A blended hazard identification methodology to support process diagnosis

Benjamin J. Seligmann[a], Erzsébet Németh[a], Katalin M. Hangos[b,c], Ian T. Cameron[a,*]

[a]*School of Chemical Engineering, The University of Queensland, Brisbane, QLD, Australia 4072*
[b]*Process Control Research Group, HAS Computer and Automation Research Institute, Budapest, Hungary 1111*
[c]*Department of Electrical Engineering and Information Systems, University of Pannonia, Veszprém, Hungary 8200*

Abstract

A novel hazard identification methodology applied to process systems is presented in this paper. This Blended Hazard Identification (BLHAZID) methodology blends two different types of HAZID method: the function-driven and component-driven approach. The BLHAZID method is based on a conceptual framework called the Functional Systems Framework, which describes structure–function–goal relationships in process systems.

The goals of the BLHAZID methodology are to generate outcomes that contain a high coverage of hazards, describe detailed failure causality in process systems and express this knowledge in a structured form for effective reused in subsequent applications, such as fault diagnosis, operator training, design reviews, fault and event tree construction and hazard updates to satisfy major hazard facility requirements.

Both the BLHAZID methodology and the Functional Systems Framework were developed with involvement and advice from two major industrial partners. An industrial case study of a benzene saturation unit is presented to illustrate how the BLHAZID methodology operates in practice.

Keywords: hazard identification, failure, function, blending, causality, knowledge

*Corresponding author.
Email address: i.cameron@uq.edu.au (Ian T. Cameron)

1. Introduction

Preventing and mitigating serious accidents is a major goal in the process industries. Major accidents continue to occur, as evidenced by Bhopal (1984), Longford (1998) and recently Deepwater Horizon (2010). This is in spite of the developments in hazard identification (HAZID) techniques over the last 40 years. Hazard identification, as the first major step of risk management, is a crucial activity for reducing accidents and other operability related losses. Therefore a change in HAZID practice may be necessary to address the issues raised by ongoing accident occurrence.

For hazard identification to be effective it is important to recognize process systems for what they are: socio-technical systems (Rasmussen and Petersen, 1999). Operating under the viewpoint that process systems are more than just equipment, material streams and control loops greatly affects the understanding of how these systems operate and hence how they can fail. The importance of having a sufficient scope for HAZID analysis is enhanced by the realization that between 40% and 70% of abnormal conditions in process systems are people related (Venkatasubramanian et al., 2003a; Fiske, 2009). Therefore a move towards more holistic and integrated frameworks and techniques for HAZID is important for reducing hazard, failure and accident occurrence. As the first part of this work, this paper presents a new method for identifying failures in plant components only.

Many different HAZID methods have been previously developed. These include the Safety Review, Checklist analysis, What-If analysis, Hazard and Operability (HAZOP) analysis and Failure Mode and Effects analysis (FMEA). These and other methods are described in many publications, including Schüller et al. (1997), Mannan (2005), CCPS (1992) and Cameron and Raman (2005).

One of the main drawbacks of traditional HAZID methods is that they tend to be expensive for companies to undertake, laborious and generate outcomes that can vary considerably due to the subjectivity and variability of analysis teams. This can lead to frustration among team members which may adversely affect the outcomes (Trammell and Davis, 2002). Due to the laborious nature of HAZID analyses, there have been efforts in the past to augment HAZID analysis with the use of computer aided tools. Venkatasubramanian et al. (2000), McCoy et al. (1999) and Dunjó et al. (2009) all con-

tain thorough reviews of the development in automated and computer aided methods for HAZID. Thus, a detailed review of computer aided methods will not be discussed here. The major efforts in this area, for example HAZOPExpert (Venkatasubramanian and Vaidhyanathan, 1994) (Vaidhyanathan et al., 1996), tend to focus on the emulation of traditional HAZID methods using qualitative models such as the signed-directed graph (Venkatasubramanian and Vaidhyanathan, 1994; McCoy et al., 2006).

Reusing the knowledge generated during a HAZID analysis is critical for implementing the desired corrective actions. The computer aided HAZID work devotes a lot of effort towards managing the different types of knowledge that are relevant during HAZID analysis. However, knowledge representation and reuse issues associated with HAZID are significant and the solutions are non-trivial. In light of this, investigating effective reuse of HAZID outcomes is a key feature of the current work. Any change in HAZID practice will likely include improved knowledge management techniques and approaches. As an example of the growing emphasis on knowledge management in the process industries, Pasman (2009) comments, 'are we making progress in learning from past accidents? Yes, there is progress, but the efficiency could be much higher'. The introduction of process systems data modelling standards, such as ISO 15926 (ISO, 2003, 2004), or ontologies (Gruber, 1993) that formally describe process system concepts, for example OntoCAPE (Marquardt et al., 2010) or the work of Batres et al. (2006), can be used for improving process system knowledge representation and management.

A systematic approach to HAZID, based on fundamental HAZID theory, is required for better supporting fault diagnosis, which in turn will contribute to reducing the occurrence of accidents. In response to this need in the field of HAZID, a foundational conceptual framework, called the Functional Systems Framework (FSF) (Cameron et al., 2007, 2008; Seligmann et al., 2009; Németh et al., 2009; Seligmann et al., 2010; Cameron et al., 2010), was developed to support the creation of a novel HAZID methodology, developed in this paper, which is based on the *blending* of two fundamentally different types of HAZID method, exemplified by HAZOP and FMEA respectively. Blending HAZID methods is a strategy for improving the amount of hazards and failures identified, called the *coverage*, and producing detailed causal knowledge. Graham (2005) and Trammell and Davis (2002) have investigated various combinations of HAZOP and FMEA for different application areas. In this work HAZOP was chosen because it is used widely across the process industries, is familiar to many industrial personnel and is focussed on

examining stream deviations. Whilst FMEA is not widely used in the process industries, it is complementary to HAZOP since it focusses on analysing failures in equipment and can be extended to people and procedures. The resultant HAZID method was called the Blended Hazard Identification (BLHAZID) methodology. Previous versions of the BLHAZID methodology have been presented by Cameron et al. (2007, 2008); Seligmann et al. (2009); Németh et al. (2009); Seligmann et al. (2010); Cameron et al. (2010).

The intention of the BLHAZID method is to generate outcomes that contain structured knowledge of causality in process systems undergoing failure. In effect, the knowledge that is generated is a causal model, using a structured language to describe failures and their causal relationships. The knowledge contained in the outcomes is clear and unambiguous, so that it can be used effectively in a variety of subsequent applications which include fault diagnosis (Németh et al., 2007), operator training, design reviews, fault and event tree construction, auditing and hazard updates to satisfy Major Hazard Facility (MHF) requirements.

Section 2 describes the FSF and shows how it functions as a theoretical foundation for the BLHAZID methodology. The concept behind blending different HAZID methods that has been used is also outlined, before describing the BLHAZID methodology in Section 3. The structured language used to express knowledge associated with the BLHAZID methodology is described in Section 4. Section 5 contains an industrial case study of a benzene saturation unit (BSU) to illustrate some of the main features of the BLHAZID. The discussion in Section 6 contains a number of reflections on some of the main characteristics of the BLHAZID methodology, including a comparison between the BLHAZID and other major HAZID approaches.

2. Theoretical foundations and blending hazard identification methods

The concepts in the FSF are based on general systems theory (von Bertalanffy, 1968) and specifically the description of systems concepts found in the ontology of Bunge (1977, 1979), and serves as a theoretical foundation for the development of the BLHAZID methodology.

2.1. The Functional Systems Framework (FSF)

The FSF describes the structure–function–goal relationships within a process system and is shown in Figure 1. The process system structure is made

up of *components*, which are either plant equipment, people or procedures, *connected* to the various types of *streams*. These stream types include material streams, signals in control systems or communication between people, and are modelled as generalized *information* streams (Németh et al., 2009). Just as streams carry information expressed by properties, such as temperature, pressure or concentration, holdups in vessels and equipment can also have properties of interest, such as mass, level or liquid-vapour ratio.

