INDUCTIVE AND DEDUCTIVE REASONING: THE CASE OF IDENTIFYING POTENTIAL HAZARDS IN CHEMICAL PROCESSES

Christopher Nagel[1] and George Stephanopoulos

Laboratory for Intelligent Systems in Process Engineering
Department of Chemical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

All reasoning carried out by computers is *deductive*; i.e., any software system has all the necessary data, stored in various forms in a database,

[1]Present address: Molten Metal Technology, Inc., Waltham, Massachusetts.

ADVANCES IN CHEMICAL ENGINEERING, VOL. 21

and possesses all the necessary algorithms to operate on the set of data and *deduce* some results. Many researchers in the area of cognitive psychology make similar claims on the reasoning mechanisms of the human beings. The fact, though, remains that both humans and machines can use very simple "algorithms" on a small set of data and produce results, which could not have been visible by the "naked eye" of direct reasoning. In such cases, we tend to talk about the *inductive* capabilities of either of the two. These ideas are nowhere more prominent than in the area of *hazards identification and analysis*. One often hears, "if I knew that the conversion of A to B could be catalyzed by the presence of C then I would have forseen the last disaster, and have done something about it," with the speaker converting a problem of *inductive* identification (i.e., induce the possibility of a hazard from the list of chemicals) into an issue of deductive statement. In this chapter we try to demonstrate that the identification of hazards is essentially an interplay between inductive and deductive reasoning. Through inductive reasoning we attempt to generate all potential hazardous top-level events, which can be justified by the presence of a set of chemicals. We call the reasoning inductive because it has the potential to generate specific knowledge that was not "visible" ahead of time. Once the potentially harmful top-level events have been identified, deductive reasoning attempts to "walk" through the processing scheme and its unit operations and their design or operating characteristics (assumptions, or decisions), and generate the preconditions, which would enable the occurrence of a specific top-level event. The inductive reasoning procedures operate on a set of chemicals and create in an *exhaustive*, *bottom–up* manner many alternative reaction-pathways, some of which could lead to a hazard, e.g., release of large amounts of energy over a short period of time. On the other hand, the deductive reasoning procedures are *goal-directed* and operate in a *top–down* manner. In this chapter we will develop the detailed framework for the implementation of these ideas, which among other benefits offer the following advantages: (1) formalize the hazards identification problem and unify the methodological approaches at any stage of the design activities and (2) systematize the generation and evaluation of mechanisms for the prevention of hazards, or containment of their effects.

I. Introduction

When a process is transferred from the laboratory to the pilot or/ and commercial scale, a variety of hazards may appear that had earlier been

well controlled under the relatively small scale of the discovery effort. Throughout the period of a process's development, two factors that may introduce new hazards and must be examined continuously are *change* and *scale-up* (Brannegan, 1985). Change complicates hazards evaluation by introducing new components that are associated with hazards that may be unknown. Scale-up, particularly initial scale-up, can generate significant potential hazards by escalating the magnitude of effects, initially thought to be benign. Since changes often occur throughout the life of the plant, the need to identify hazards as early as possible in the development stages does not imply that hazard identification ends when the design specifications have been approved. In fact, approval of a design means only "At the time of the study, the study team believes that, provided that the plant is constructed and operated in accordance with their recommendations, the plant will be acceptably safe" (Lowe and Solomon, 1983). The first uncontrolled change during construction, or the first unapproved modification during operation, negates this approval. Consequently, hazard identification is a continuing concern and a permanent ingredient of safe operations and should be applied, sometimes in a very simple form, to control any changes from the original intentions of the designers. Several approaches have been presented in the past decade to systematize hazard identification and hazard analysis, but procedural robustness is often constrained by the quality of information available and the expertise of the individuals involved.

