -10 is present in the vector ,and will consequently change the phenomenological modeling.

The phenomenological modeling software should return the value of the consequences as a real number; the interface software between the two modeling will save it in a correct format readable by IDDA.

In addition, the phenomenological modeling can return a description of how the system behaves. Finally, the analyst can verify that the logical description is in agreement with what is described by the phenomenological modeling. After that, the logical modeling can be changed and consequently the event trees can be extracted as a function of process conditions.

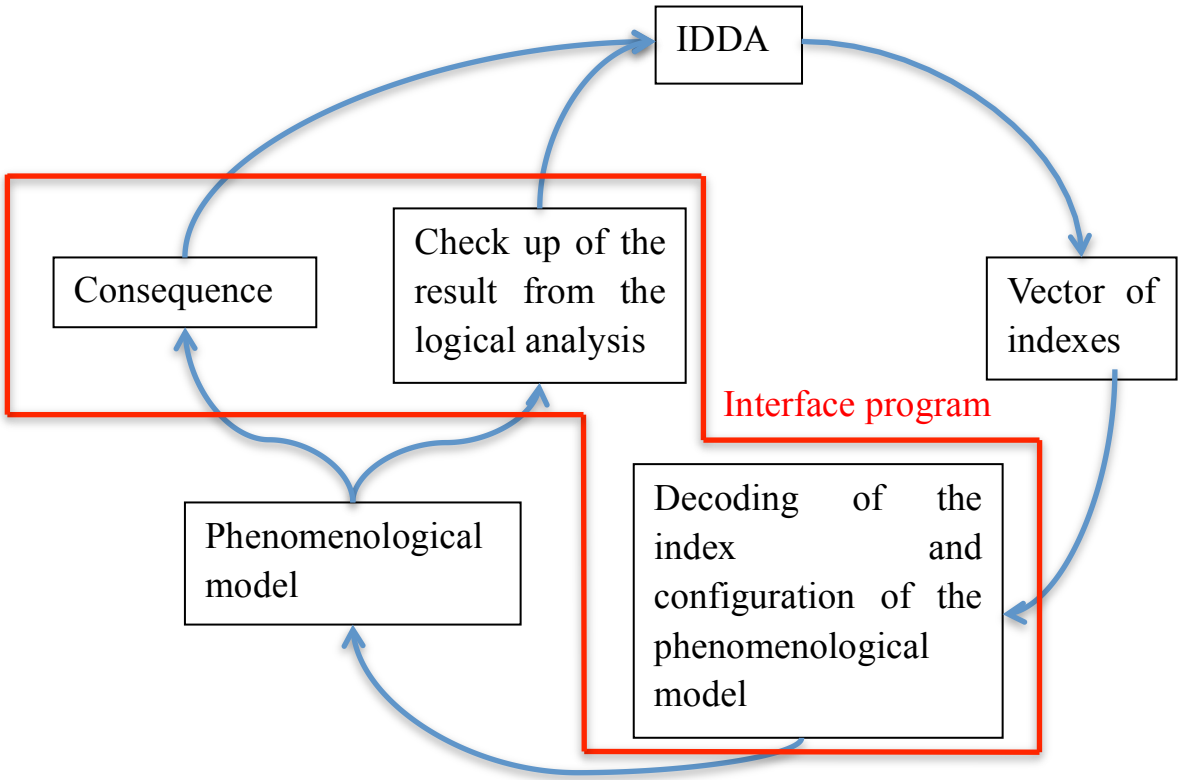


Figure 50: Operating diagram of the interaction between the logical and phenomenological modeling