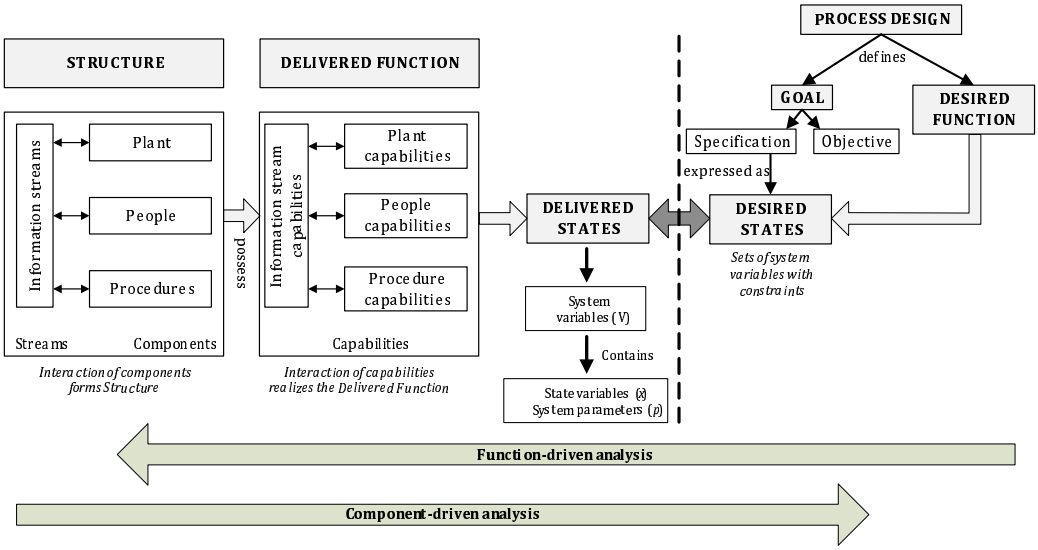

Figure 1: HAZID methods viewed in the FSF

All components and streams have *capability sets* (Németh et al., 2009). A capability (Cameron et al., 2010; Jalashgar, 1999; Németh et al., 2009; Seligmann et al., 2010) is the ability of a component or stream to *act* in a certain way, ultimately to affect the state of the system, and is a central concept in the FSF. A capability is described as an <action> on a <property>. Capabilities of process equipment include:

- a pump can <increase><pressure>
- a reactor can <increase><extent of reaction>
- tanks can <hold><mass>
- pipes can <permit><flow>

- valves can <stop><flow>
- a natural gas stream can <supply><heat> or <react with><oxygen> during combustion

When the components and streams are connected together to form the structure, certain capabilities are *activated*. For example, heat transfer in a shell and tube heat exchanger can only occur *when the streams are actually passing through* the exchanger. The hot stream must be *connected* to the tubes of the exchanger before heat can be transferred. The set of all activated capabilities is the *function* of the system. The function meets system *goals*, by delivering the desired system *state*. An example of this is where there is a desired heat flux that the heat exchanger should deliver in order to meet the goal of raising the temperature of the cold stream by a particular amount. In order for the goals to be met, the delivered function must be the same as the desired function specified by the process system design.

2.2. Component-driven and function-driven hazard identification methods

HAZOP and FMEA are representative of two fundamentally different approaches to HAZID. HAZOP is an example of a function-driven approach and FMEA of a component-driven approach. The terms 'function-driven' and 'component-driven' are derived from the FSF. Classifying HAZID methods as either function- or component-driven is actually interpreting those methods in light of the FSF.

Viewing HAZOP in light of the FSF shows it to be a *function-driven* method since it investigates how the function of a system is lost or degraded. Questions that are asked during HAZOP sessions are related to how loss of function occurs so that the design goals are not met, such as, "how could this system not deliver its design intention?". Conversely, viewing FMEA in the FSF shows it to be a *component-driven* method, since it seeks to identify failures in the components of a system and elicit the effects of these failures on the function of the system. In fact, many hazard identification methods currently used in industry can be classified as either function or component-driven methods, as shown in Table 1.

2.3. Blending component-driven and function-driven HAZID methods

Investigating how HAZOP and FMEA may be blended provided insights into how to blend the more fundamental function-driven and component-driven approaches. The purpose of blending is to take advantage of the

Table 1: Function and component-driven classification of well-known HAZID methods

METHOD	Component-driven	Function-driven
Checklist Analysis		X
FMEA	X	
Fault Tree Analysis	X	
HAZOP		X
Preliminary Hazard Analysis	X	
Relative Ranking	X	
Safety Review		X
What-If Analysis		X

strengths of each method whilst compensating for their weaknesses. It is important to note that *combining* HAZID methods is not the same as *blending* them. For example, Dunjó et al. (2009) refers to different efforts to combine HAZOP and fault tree analysis (FTA). In a number of these combinations, HAZOP was performed first to identify hazards and *then* FTA used to quantify the frequency of the hazards. A truly blended method, however, is where the fundamental elements of two or more methods are identified and a new, single, methodology is formed from these elements. It is crucial to understand how these elements fit together so that the methods can be blended effectively. Blending requires a deep knowledge of how the methods operate and the fundamental concepts underlying them. The FSF is a conceptual framework that can be used to classify and understand the fundamental concepts of HAZID methods, thus providing a consistent framework for how to identify their fundamental elements and blend them.

Function-driven and component-driven HAZID approaches are both *complementary* and *overlapping*. They are overlapping because both identify failure causation in the *same* system. Failure events and their causes and implications are fundamentally the same no matter which method is used to identify them. Also, certain failure events are more easily identified with particular methods. For example, the relationship between a *leak in a pipe* and the *corrosion* that caused it can be more easily identified with FMEA than with HAZOP, since FMEA is concerned with establishing the relationship between failure modes and failure mode causes. The effect of the leak on the process system function as a whole, however, is more effectively investigated with HAZOP, since HAZOP can be effectively used to identify downstream

consequences. Thus, different methods can be complementary to each other. Blending a function-driven and a component-driven approach together can theoretically yield outcomes with a higher coverage of hazards than using them separately.

The main 'failures' identified during a HAZOP are *deviations* in system variables that affect the system state, which are variables usually associated with material, energy or signal *streams*. For example, if quality constraints set on gasoline products from petroleum refineries are not met, it can be said that a deviation in the product stream has occurred. Equipment failures are examined in a HAZOP, but only as causes or consequences of streams failures. In contrast, FMEA is primarily concerned with *equipment* failures. Therefore, HAZOP and FMEA are complementary since HAZOP generally focusses on streams and FMEA on equipment.

The main idea with blending HAZOP and FMEA is that by considering the function and the structure of a system, failures are identified in both the components and streams, yielding a methodology able to produce outcomes of increased scope and hence higher coverage, as well as elucidate detailed causal pathways, better than either method separately.

3. The blended hazard identification methodology

Using the FSF as an underlying framework, a novel HAZID method, called the Blended Hazard Identification (BLHAZID) methodology, has been developed. The BLHAZID method was constructed directly from the FSF while using concepts from both HAZOP and FMEA and the blending of these methods from Graham (2005). Figure 2 shows the workflow of the BLHAZID methodology.

3.1. System decomposition and subsystem level BLHAZID analysis

After selecting the system for analysis, the next major step of the BLHAZID is to decompose the system into subsystems. BLHAZID analysis is performed within one subsystem at a time. This is done in order to support the analysis team by reducing the complexity of the analysis: one subsystem can be focussed on at a time.

The structure of a subsystem, which is based on concepts from the FSF, is presented in Figure 3, showing both internal and boundary variables. A subsystem is shown with its boundary, inlets and outlets. The boundary is between different subsystems or between a subsystem and the environment

1. Select system for BLHAZID analysis and designate system boundary.
2. Decomposition of system into subsystems.
3. For each subsystem:
    a. Identify the initial set of characterizing variables (c-vars).
    b. For each characterizing variable (c-var):
        i. Generate deviations.
        ii. For each deviation:
            - Elicit possible causes.
            - For each possible cause-deviation pair:
                - Elicit implications.
    c. For each component:
        i. Elicit failure modes (FM).
        ii. For each FM:
            - Elicit failure mode causes (FMC).
            - Elicit implications.
    d. Collect new c-vars, deviations and failure modes that have been identified throughout the analysis. Repeat analysis until no new information is elicited. This includes c-vars or deviations related to failures that have been passed to the current subsystem from another connected subsystem that has been previously analysed.
    e. Collate consequence list.
4. Review of failures passed between subsystems.
5. Final review.
6. Qualitative risk assessment and recommended action.

Function-driven analysis

Component-driven analysis

Figure 2: The workflow of the Blended HAZID methodology

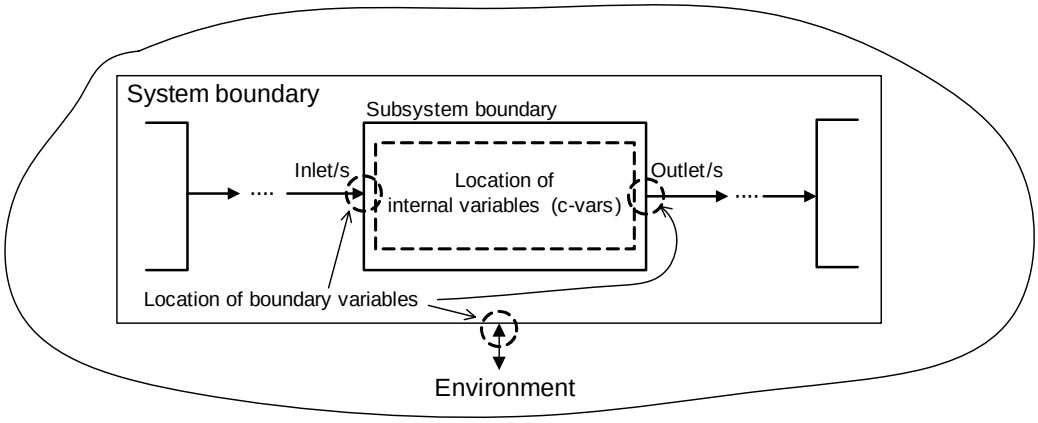

Figure 3: Subsystem structure: internal and boundary variables

of the system. For example, if a leak in a tank causes gas to escape the gas will be lost to the environment directly, without passing through any other subsystems.