No one doubts the importance of hazard identification, in advance of an unwanted event. However, the quality of the risk analysis results can be no better than the extent to which hazards are recognized in the first place. Furthermore, the analysis is no better than the analyst's understanding of the plant's design and its operations. Decisions about safety are made continuously. These decisions are made in light of all the uncertainties and are based on the understanding of the characteristics of the facility and the substances involved. Formal analytical processes may or may not be involved in the decision process. Studies recently indicated that design errors—a design error is deemed to have been committed when the design is changed after an incident—were rarely revealed before startup and accounted for 25% of all accidents (Haastrup, 1983). India's experience with design error is closer to 40%, and it has been suggested that if the definition is broadened to include management and organizational aspects of process design and engineering, design errors would account for nearly 90% of the recorded incidents (Batstone, 1987). Moreover, the percentage of precursors (leading to a hazardous incident), perceived to be known at the time of the incident, varies. It depends on the perceptions that an individual formed in his or/her particular

capacity. Our survey suggests that plant personnel believe that 90–100% of the precursors leading to a hazard are known at the time of the incident, hazard specialists believe that 40–60% of those precursors are known at the time of the incident, and insurance analysts believe that only 20–30% of the precursors initiating or propagating the preconditions to a hazard are known at the time of the incident. Such data are exacerbating and they suggest, to some, that improved hazard identification methods are unnecessary, perhaps even unwanted. Yet, incidents continue to occur.

A. Predictive Hazard Analysis

The basic objective of hazard analysis is to identify and assess potentially hazardous situations, and their possible consequences and associated risk, in order to provide a rational basis for determining where risk reduction measures are needed. Hazard identification always has been an integral part of design and operational practice. However, it is to a large degree still an informal process depending on the experience of those directly involved.

Structured hazard identification methods can be classified roughly in two groups: (1) *comparative* methods that rely on systematic comparison of the process design against some recognized code or standard and (2) *fundamental* methods that can be applied in almost any situation (Boykin and Kazarians, 1987; Ozog and Bendixen, 1987). Individual experience is the essential ingredient of hazard identification for the first group of methods. However, such an approach requires that individual experience be collected, organized, recorded and standardized and become accessible information to those designing the equipment (e.g., through national and international codes and standards). Such codes and practices provide minimum standards against which deviations from safe practice can be identified and appraised. An important feature of these methods is that the experience gained through many years is incorporated in the company's practices and therefore available for use at all stages of design and construction. For new processes, the hazard identification procedure is strongly dependent upon information obtained a priori, and derived from the efforts of the research and development engineers and the hazard identification team members. This requires that hazard identification methods must be directed toward stimulating the team members to apply their own experience of safe and unsafe as the standard by which to appraise the design, mainly by raising a series of "what if" questions. Fundamental methods of hazard identification are aimed at two outcomes: to identify serious incidents that may result in personal injury or financial

loss [known as "top-level events" or (TLES)]; and to identify the underlying root causes leading to top-level events.

In general, methods that identify actions that eliminate, avoid, or reduce the potential hazard of a particular design are referred to as *intrinsic*. This approach evolves the design technology toward inherently safer configurations through the use of codes, guidelines, and checklists. Intrinsic methodologies can be effective at the stages where the process scheme is conceived and the process flow diagram is developed. At these stages equipment changes are easily made without adversely affecting construction costs and schedules (see Fig. 1), but they lack formality and afford no means for complete and consistent hazard identification. Alternatively, methods that identify actions that reduce the likelihood of a hazard through control and safety devices are referred to as *extrinsic*. These methods tend to control an identified hazard, or the conditions leading to the hazardous state and often generate solutions that increase plant complexity and operational costs. The more formal methods, such as HazOp, FMECA (Failure Modes, Effects, and Criticality Analysis), fault trees, event trees, and cause–consequence analysis, require information that is often sparse in the early design stages (see Fig. 1) and the certainty of which may be indeterminate.

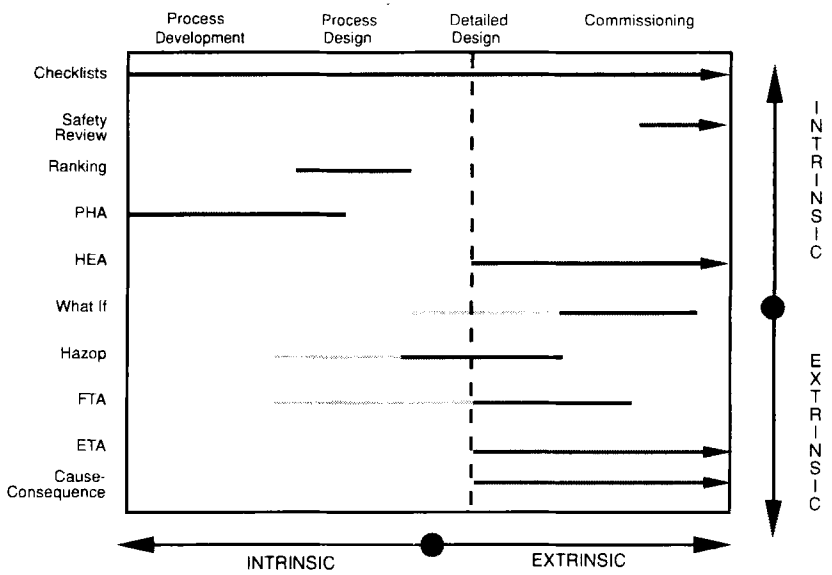

FIG. 1. Applicability of various hazards identification and analysis methodologies during various phases of the design process.

Lees (1980) stated that, "The safety of the plant is determined primarily by the quality of the basic design concept rather than by the addition of special safety features." This point cannot be overemphasized. The degree to which it is economic to eliminate, as opposed to control, a hazard is very dependent on when the hazard is recognized. By the time the design has reached the stage of sufficient documentation to allow a detailed hazard identification, the flexibility to eliminate hazards entirely is very reduced. If hazard/error detection is to be shifted to predesign review, a less encumbered approach must be developed and integrated into the design process.

B. Incompleteness of Conventional Hazard Analysis Methodologies

The identification of potential hazards and the evaluation of their effects should be a continuing process from the conception of the processing scheme to plant shutdown and decommissioning. However, because conventional methods are limited by their scope and useful life span, it is difficult to integrate hazard analysis and evaluation into the design process. More importantly, all conventional methods are incomplete. Their ineffectiveness is inherent in their methodological approach and their limitations in capturing and utilizing all forms of available knowledge. These weaknesses result in two principal deficiencies: (1) the strength of analysis is dependent on the a priori identification of hazards and/or events leading to these hazards, and (2) they are unable to offer a reliable estimate of the quality of the design technology. The same weaknesses prevent the available methods from reaching an adequate balance among the following non-commensurable objectives: (a) early identification of hazards to avoid costly redesign or construction modifications, (b) postponement of evaluation to await more detail, and (c) avoidance of costly duplication of effort. Currently, there is no single solution to the problem. Multiple methods are used over the extended time period of the design process (development, construction, operation). Each of these methods suffers from particular deficiencies, which are born out from the specific design context that they were to service and the character of the approach that they have adopted.

Intrinsic methods, for example, although they tend to increase the quality of the process design, must generally be employed in the early engineering stages. The window of opportunity for their application is very brief since the incorporation of intrinsic safety features at a late stage in the design will usually require major design changes with adverse consequences on the cost and the schedule for the commissioning of the plant's

operation. Moreover, they do not provide a creative search for new hazards when experience is lacking, nor can they provide quantification as to the quality of a particular design. Heuristics have been proposed to fortify intrinsic methods (Dale, 1987; Kletz, 1985). They are ad hoc pieces of knowledge, offering no metric of sound origin to discriminate among alternative design concepts. However, they can be suggestive as to a design's quality provided they are intelligently applied; blind usage often leads to unforeseen difficulties.