A structured decomposition procedure has been developed to decompose a process system represented on a Piping and Instrumentation diagram (P&ID) into subsystems representing the main functional units of the system (Németh et al., 2009). All of the subsystems are logically linked together by specified inlet and outlet ports. Failures are then identified within each of these subsystems in turn, similar to analysis nodes in a HAZOP.

A software tool is in development for extracting process-specific knowledge from intelligent P&IDs and utilizing it to perform automated decomposition. This approach is discussed in Németh et al. (2009). The analysis team must review the decomposition results and should make changes as appropriate.

After the decomposition is complete, a subsystem is selected and then the core of the BLHAZID analysis begins. The analysis has two main parts: function-driven analysis and component-driven analysis.

3.2. Function-driven analysis

Each subsystem included in the design of a process system is associated with goals in the form of *characterizing variables*, or c-vars, that have constraints upon them (Seligmann et al., 2010). Characterizing variables are types of design variables, such as desired pressures or temperatures, but are specifically associated with internal subsystem variables. Focussing on internal variables helps maintain the systematic nature of the BLHAZID methodology, where failures are always considered to occur *inside* a subsystem with causes and implications propagating from the inside across the boundary to other subsystems.

Once a subsystem has been selected for analysis, the characterizing variables of that subsystem are identified in order to identify potential *deviations* in the value of those c-vars. Deviations in variables are also called *functional failures* since they are related to a loss or degradation of system function. Unlike a HAZOP, only a specific set of c-vars are *initially* identified. These are associated with variables that are most relevant for describing the function of the subsystem. For example, in certain reactors the 'liquid-vapor ratio', 'catalyst activity' and 'extent of reaction' may be more relevant for describing the function than the 'flow' of material through the reactor.

The identification of c-vars occurs through a discussion guided by a number of knowledge sources. This discussion is similar to the brainstorming aspect of a HAZOP analysis. The systematic approach of examining the different knowledge sources is complemented with a general discussion of function, allowing significant hazard scenarios to be identified and linked to deviations in particular c-vars, similar to the approach by Rossing et al. (2010) for identifying HAZOP deviations. The knowledge sources are:

Measured variables indicated on the P&ID These variables are identified during the design phase of the system as important variables that require monitoring. Since they are important to system operation they are included as c-vars.

Capability sets of components Examining the capability sets of each subsystem component reveals the variables affected when components act on streams, hence giving insights into what are the c-vars.

Physical and chemical processes occurring in the subsystem Examining the physical or chemical processes occurring in the subsystem can yield insights into what the significant c-vars are. For example, in a distillation column the temperature of the feed is important for effective operation, and therefore TEMPERATURE will be a c-var.

Design documents The design documents generally contain details of variables that are important to system operation and hence can be identified as c-vars, but do not have measurements associated with them on the P&ID.

Failures passed from a connected subsystem If a failure, in the form of a deviation in a stream, has been passed downstream to a subsystem then the variable that is associated with the deviation becomes an important c-var to consider in the current subsystem under analysis. This is also the case for process recycles where, foreseeably, deviations could be passed to subsystems *upstream* of where the deviation originated.

Once the initial set of c-vars are generated, a single c-var is selected and appropriate guidewords are applied to form deviations, as in HAZOP (SA, 2003). A deviation is formed by concatenating a guideword with a c-var, resulting in <guideword><c-var>. Examples include <high><temperature>

or <low><concentration>. One deviation is selected and causes and implications of that deviation are elicited.

BLHAZID analysis within subsystems is crucial in order to generate causal knowledge that can be used for subsequent activities such as fault diagnosis. The knowledge must be both detailed and free from ambiguity so that failures can be traced across subsystems boundaries and throughout the system. To ensure this, the search for causes is limited to the subsystem boundary.

Causes identified on the boundary are able to be linked to other failure events in other subsystems since the inlet and outlet ports of adjacent subsystems are connected together. Therefore, the types of causes are limited to the following set:

- deviations in other variables within the subsystem
- deviations in other variables at subsystem inlets or outlets; that is, on the subsystem boundary
- deviations in other variables at an environment port; that is, a connection to the environment
- component failures; that is, failure modes in components

After causes for the selected deviation have been elicited, logical implications are identified. The types of implications are the same as the types of causes.

In traditional HAZID studies, the term *consequence* has been used to express the significant events that are caused by failures or hazardous situations. However, a distinction is drawn here between the terms *implication* and *consequence*. An implication is considered here as a logical implication that causally flows from a variable deviation or component failure mode. These outcomes are linked across logically connected subsystems allowing knowledge of system causality to be constructed. A consequence is regarded as an implication that generates a significant impact or hazardous situation. In the outcomes of the BLHAZID analysis there may be many implications, of which a subset would be consequences. A list of consequences is extracted from the list of implications at the conclusion of the analysis of each subsystem. Particular attention needs to be given to consequences so that effective decisions can be made for preventing or mitigating them.

In the BLHAZID methodology, implications of deviations are taken as being implications of a pair (cause, deviation) in order to generate more detailed causal knowledge. That is, an implication does not occur simply as the result of a deviation but of a deviation resulting from a *particular cause*. This allows the causal structure of the triplet (cause, deviation, implication) to be generated and captured. Capturing causal knowledge in the form of triplets during BLHAZID analysis helps the analysis team to think deeply about system causality. As well as this, the performance of inferencing by diagnostic tools on the knowledge captured in the BLHAZID outcomes is supported and enhanced. This is because the establishment of causal links between deviations, causes and implications is begun during the BLHAZID analysis.

After the causes and implications of the selected deviation are elicited, another deviation is selected and the procedure repeated. This occurs until all relevant deviations have been investigated. At this point a new c-var is chosen and analysed. All the deviations are examined for each c-var in the subsystem. Once the analysis of all c-vars is complete, the component-driven analysis is commenced.

3.3. Component-driven analysis

The component-driven section of the BLHAZID methodology analyses failure modes of components in the subsystem and their causes and implications.

To apply the definition from Rausand and Oien (1996), a failure mode is the observation of a failure in equipment, and is also called a component failure. The description of equipment failure modes should therefore follow this definition and be more closely related to what can actually be observed. For example, <leak> rather than <material damage>.

Component failure modes (FMs) in the subsystem are either elicited during the analysis or established *a priori*. Failure modes of a component type can be classed as static knowledge, which is application independent. The specific context of a component may affect its set of failure modes, and therefore a review of the *a priori* FMs should occur for each component type in the subsystem before the analysis commences.

Rausand and Oien (1996) presented a scheme linking failure modes with the operational modes of the component in question. This view is supported by OREDA (2009). We have extended this by including the concept of *activated capabilities*. This is an extension of the capability concept described

Table 2: Pump: operational modes and capabilities

Operational Mode	**Activated Capabilities**		
ON	Hold mass	Permit flow	Increase pressure
OFF	Hold mass	Permit flow	

in Subsection 2.2. A component or stream always have particular capabilities, but these are not always used to contribute to system function. Different operational modes specify which capabilities should be activated in order to deliver the correct function, shown in Table 2.

When the operational mode for a pump is ON the activated capability set for the pump should be <hold><mass> and <increase><pressure>; when the pump is OFF the pump should only <hold><mass> and should cease to <increase><pressure>.

The particular activated capabilities are related to the failure modes of the component. For example, a failure mode <leak> is related to a loss of the capability to <hold><mass>. Examining the activated capability set of a component, complemented by reference to such works as OREDA (2009), supports the elicitation of the failure modes of that component. The concept of activated capabilities provides a simple and systematic way of identifying the desired function of process system components and hence their failure modes.

Once the list of failure modes is compiled, a single failure mode is selected and its causes are elicited. There are a number of appropriate knowledge sources of failure modes causes (FMCs). For example the OREDA database (OREDA, 2009) contains information about the failure modes and failure mode causes of various equipment types. The analysis team should augment such knowledge sources with their own information on the specifics of the process system under analysis.

The procedure for identifying failure mode causes can be less involved than for identifying causes of deviations. This is because it is assumed that failure mode causes, as well as the failure modes themselves, are generally considered to be static knowledge and therefore can be elicited a priori to the BLHAZID analysis. Therefore, an a priori database can be used as a primary knowledge source for the failure mode causes.