Extrinsic methods attempt to quantify the implications of a hazard's occurrence by beginning with a detailed design. Unfortunately, by the time a detailed design is available, it is often too late to avoid hazards. The control of the hazards through external, safety devices is the only economic alternative. In principle, controlling the effect of hazards leads to an acceptable solution, although there is no assurance that the envisioned control scheme will effectively mitigate all eventual outcomes. In fact, the fundamental flaw in any method that is based on controlling hazards lies in the assumption that they have the ability to both identify accurately and pinpoint precisely the location of a future hazard. Virtually all experience suggests that the contrary is true. Indeed, wherever we suspect the possibility for the initiation of a hazard, we take added precautions to add control and safety devices in order to reinforce the assurance that the hazard will not occur. Consequently, conventional methods are approximations to hazard analysis and assessment. They are incomplete because the set of axioms they use and the deductive methodologies they employ are both incomplete. Furthermore, their performance and effectiveness is fundamentally limited due to their lack of expressive power. Conventional methods avoid the use of models and seek solutions from techniques that do not have firm chemical engineering foundations. What is needed is a formal, unified approach for systematically, automatically, and completely identifying hazards and pathways leading to hazards in design alternatives —at all stages of the design process. But how does one design such a methodology?

C. Premises of Traditional Approaches

The weaknesses of the traditional approaches, as discussed above, are all due to their inherent *lack of representational expressiveness*. This shortcoming can manifest itself in the following ways: (1) a methodology exhibits strength only in a particular design phase, where it captures and uses most of the knowledge available during that phase of the design process; (2) a methodology is incapable of transferring information derived at one phase to another phase of the design process, nor can it reason

about the chemical process and its surrounding environment; (3) discrimination among design alternatives varies depending on the technique employed; and (4) complete identification of hazards cannot be guaranteed and therefore the quality of the analysis cannot be assessed. The fundamental premise of traditional approaches is that "*in hazards analysis there is no unifying theme*"; all hazards are different and that every unreliable design creates hazardous situations in its own way. The stand-alone nature of previous approaches is evidence of this statement.

We have found that the truth is somewhat different: *representation limitations of past efforts have prevented the exploitation of unifying themes.* Moreover, conventional methodologies do not include for use, or produce *explicit* information, such as (1) *underlying assumptions* on operational modes, phase equilibria, fluid mechanical aspects, reaction rates, mechanisms of mass and heat transfer, and selected approaches in estimating the value of thermodynamic properties; (2) *simplifications* made by the analyst to limit the model's validity over a given range of conditions or to underscore the relative importance of various physicochemical phenomena; (3) *missing relationships* including qualitative relationships, order-of-magnitude reasoning, and inequalities; and (4) *scope of the task*, that is, what was the process intended for. Difficulties are encountered in the identification and elucidation of root causes leading to the top-level event because their basis is neither concise nor explicit.

Efforts in engineering science reinforce these observations (Stephanopoulos, 1987). Many tasks are not achievable if representational expressivity isn't sufficiently rich to allow the description of the necessary concepts (Brachman and Levesque, 1985). Concepts must be manipulated directly if powerful reasoning is to be achieved. The success of any advanced computer-aided tool for enhancing the identification of hazards requires: (1) the development of a representational language sufficiently rich to embody advanced scientific concepts and (2) a means for manipulating these concepts and reasoning about them, directly.

D. Overview of Proposed Methodology

In this chapter we will attempt to describe a concise framework for the development of hazards identification and analysis techniques. It is based on the following premises:

1. Every top-level hazardous event has been initiated by a physicochemical reaction.
2. A physicochemical reaction initiating a top-level event is determined solely by the type, chemical reactivity, and physical properties of the materials involved.

3. Every top-level event has been activated by the logical satisfaction of all its physicochemical precursor conditions.
4. The satisfaction of the precursor conditions in premise 3 depends on the structure and the design characteristics of the particular process, as well as its operating conditions.

The implications of these four stipulations are very important and determine the logic, character, and implementation of the hazards' identification and analysis approach discussed in this chapter. Specifically, we can conclude the following:

- Premises 1 and 2 (above) imply that an *inductive* generation of *all* physical and chemical reactions leading to potential releases or generations of uncontainable amounts of energy or mass, or both, per unit time is the essential foundation for the complete identification of all potential hazards. Such task *depends entirely on the chemical behavior of the materials/species involved and not on the structure of a processing scheme, or the operating conditions of the associated plant.*
- Premises 3 and 4 (above) imply that the design or operating modifications of a process leading to the elimination or containment of hazards (identified by the inductive approach) can be generated *deductively* from the knowledge of the plant and its operating conditions.

Observing the above logic, we have organized the material of the subsequent sections in this chapter as follows: Section II provides the essential foundation for the reaction-based identification of hazardous top-level events, while Section III describes the inductive reasoning used to identify these hazardous events from the set of available materials. Both of these two sections draw heavily from the material of the first chapter in this volume and specifically from the LCR, the modeling language developed by the authors to describe chemicals and their chemical reactivity. Section IV outlines the methodological framework for the deductive determination of the causes of hazards, using the available knowledge for the description of the plant's design and operating conditions. A fairly detailed example provides an illustration of the ideas and techniques discussed in Section IV.

II. Reaction-Based Hazards Identification

We present a new approach to hazard analysis and assessment, one that begins with the inductive determination of hazardous chemical or physical

reactions and proceeds deductively through the network of processing steps to identify completely all causes initiating these hazardous reactions. This strategy is *more efficient* in the identification of reaction-based hazards than conventional methodologies. The approach ensures *completeness* and resolves uncertainty of design quality through first-principles-based quantification of the design risk. Furthermore, it enables a computer-aided automation. The methodology is based on two fundamental postulates that formalize and extend Bretherick's observation: "With the exception of releases of toxic or corrosive material, all accidents in storage, handling, or processing of chemicals involve the release of energy at rates too high to be dissipated in the immediate environment without damage." (Bretherick, 1990). These two postulates establish the theoretical framework that enables the design of a system for automatic identification and analysis of hazards in a decidable manner. By unifying the analysis of potential hazards, the framework can utilize the maximum knowledge available at every point in the design process. Regardless of what stage the design is in, a designer's attention can be focused at (1) earlier design stages and their vulnerable areas so that the associated hazards can be eliminated or (2) later stages, where the a priori sequence of events that leads to hazards can be used as an early warning structure (Lees, 1983).

The methodology employs domain-specific modeling languages (see first chapter in this volume) to describe

(a) Chemicals and their reactivity during the *inductive* identification of potential reaction-based hazards (see the modeling language LCR in first chapter in this volume).
(b) Processes during the *deductive* identification of their process-based causes (see MODEL.LA. in first chapter).

A. System Foundations

We now establish the *criterion for completeness* for any hazard identification methodology. [Herein, completeness refers to the ability of a methodology to identify all possible top-level events (Nagel, 1991.)] The methodology presented establishes how we use this criterion to completely and systematically identify hazards in an *efficient* manner.

Theorem. *Any hazard identification methodology must cover all subsets of sources present in a process network in order to guarantee completeness.*

In the discussion that follows we restrict our attention to hazards whose enabling path involves reactions among the chemical species, present in

the process. This restriction excludes, for example, hazards (i.e., top-level events) created by falls, electrical shock, or impact with stationary objects, since there is no finite set of root causes leading to these hazards, thus making impossible the guaranteeing of completeness for hazards of this type. But, the preceding restriction does not exclude hazards initiated by these processes, provided they cause interactions among the chemical species present in the plant. For example, a static charge initiating a chemical reaction or the overheating of a tank's contents leading to its volatilization and subsequent over-pressuring, would be covered. Our approach is based on two fundamental postulates.

Postulate 1. *Hazards can only be created by the interaction of a system through its boundaries, or the altering of internal restraints of the system, such that the system proceeds toward a new equilibrium state.*

This postulate follows naturally from the First and Second Laws of Thermodynamics. Since states at equilibrium have no irreversible interactions, a state change is required to proceed toward equilibrium; these changes can only be brought about by interactions through a system's boundary or the altering of internal constraints. Therefore, a state change must accompany a hazard or the system would be at equilibrium. Although many nonequilibrium states may be transgressed in the development of a hazardous system, we need only consider the equilibrium states to identify potentially hazardous systems. This is possible because state changes can only be brought about by energy and entropy changes. Since these are state functions, an upper limit can be established on the potential of a hazardous state through the analysis of the equilibrium states leading to it. Our goal in inductive identification of hazards focuses on the identification of these states. Since any equilibrium state can be characterized completely by $(n + 2)$ variables, where n represents the masses of the particular chemical species initially charged and 2 represents two independent variable properties (e.g., temperature and pressure), our task is divided into

- The identification of all chemical species sets, which can potentially be present in the process.
- The identification of the independent variable properties that specify the environment of the chemical species.