For simplicity, failure mode causes are expressed as deviations, in either internal of boundary variables, in streams connected to the component where the failure mode is being observed. Therefore:

- a failure mode of a component is considered to have no failure mode causes and is itself expressed as a root cause when a physical failure has occurred due to spontaneous degradation in the component
- a failure mode of a component has failure mode causes, expressed as functional failures, when a part failure has occurred due to stream deviations affecting the parts of the component

Once the failure mode causes are identified for the selected failure mode then the implications of that failure mode are elicited.

Unlike the implications of deviations, the implications of failure modes are not the result of a pair (FMC, FM). This is because failure modes have the same effect regardless of the particular cause. For example, a leak in a pipe will lead to a flow of material to the environment regardless of whether the leak was caused by corrosion or high pressure. Quantitatively there could be a difference in flowrates, however this is not relevant since the BLHAZID method is used to generate *qualitative* causal knowledge.

These are the possible types of failure mode implications:

- deviations in stream variables within the subsystem;
- deviations in stream variables at subsystem inlets or outlets;
- deviations in stream variables at an environment port.

The component-driven analysis steps of choosing a failure mode and eliciting its causes and implications are repeated for each component in the selected subsystem.

3.4. New characterizing variables or failure modes

New c-vars or failure modes may be identified throughout the course of the subsystem analysis that were not initially identified. These new c-vars are now added to the initial list of c-vars. Deviations of those newly identified c-vars are then analysed. This is repeated for newly identified failure modes.

3.5. Collation of consequences

After the subsystem has been analysed the list of *consequences* is collated. The list of consequences is a subset of all the implications generated during the analysis. These consequences require subsequent discussion by relevant

personnel so that appropriate action can be taken to prevent them or plan mitigation strategies.

Once the consequences have been collated a non-analysed subsystem is selected and then analysed. This procedure is repeated until all subsystems have undergone BLHAZID analysis.

3.6. Review of failures passed between subsystems

This review step should investigate whether any subsystems require further analysis. It may be necessary that a subsystem that has already been analysed has new failures passed to it that require further examination. These failures may be related to a completely new c-var that was not analysed initially, or a new deviation of a c-var that was not initially taken into consideration. For example, it may become apparent that a reverse flow situation in a particular subsystem, which was not initially identified has become relevant through analysis of the rest of the system.

3.7. Final review

A final review is then performed to check that causes and implications were elicited for all relevant c-vars in all subsystems, and that all component failure modes were examined to elicit relevant causes and implications.

3.8. Qualitative risk assessment and recommended action

In the software tool that has been developed to perform the BLHAZID analysis there is functionality to attribute extra information to the failures contained in the BLHAZID outcomes. For each failure identified the following information can be added in the tool: general *comments*, how the failure may be *detected*, what *action* can be taken to prevent or mitigate the failure, a specification of *personnel responsible* to ensure that the action is carried out and a *status* showing whether the action has occurred. Risk-related data can also be added. The analysis team is able to define likelihood and severity levels in the software tool to produce a user-defined risk matrix. The likelihood and severity of each failure can be described, giving a level of risk for that failure.

4. Structured language for BLHAZID methodology

A formally represented knowledge is based on a conceptualization that involves the entities and the relationships that exist amongst the entities

defined by the FSF. An ontology - as an explicit formal specification of terms in the domain and relationships among terms defines a common vocabulary, shares a common understanding of structured information among people or software systems and enables reuse of domain knowledge (Gruber, 1993).

For process system BLHAZID analysis, three different knowledge types have been distinguished:

1. *General process system (a priori) knowledge*: contains the generic knowledge about different component types (like capabilities, failure modes) and variable types with their relevant guide words accessible during the BLHAZID analysis;
2. *Process-specific knowledge*: describes the components, their connections, and subsystem decomposition related to process system be analyzed;
3. *BLHAZID generated knowledge*: all type of failures (functional, component, environment, part), and the causal relationships between them.

Vaidhyanathan et al. (1996) used similar process-specific concepts as compared to process-general knowledge in the expert system based HAZOP emulator, HAZOPExpert.

Figure 4 shows an example of how the different types of knowledge are used to describe some of the knowledge associated with an illustrative material transfer subsystem. Table 3 shows the main elements of the formal structured language used to express knowledge associated with the BLHAZID methodology.

4.1. Reusing BLHAZID knowledge

The effective and systematic generation of causal knowledge to aid diagnosis, operator training and process design is a vital aspect in overall process risk management. The underlying ontologies combined with the BLHAZID methodology used knowledge and generated outcomes that permit a range of applications related to process risk management. Causal knowledge of process systems as an outcome of the BLHAZID analysis provides the basis for a wide range of applications, such as diagnosis tools, operator training systems, planning, as well as process and control system retrofit and design, aiding in auditing requirements under major hazard facility regulations.

This structured BLHAZID methodology generates knowledge that is easily accessible to inference engines that can elucidate potential root causes

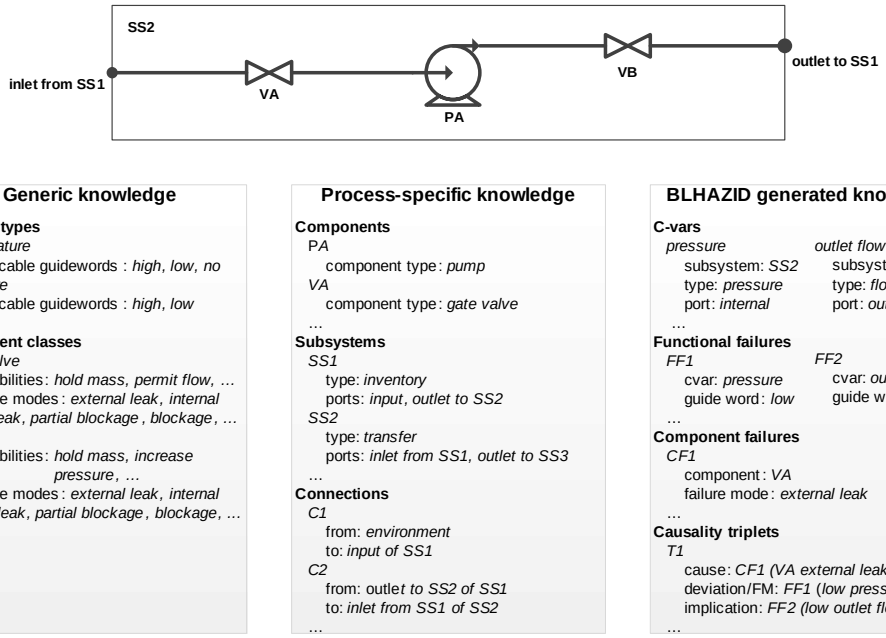

Figure 4: Illustration of the different types of knowledge for a simple transfer subsystem

Table 3: Main elements of the structured language elements used in BLHAZID methodology

Concept	**Syntax**	**Example**
Capability	<action><property>	<increase><pressure>
Component failure (CF)	<component> <failure mode>	<VA> <blockage>
Functional failure (FF)	<guideword><c-var>	<high> <temperature>
BLHAZID triplet (cause, deviation, implication)	(FF, FF, FF) or (CF, FF, FF) or (CF, FF, CF) or (FF, CF, FF) or (CF, CF, FF) or	(<high><inlet temperature>, <high><temperature>, <high><pressure>) (<high><pressure>, <TA><rupture>, <high><flow to environment>)

and the implications related to failures. The adopted structured language, including the causal relationships, captured as semantic triplets during the

BLHAZID analysis, facilitates the determination of the failure propagation through the system, and the determination of potential root causes and possible consequences of a deviation using backward and forward reasoning. The generated causal knowledge represented in a structured language is amenable to visualization of the causal knowledge as cause-implication graphs or causal graphs (Németh et al., 2011). The nodes of the graph are the failures and each edge represents a causal relationship between nodes. Generating and using causal graphs is a powerful approach for allowing process personnel to quickly visualize and diagnose causes and implications of a failure. Causal graphs have great utility in operator training, on-line diagnosis and application to design decisions.

In order to better understand the BLHAZID approach, an industrial case study is presented and the outcomes discussed. A number of causal graphs are shown in the case study to show their utility for supporting process operations.

5. Case study: Benzene Saturation Unit

A Benzene Saturation Unit (BSU) is studied here to show the utility of the BLHAZID methodology for generating HAZID knowledge of intricate and industrially significant failure scenarios. The purpose of this case study is threefold:

- to show the outcomes of decomposing a system into subsystems and BLHAZID analysis within subsystems;
- to present how the BLHAZID is performed for large process vessels, material transfer subsystems and control systems;
- to emphasize the benefits of using a structured language for representing BLHAZID outcomes and the subsystem representation for supporting effective knowledge reuse.