The latter requires the identification of the processing environment, whereas the former establishes the need to identify the occurrence of all potential reactions. The inductive generation of all potential reactions will be discussed in detail in Section III. Whereas postulate 1 establishes the framework for inductively identifying hazardous states, postulate 2

advances the mechanism by which the pathway of events leading to a hazardous state can be identified deductively.

Postulate 2. *The degree of completeness of the set of internal restraints and exogenous factors specifying a hazardous system and its transformation, determines the degree of completeness with which the pathways leading to a hazardous state can be identified.*

Moreover, postulate 2 suggests that the deductive identification process is restricted by our understanding of mechanisms that allow states comprising the hazardous system to interact. Since any interaction results in an energy or entropy change of these states, the opportunities for preventing a hazard are limited by the incompleteness of our knowledge to determine the restraints and the external variables that specify these states. These in turn are more limited by the quality of the available details in describing the particular processing system (Battelle, 1985). However, there may not be a finite set of root causes leading to a hazardous state, or the identification of a finite set of causes may not be possible. As a consequence, the pathway of precursor events leading to a hazardous state is incomplete by definition. Despite this limitation, the constitutive equations that promote a hazardous state can be used to optimize the inherent safety of a particular design technology. The combined approach presented in this chapter allows the following:

1. Establishment of a systematic and formal methodology for increasing the inherent safety of a design technology.
2. A means for quantitatively assessing the inherent safety of design alternatives.
3. Establishment of a systematic and formal strategy for optimizing the inherent safety of a design technology through the selection of appropriate control points.
4. Guarantees on the correctness and completeness of the pathways leading to top-level events.

B. Modeling Languages and Their Role in Hazards Identification

Any methodology, whether it is applied in an automatic or manual manner, requires a rich representation of the process if hazards are to be effectively identified. This representation must have sufficiently expressive power to allow chains of precursor states or events to be easily identified. To satisfy these requirements, we have chosen to map the process descrip-

tion into a representational form that allows a multi-level, multi-faceted process description (see the modeling language MODEL.LA. in first chapter in this volume).

Research efforts outside the domain of hazards analysis have established that expressive, fully declarative, domain-rich modeling languages are indispensable, if complex integrated systems capable of synthetic tasks are to developed. Furthermore, they have shown that computer-aided systems with advanced reasoning capabilities require the satisfaction of the following three conditions:

- All declarative models should be fully articulated.
- Declarative knowledge should be completely decoupled from the procedural knowledge.
- A modeling language should be rich with domain-specific knowledge and thus allow the user to think about the task at hand in terms that are familiar (Stephanopoulos *et al.*, 1990a, b; Kritikos, 1991).
- For a computer-aided chemical reasoning system, we add a fourth condition. The logical structure of chemistry should be exploited by imbedding natural constraints into the declarative models describing chemicals and chemical reactivity.

The system described herein for the automatic identification of hazards and the pathways leading to them utilizes two modeling languages which satisfy the above requirements. Specifically, we have used

- LCR (language for chemical reasoning; see first chapter in this volume) to describe chemicals and their structure and atomic and molecular properties, as well as chemical reactions and their structure, directionality and contextual character.
- MODEL.LA. (modeling language; see first chapter in this volume) to describe processing systems and their unit operations and behavior, and to encapsulate the design decisions and operating conditions associated with any specific plant.