5.1. System description

The BSU removes benzene from a hydrocarbon feed via a catalytic hydrogenation reaction where benzene is converted to cyclohexane. The benzene-rich feed is mixed with hydrogen and a large recycle stream prior to entering reactor 940D. The effluent from the reactor is separated into gas and liquid streams in separator drum 944F. Most of the liquid stream is recycled into

the feed and the remainder sent downstream for further processing. The liquid recycle dilutes the benzene-rich feed. Due to the exothermic nature of the hydrogenation reaction, if the temperature in the reactor becomes too high then an undesirable hydrogenolysis reaction can occur, leading to temperature runaway and potentially causing physical damage to the system. The recycle flow can be directed through fin-fan heat exchanger 948C by the action of control system *TC9465* to control the inlet temperature to the reactor.

A simplified P&ID of the BSU process is shown in Figure 5. Figure 5 also shows the subsystems identified in the decomposition process. This simplified view is very useful during BLHAZID analysis since both the subsystems and components in the system are shown on the same diagram.

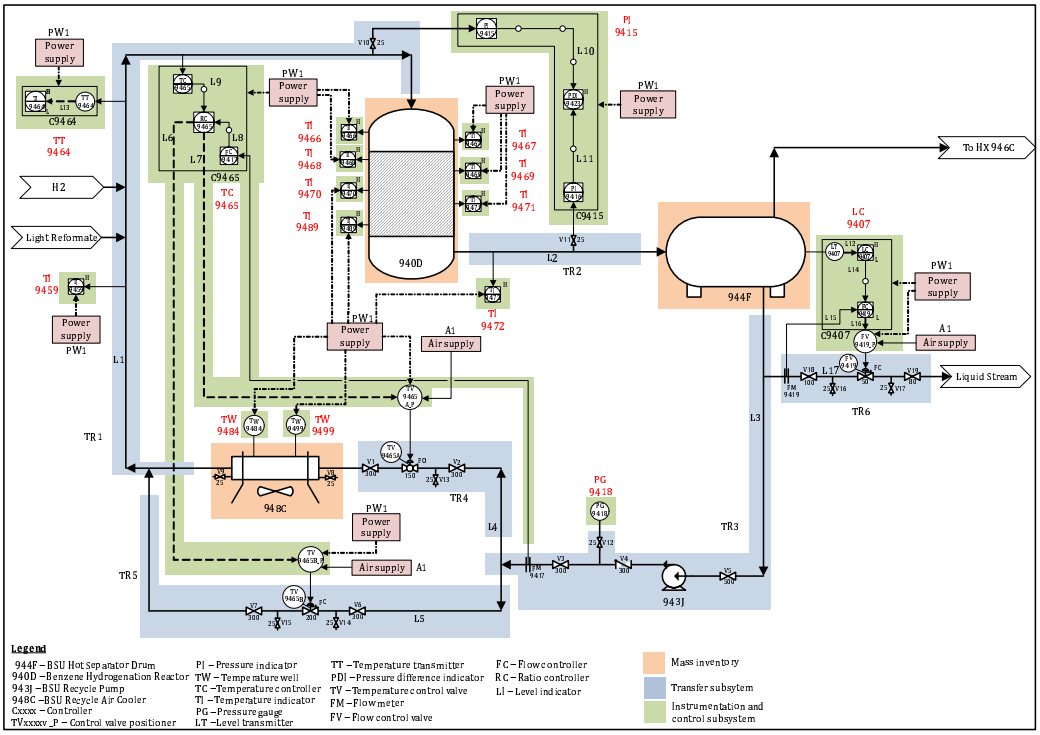

Figure 5: Benzene saturation unit

5.2. System decomposition

The decomposition process breaks the system up into subsystems. There are three types of subsystems identified in the BLHAZID analysis: mass

inventories, transfer subsystems and control subsystems. Each component in the BSU was identified as belonging to one of the subsystems shown in Figure 5. An alternative representation of the decompositions outcomes is a *system graph*, shown in Figure 6. The system graph shows the different types of subsystems and how they logically link together. Failure causality can be traced across the entire system due to the explicit, logical links between the subsystems.

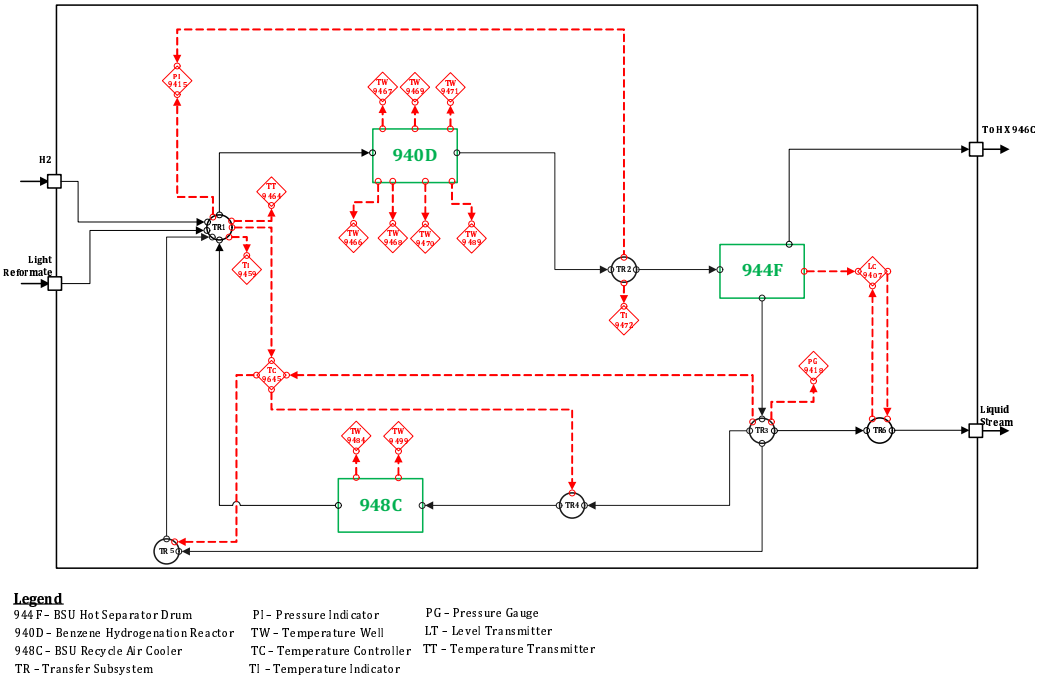

Figure 6: BSU system graph

5.3. Reactor 940D BLHAZID outcomes

The BSU hydrogenation reactor converts benzene to cyclohexane by reacting it with hydrogen over a catalyst. The initial set of c-vars with their appropriate guidewords along with relevant knowledge sources for reactor 940D are shown in Table 4. The failure modes of reactor 940D are shown in Table 5.

A selection of the BLHAZID analysis outcomes of reactor 940D are contained in Figures 7 and 8. Figure 7 shows the causes and implications of <high><temperature>, presented in the *pair* view, where the implications

Table 4: Reactor 940D: characterizing variables and guidewords

Knowledge source	**Characterizing variable**	**Guidewords**
Measurements	Temperature	high, low
Capability set	Flow	high, low, no
Processes	Benzene concentration	high
Processes	H_2 concentration	high, low, no
Capability set	Extent of reaction	low
Processes	Hydrogenolysis extent of reaction	high
Design documents	Catalyst activity	low
Design documents	Vapour-liquid ratio	high, low
Design documents	Impurity concentration	high
Design documents	Exotherm	high, low
Capability set	Pressure	high, low
Failures passed to subsystem	Pressure drop	high, low

Table 5: Failure modes of hydrogenation reactor 940D

• external leak	• inlet partial blockage
• rupture	• outlet partial blockage
• inlet blockage	• catalyst bed partial blockage
• outlet blockage	• maldistribution
• catalyst bed blockage	

are not connected to a pair (cause, deviation). This is in contrast with the outcomes presented in Figure 8, which show the causes and implications of <hold><mass> presented in the *triplet* view. Figure 8 is an excerpt of the full study.

Figure 9 shows the causes and implications of reactor 940D's failures modes. Failure modes are presented in the pair view, as discussed in Subsection 3.3.