These languages are not ad hoc constructs but are based on clear formalisms and satisfy the essential premises of grammar, vocabulary, and semantics on which programming languages are founded. Both modeling languages allow multilevel description of processes, reactions, and materials with internal consistency (logical and quantitative) among the various models defining the corresponding objects at various levels of abstraction. Moreover, they support the representation of materials, reactions, and processes at multiple, coexisting contexts, an explicit comparison of the contextually alternative representations, and a systematic backtracking of

decisions and assumptions that led the specific descriptions. Thus, different perspectives of the process, reactions, and materials—such as structural, topological, and physicochemical relationships—can be investigated independently. Both of these languages allow the following operations on processes, reactions, and chemicals:

- Multiple viewing in terms of structure, topology, and behavior
- Disaggregation of abstract descriptions to more detailed ones and aggregation of detailed descriptions to more abstract
- Contextual description of alternative models for the same process
- Controlled flow of information among the models at various levels or in various contexts, and detection of modeling conflicts
- Propagation of qualitative and quantitative knowledge through the defining models of behavior

Utilizing the specialized modeling language, MODEL.LA. (Stephanopoulos et al., 1990a, b; see also first chapter in this volume), we can construct a process description that is suitable for the tasks of hazards analysis or/ and identification. The process description we have employed is an abstraction of the conventional process representation (centered around the topology of a specific network of processing units). It is built on top of the conventional representation and thus provides complete access to the functionality contained in the base representation. This functionality comes with a set of tools that allow us to solve heat and mass balances, identify work interactions, evaluate phase partitioning, etc., and permits the propagation of qualitative and quantitative knowledge through the defining network. By focusing our representation on the thermodynamic state description of the process, we can facilitate the identification of pathways leading to potential hazards in the most efficient manner. The equipment state space (i.e., the process flowsheet representation) can be mapped to a thermodynamic state space, if we know the trajectory of thermodynamic states, comprising a process, combined with the transformations that connect these states. The mapping process that allows us to construct a thermodynamic state-based graph representation of a process flowsheet focuses on the state description of the process. The representation is composed of *streams* and *nodes* in accordance with the following definitions:

- *Streams* are idealized flows that connect nodes, and they have no accumulation or energy losses.
- *Nodes* are identified as points where discrete thermodynamic transformations occur, the result of changes in intensive or extensive variable values.