5.4. Causal graph representation

Useful causal information about how failures may be caused and propagate can be difficult to ascertain from observing the outcomes in tabular form, as shown in Figures 7 to 9. A visual representation of the outcomes

Subsystem	Causes	Deviation	Implications
940D: Reactor	High inlet temperature from TR1 High Hydrogenolysis extent of reaction High Benzene concentration Low flow 940D maldistribution High air impurity concentration	High temperature	High Hydrogenolysis extent of reaction High exotherm High outlet temperature to TI9466 High outlet temperature to TR2 940D external leak 940D rupture High vapour-liquid ratio

Figure 7: Pair view of the BLHAZID outcomes of deviation <high><temperature> in Reactor 940D

Subsystem	Causes	Deviation	Implications
940D: Reactor	High inlet temperature from TR1	High temperature	High outlet temperature to TR2
940D: Reactor	High Benzene concentration	High temperature	High outlet temperature to TR2
940D: Reactor	940D maldistribution	High temperature	940D external leak
940D: Reactor	High inlet temperature from TR1	High temperature	940D external leak
940D: Reactor	High Benzene concentration	High temperature	940D external leak
940D: Reactor	High inlet temperature from TR1	High temperature	940D rupture
940D: Reactor	High Benzene concentration	High temperature	940D rupture
940D: Reactor	940D maldistribution	High temperature	940D rupture
940D: Reactor	High air impurity concentration	High temperature	High outlet temperature to TR2
940D: Reactor	High air impurity concentration	High temperature	940D external leak
940D: Reactor	High air impurity concentration	High temperature	940D rupture
940D: Reactor	High inlet temperature from TR1	High temperature	High vapour-liquid ratio
940D: Reactor	High Benzene concentration	High temperature	High vapour-liquid ratio
940D: Reactor	High air impurity concentration	High temperature	High vapour-liquid ratio
940D: Reactor	High inlet temperature from TR1	High temperature	High outlet temperature to TI9466
940D: Reactor	High Benzene concentration	High temperature	High outlet temperature to TI9466
940D: Reactor	940D maldistribution	High temperature	High outlet temperature to TI9466
940D: Reactor	High air impurity concentration	High temperature	High outlet temperature to TI9466
940D: Reactor	High inlet temperature from TR1	High temperature	High exotherm
940D: Reactor	High inlet temperature from TR1	High temperature	High Hydrogenolysis extent of reaction
940D: Reactor	High Hydrogenolysis extent of reaction	High temperature	High outlet temperature to TR2
940D: Reactor	High Hydrogenolysis extent of reaction	High temperature	940D external leak
940D: Reactor	High Hydrogenolysis extent of reaction	High temperature	940D rupture
940D: Reactor	High Hydrogenolysis extent of reaction	High temperature	High vapour-liquid ratio
940D: Reactor	High Hydrogenolysis extent of reaction	High temperature	High outlet temperature to TI9466
940D: Reactor	High Hydrogenolysis extent of reaction	High temperature	High exotherm

Figure 8: Triplet view of the BLHAZID outcomes of deviation <high><temperature> in Reactor 940D

in the form of causal graphs is more useful, as discussed in Subsection 4.1, where failure propagation pathways can be easily perceived and hence more effectively used for supporting process diagnosis.

An excerpt from the full causal graph of the causes of <low><extent of reaction> in Subsystem *940D* is shown in Figure 10. In Figure 10, the pentagon is the failure of interest, ellipses are functional failures, rectangles are component failures and diamonds are 'unfolded nodes' where the causal pathways are not currently visible.

Subsystem	Causes	Failure mode	Implications
940D: Reactor	Failure of inlet PEOPLE from environment	940D maldistribution	Low extent of reaction High exotherm High temperature High flow Low pressure drop
940D: Reactor	High exotherm High corrosion products impurity concentration High temperature High pressure	940D external leak	Low pressure drop Low flow Low pressure High outlet flow to env
940D: Reactor	High exotherm High temperature High pressure	940D rupture	No flow Low pressure drop Low pressure High outlet flow to env
940D: Reactor		940D catalyst bed blockage	High pressure High pressure drop No flow
940D: Reactor		940D outlet blockage	High pressure High pressure drop No flow
940D: Reactor		940D inlet blockage	High pressure drop Low pressure No flow
940D: Reactor		940D inlet partial blockage	High pressure drop Low flow
940D: Reactor		940D catalyst bed partial blockage	Low flow High pressure drop
940D: Reactor		940D outlet partial blockage	High pressure drop Low flow

Figure 9: Failure modes, their causes and implications in Reactor 940D

Causal graphs can also be generated across multiple subsystems. Figure 11 shows how the implications of <low><extent of reaction> in reactor 940D flow into subsystem *TR2*. The graph in Figure 11 is again an excerpt of the full causal graph to show that failure causality can be traced across subsystem boundaries, allowing downstream consequences to be more easily linked to upstream root causes.

6. Discussion

A number of aspects of the BLHAZID methodology are discussed in the following paragraphs.

Decomposition process. The decomposition process is a procedure that is a mixture between flexibility and rigidity and is more advantageous than a procedure that is completely flexible or completely rigid. Flexibility is built into the decomposition procedure, appearing in the review steps. If a different subsystem structure seems more advantageous to the analysis team then during the review stages changes can be made, as in the case with separating subsystems *TR4* and *TR5* from *TR3*. Subsystems *TR4* and *TR5* were separated so that a clear distinction can be made between the

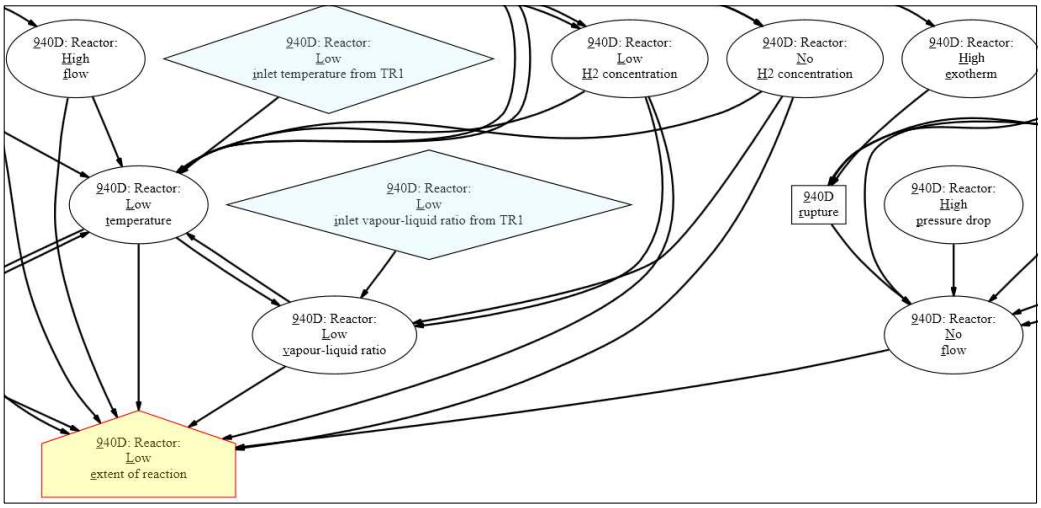

Figure 10: Causal graph excerpt: causes of <low><extent of reaction>

heat exchanger 948C feed and bypass. The rigidity of the decomposition steps give clear guidance on forming a consistent system graph that is free from ambiguity and therefore can be represented in software systems for subsequent applications such as fault diagnosis. This also means that the procedure can be semi-automated since the steps are detailed and clear.

As well as this, analysing smaller subsystems is advantageous as compared to large subsystems. This is because dealing with smaller subsystems can better support analysis teams by reducing the complexity associated with analysing larger subsystems with many components. This is because fewer components have to be considered for each failure scenario, which reduces the amount of interactions between the stream and the components. Therefore far fewer causal interactions need to be examined, allowing teams to operate in a more focussed way on the subsystem under analysis. The trade-off is that while smaller subsystems confer simpler analyses there are more subsystems to examine for a given system, meaning that the overall time for the analysis may not be reduced. Thus the final decision as to how many subsystems is useful will depend on the nature of the system, the end-use for the BLHAZID outcomes and how teams want to analyse the system.

Overlapping and complementary nature of the BLHAZID. One of the main issues with the BLHAZID approach is that there is often a overlap in the outcomes between the function-driven and component-driven parts of the anal-

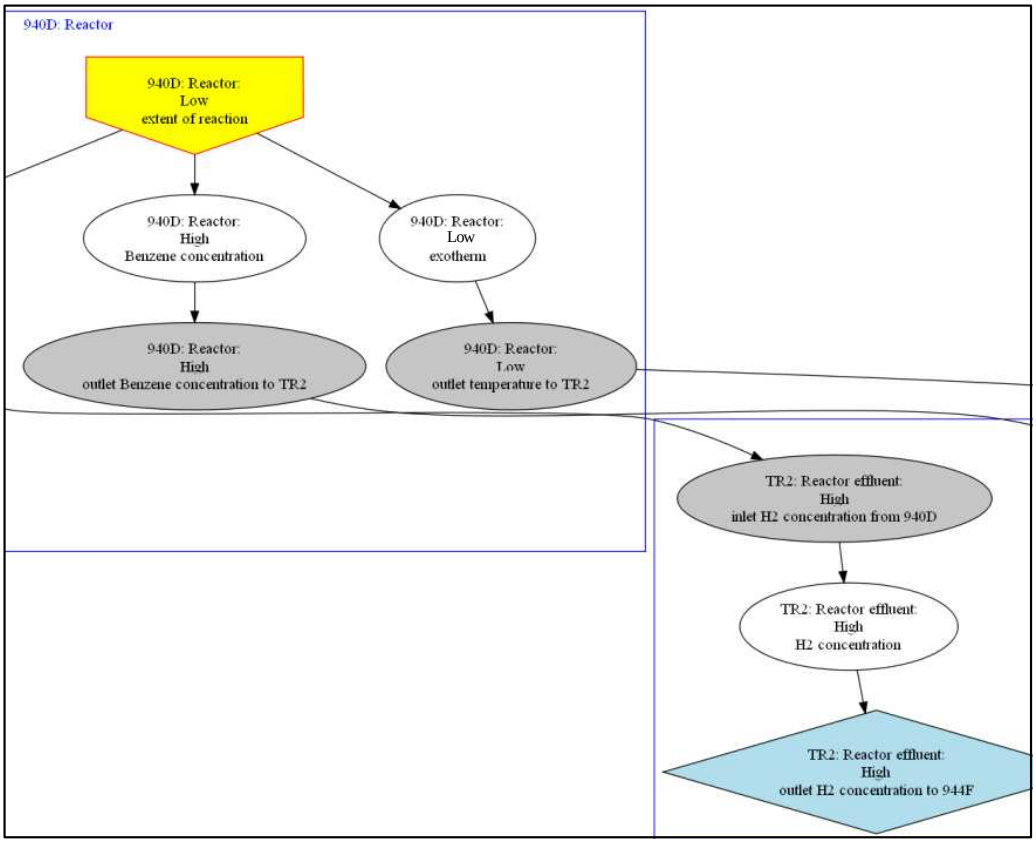

Figure 11: Causal graph excerpt: implications of <low><extent of reaction>

ysis. For example, the (cause, deviation) pair (<940D><maldistribution>, <high><temperature>) in Figure 8 is repeated as the pair (failure mode, failure mode implication) in Figure 9. This overlap situation highlights the importance of supporting BLHAZID analysis with computer aided methods, such as semi-automated instantiation of large amounts of 'routine' causal knowledge. For example, the triplet (<high><inlet temperature>, <high> <temperature>, <high><outlet temperature>) in Figure 8 could be semi-automatically generated and instantiated in the outcomes since it is a causal pathway that is relevant in many situations. Any semi-automatically generated results would need to be checked by process experts. This a similar philosophy to HAZOPExpert (Vaidhyanathan et al., 1996) of using general process knowledge stored in databases to support the analysis. The software tool that is in development for supporting the BLHAZID methodology is in-

tended to be able to perform the semi-automated analysis, where the routine BLHAZID outcomes are generated by the tool. Analysis teams still should examine the system for failures that are not routine or difficult to identify.

Benefits of structured language and subsystem representation – causal graphs. The causal graph representation is a powerful way to reuse the knowledge contained in the BLHAZID outcomes. Long causal pathways can be visualized and easily presented in the form of casual graphs. These graphs are useful for seeing how failures propagate throughout process systems to support how preventative or mitigative actions are planned and implemented. Many other aspects of process system operations can benefit from the use of these causal graphs, including operator training, fault diagnosis, design reviews, fault and event tree construction, auditing and hazard updates to satisfy major hazard facility (MHF) requirements. A previous paper by Németh et al. (2011) on the generation of these causal graphs from BLHAZID outcomes has recently appeared.

Flexibility and rigidity of the workflow. The BLHAZID workflow is a combination of flexible steps and rigid steps. The rigid parts of the workflow are related to, but not limited to:

- the focus on a single subsystem at a time
- examining of c-vars at the subsystem level
- examining of failure modes at the component level
- specifying particular *types* of causes and implications and how these types arise from the component-stream model

The flexible parts of the workflow are related to, but not limited to:

- depth of causal detail between c-vars
- place where new information can be included; for example, newly identified c-vars can be analysed before the component-driven analysis has commenced
- order of analysis of subsystems/c-vars/failure modes

The generation of this initial set of c-vars is non-trivial and extensive discussion can take place to identify them. During this discussion it is important to understand how the variables interact so that a useful set of c-vars is identified that sufficiently captures the function of the reactor. This process of identifying c-vars for complex units effectively constitutes a simple modelling exercise for the analysis team. This is because the interrelationships between the variables are discussed at length.

The reactor 940D has many interacting c-vars and the guided discussion for identifying them is often required to be quite detailed. This discussion is not only important for establishing the initial set of c-vars but also for understanding the function of the reactor at a deep level. The process of deciding which variables should be considered as c-vars involves discussing the causality between the variables, which directly supports the subsequent elicitation of causes and implications. The creative and flexible aspect of this discussion is important for identifying c-vars of subsystems with many interacting variables.

The ability of the BLHAZID analysis to capture failure situations that have many interacting variables in a structured way is an advantage over techniques such as HAZOP, where it may be difficult to express these situations in a clear manner. HAZOP and similar techniques have flexibility built into their workflows, which confers certain benefits but is not as effective at describing detailed failure causality.

This characteristic of being flexible in parts and rigid in parts places the BLHAZID as a hybrid between HAZOP and FMEA, since HAZOP tends to be flexible while FMEA is more rigid. The BLHAZID methodology exhibits flexible characteristics akin to HAZOP and rigid aspects similar to FMEA: another way in which it is *blended*. For example, the guided discussion for identifying c-vars is similar to the creative discussions that occur during a HAZOP analysis. In contrast the detailed steps use to elicit causes and implications are similar to the rigorous approach of the FMEA analysis. This blending of creativity and rigour allows the methodology to be flexible when required whilst still generating outcomes that are structured and can be reused.

Differences between BLHAZID, HAZOP, FMEA and other computer-aided HAZID methods. The main purpose of the BLHAZID methodology is to generate causal knowledge of failures in process systems so that hazard information can be extracted, in the form of consequences, and the knowledge

be available for use in a number of subsequent applications, such as fault diagnosis. This purpose overlaps with but is different from the purposes of HAZOP, FMEA and the many examples of computer aided HAZID approaches. HAZOP specifically identifies *hazards* and is less concerned with establishing detailed causal pathways. HAZOP is effective because it gives users a flexible approach for identifying complex hazard scenario's across multiple sections of plant. An FMEA can generate detailed causal pathways, where this knowledge is focussed on equipment failures at different indenture levels. This is very useful for maintenance, but not so strong on the integration between equipment and system-wide process issues.

Most of the computer aided approaches emulate traditional HAZID methods, utilizing knowledge management strategies and tools such as ontologies, quantitative or qualitative process models. These approaches produced very detailed causal knowledge of failures while reducing the heavy burden on analysis teams. However, the BLHAZID method facilitates a integrated perspective of considering both functional issues along with component focussed issues. Since the BLHAZID is based on a highly structured conceptual framework, the FSF, both the concepts and workflow of a function-driven and component-driven approach are blended effectively. This allows detailed causal links to be formed between failure events in components and streams seamlessly, greatly supporting the generation of outcomes with a high coverage of hazards.

7. Conclusion

A novel blended hazard identification method has been developed which blends two fundamentally different HAZID approaches, namely the function-driven and component-driven approaches, in order to generate outcomes that have a high coverage of hazards, describe rich causal knowledge and are structured so they can be effectively reused. The BLHAZID is based on a formal conceptual basis called the Functional Systems Framework (FSF), which was also used as an underlying process system model upon which BLHAZID analysis is performed. A structured language has been developed in order to express the knowledge contained in the method outcomes so that this knowledge can be effectively reused for a number of applications, including fault diagnosis. A case study of a benzene saturation unit was presented to show how the BLHAZID methodology operates in practice.

The outcomes contain detailed causal knowledge of failures events and their causes and implications. The depth of detail is dependant on the end-use requirements for the outcomes. For example, for fault tree construction focussed on a particular hazardous scenario a very deep understanding of the relationships between c-vars is required in order to describe the dependencies in the tree. Alternatively, if diagnosis were the main application, a focus on deviations of measured variables and how they are related to component failure modes may be required.

The knowledge generated through the use of the BLHAZID methodology can be applied in a number of ways in the process industries. Operators and plant engineers can use the knowledge of failures, and the causal links established between them, during online diagnosis. Given the right software tools, reasoning can be performed on the BLHAZID outcomes to generate failure propagation pathways, thereby supporting the diagnostic effort. Similarly, the a priori knowledge of component capabilities and failure modes can be utilized during front-end engineering design to show how failures can originate and propagate in designs before they are implemented.

The BLHAZID workflow supports the generation of very detailed outcomes, however the procedure is quite intensive and requires users to be highly skilled in the method and have a deep understanding of the causal dependency between different process variables as well as component failure modes. In light of these issues, computer aided approaches are being developed, which include knowledge extraction from intelligent P&IDs for automated system decomposition (Németh et al., 2009) and automated instantiation of 'routine' causal pathways, characterizing variables and failure modes from *a priori* databases and decomposition results.

8. Acknowledgements

We acknowledge support from the Australian Research Council Linkage Grant LP0776636 and from the Hungarian Research Fund Grant number 83440. We also thank BlueScope Steel (Port Kembla, Australia) and BP Refinery (Bulwer Island, Australia) for their financial and technical support for this work.

References

Batres, R., West, M., Leal, D., Price, D., Masaki, K., Shimada, Y., Fuchino, T., Naka, Y., 2006. An Upper Ontology based on

ISO 15926. Computers & Chemical Engineering 31, 519–534, doi: 10.1016/j.compchemeng.2006.07.004.

Bunge, M., 1977. Ontology 1: The Furniture of the World. Treatise on Basic Philosophy. D. Reidel Publishing Company.

Bunge, M., 1979. Ontology 2: A World of Systems. Treatise on Basic Philosophy. Dordrecht, Boston, Reidel.

Cameron, I. T., Németh, E., Seligmann, B., Hassal, M., Sanderson, P., Hangos, K., Hockings, K., O'Brien, C., 2010. An integrated functional systems approach to improving diagnosis in complex process systems. In: CHEMECA 2010: The 40^{th} Australasian Chemical Engineering Conference. Paper # 233 (10 pages), ISBN: 978-085-825-9713.

Cameron, I. T., Raman, R., 2005. Process Systems Risk Management. Vol. 6 of Process Systems Engineering. Elsevier Academic Press, San Diego, CA.

Cameron, I. T., Seligmann, B., Hangos, K. M., Lakner, R., Németh, E., 2007. The P^3 formalism: a basis for improved diagnosis in complex systems. In: CHEMECA 2007: The 37th Australasian Chemical Engineering Conference. On CD, paper # 224 (12 pages).

Cameron, I. T., Seligmann, B., Hangos, K. M., Németh, E., Lakner, R., 2008. A functional systems approach to the development of improved hazard identification for advanced diagnostic systems. In: 18^{th} European Symposium on Computer Aided Process Engineering (ESCAPE 18). Elsevier, Lyon, on CD, paper ID FP_00463.

CCPS, 1992. Guidelines for Hazard Evaluation Procedures, 2nd Edition. Centre for Chemical Process Safety, American Institute for Chemical Engineers, New York.

Dunjó, J., Fthenakis, V., Vílchez, J. A., Arnaldos, J., 2009. Hazard and operability (HAZOP) analysis. A literature review. Journal of Hazardous Materials 173, 19–32, doi: 10.1016/j.jhazmat.2009.08.076.

Fiske, T., 2009. Asm consortiums solution framework creates business value for adopters. Tech. rep., ARC Advisory Group, online. Date accessed 10th October, 2010. URL: http://www.asmconsortium.net/Documents/ASM Consortium Solution Framework Creates Business Value for Adopters.pdf.

Graham, P. R., 2005. *A Hybrid Method for Improved Hazard Identification in Process Systems*. BE(Chemical) thesis, The University of Queensland.

Gruber, T. R., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5 (2), 199–220, doi: 10.1006/knac.1993.1008.

ISO, 2003. ISO 15926-2:2003 Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data model.

ISO, 2004. ISO 15926-1:2004 Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities — Part 1: Overview and fundamental principles.

Jalashgar, A., 1999. Goal-oriented systems modelling: justification of the approach and overview of the methods. Reliability Engineering & System Safety 64, 271–278, doi: 10.1016/S0951-8320(98)00067-2.

Mannan, S. (Ed.), 2005. Lee's Loss Prevention in the Process Industries: Hazard Identification, Assessment and Control, 3rd Edition. Vol. 1. Burlington, MA.

Marquardt, W., Morbach, J., Wiesner, A., Yang, A., 2010. OntoCAPE: A reusable ontology for chemical process engineering. RWTH edition. Springer, Heidelberg, doi: 10.1007/978-3-642-04655-1.

McCoy, S. A., Wakeman, S. J., Larkin, F. D., Jefferson, M. L., Chung, P. W. H., Rushton, A. G., Lees, F. P., Heino, P. M., 1999. HAZID, a computer aid for hazard identification: 1. The STOPHAZ package and the HAZID code: An overview, the issues and the structure. Process Safety and Environmental Protection 77 (6), 317–327, doi: 10.1205/095758299530242.

McCoy, S. A., Zhou, D. F., Chung, P. W. H., 2006. State-based modelling in hazard identification. Applied Intelligence 24 (3), 263–279, doi: 10.1007/s10489-006-8517-4.

Németh, E., Hockings, K., O'Brien, C., Cameron, I. T., 2009. Knowledge representation, extraction and generation for supporting a semi-automatic blended hazard identification method. In: CHEMECA 2009: The 39th Australasian Chemical Engineering Conference. On CD, paper # 227 (10 pages).

Németh, E., Lakner, R., Hangos, K. M., Cameron, I. T., 2007. Prediction-based diagnosis and loss prevention using qualitative multi-scale models. Information Sciences 177 (8), 1916–1930, doi: 10.1016/j.ins.2006.10.009.

Németh, E., Seligmann, B. J., Hockings, K., Oakley, J., O'Brien, C., Hangos, K. M., Cameron, I. T., 2011. Generating cause-implication graphs for process systems via blended hazard identification methods. In: Pistikopoulos, E. N., Georgiadis, M. C., C., K. A. (Eds.), 21^{st} European Symposium on Computer Aided Process Engineering (ESCAPE 21). Vol. 29. Pages 1070–1074.

OREDA, 2009. OREDA: Offshore Reliability Data - Volume 1 Topside Equipment, 5th Edition. Vol. 1. Trondheim, OREDA participants: BP Exploration Operating Company Ltd, ConocoPhillips Skandinavia AS, Eni S.p.A. Exploration & Production Division, ExxonMobil Production Company, Gassco, Shell Global Solutions UK, Statoil ASA, Total S.A.

Pasman, H. J., 2009. Learning from the past and knowledge management: Are we making progress? Journal of Loss Prevention in the Process Industries 22 (6), 672–679, doi: 10.1016/j.jlp.2008.07.010.

Rasmussen, B., Petersen, K. E., 1999. Plant functional modelling as a basis for assessing the impact of management on plant safety. Reliability Engineering & System Safety 64 (2), 201–207, doi: 10.1016/s0951-8320(98)00063-5.

Rausand, M., Oien, K., 1996. The basic concepts of failure analysis. Reliability Engineering & System Safety 53, 73–83, doi: 10.1016/0951-8320(96)00010-5.

Rossing, N. L., Lind, M., Jensen, N., Jørgensen, S. B., 2010. A functional HAZOP methodology. Computers & Chemical Engineering 34 (2), 244–253, doi: 10.1016/j.compchemeng.2009.06.028.

SA, 2003. AS IEC 61882-2003: Hazard and operability studies (HAZOP studies) – Application guide. Standards Australia, aS61882.

Schüller, J. C. H., Brinkman, J. L., van Gestel, P. J., van Otterloo, R. W., 1997. Methods for determining and processing probabilities. Tech. Rep. CPR-12E, KEMA Nederland B.V.

Seligmann, B. J., Németh, E., Hockings, K., McDonald, I., Lee, J., Hangos, K. M., Cameron, I. T., 2010. A structured, blended hazard identification framework for advanced process diagnosis. In: 13^{th} International Symposium on Loss Prevention and Safety Promotion in the Process Industry (Loss Prevention 2010). Pages 193–200.

Seligmann, B. J., Németh, E., Hockings, K., O'Brien, C., Cameron, I. T., 2009. A blended hazard identification approach to support intelligent diagnosis in process systems. In: CHEMECA 2009: The 39th Australasian Chemical Engineering Conference. On CD, paper # 228 (10 pages).

Trammell, S. R., Davis, B. J., 2002. Using a modified HAZOP/FMEA methodology for managing process risk. Semiconductor Environmental Safety & Health Association 15 (3), 32–39.

Vaidhyanathan, R., Venkatasubramanian, V., Dyke, F., 1996. HAZOPExpert: An expert system for automating HAZOP analysis. Process Safety Progress 15 (2), 80–88, doi: 10.1002/prs.680150206.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S. N., 2003a. A review of process fault detection and diagnosis Part I: Quantitative model-based methods. Computers & Chemical Engineering 27 (3), 293–311, doi: 10.1016/S0098-1354(02)00160-6.

Venkatasubramanian, V., Vaidhyanathan, R., 1994. A knowledge-based framework for automating hazop analysis. AICHE Journal 40 (3), 496–505, doi: 10.1002/aic.690400311.

Venkatasubramanian, V., Zhao, J. S., Viswanathan, S., 2000. Intelligent systems for HAZOP analysis of complex process plants. Computers & Chemical Engineering 24 (9-10), 2291–2302, doi: 10.1016/S0098-1354(00)00573-1.

von Bertalanffy, L., 1968. General System Theory: Foundations, Development, Applications, revised edition Edition. Braziller, New York.