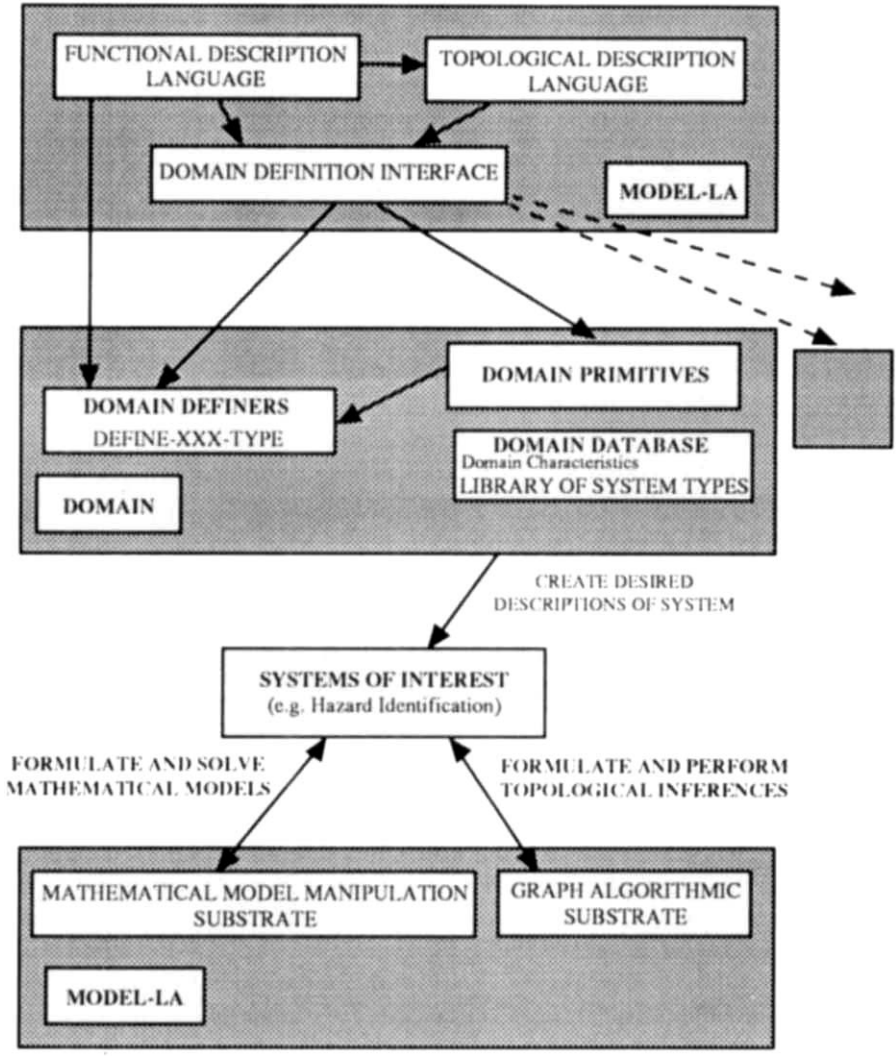

FIG. 2. Overall organization and use of the modeling language MODEL.LA.

These definitions are used to abstract the underlying process representation. As shown in Fig. 2, MODEL.LA. allows us to construct an abstract representation utilizing these definitions for our specific application from a conventional process description. This abstract representation has access to any utility, tool, or component that constitutes the base representation,

identifier:	*unbound*
element-name:	**#<state-1>**
element-type	composite-state
chemical-species-set:	*unbound*
operating-conditions:	**#<op-cond-1>**
boundary-elements:	**(#<flow-port-3> ... #<heat-transfer-port-5>)**
connecting-states:	**(#<state-2> #<state-3>)**
system-description:	**#<node-1>**
system-volume:	*unbound*

FIG. 3. The attributes of the modeling object, ***state***.

e.g.,

- The topology of the network is accessed through the graphical construction of the base representation.
- Mathematical relationships such as phase relations, energy balances, mass balances, and reaction or transport rates can be accessed through the content of the instances of objects of type ***relationships*** (see first chapter in this volume).

Knowledge about the abstract representation and the mapping process are contained in the modeling element, ***state***. Figure 3 identifies several of the attributes that describe a state. Notice that the state is a composite object, i.e., an object comprised of objects. (Herein objects bound to attributes are denoted by #⟨NAME⟩ or will appear in bold.) This allows a multilevel, hierarchical representation of a process to be constructed. For example, the operating conditions that specify a state are accessed through the object ***op-cond-1***, which is the attribute value of *operating-conditions*. Attributes of ***op-cond-1*** include the unit with which it is associated as well as the temperature, pressure, flowrate, and composition associated with that unit and their respective interval ranges.

The chemical species set of a state is the set of chemical constituents that are associated with the system description of that state. The values of the attributes *chemical-species-set*, *operating-conditions*, and *system-volume* provide the $(n + 2)$ independent variable quantities that are necessary to define a thermodynamic state. An important feature of this representation is that each state is described by a vector of *intensive* and *extensive* variables. The intensive vector defines the operational state of the process, while the extensive vector defines the maximum accumulation of mass and energy that can occur. This is bounded by flowrate, reaction rate and physical size of the process equipment. The values of these variables are accessed through the attributes: interval flowrate vector, interval accumu-

lation, and system description. The values bounding an interval are dynamically set. (Intervals are defined as the minimum and maximum allowable value that the variable being described can achieve.)

Boundary elements and connecting states are also associated with the state description. Since a state is described as a system and a system has a boundary, boundary elements identify the vehicle for transforming the current state to a new state. The trajectory of connecting states is contained in the value of the attribute, *connecting-states*. This attribute allows a thermodynamic state representation of a process to be constructed from the individual states that compose it.

Alternatively, an equilibrium state in which there is no net effect on system boundaries can still be transformed to a new state when the internal restraints that specify the system are altered. This information, as well as the mapping that takes from an equipment based representation to a state-based representation, is contained in the value of the attribute *system-description*. An expansion of this state attribute value provides access to the base representation through the description of the modeling object, ***node***. Figure 4 identifies several of the attributes describing the modeling element, ***node***. This detailed description allows us to identify system boundary partitions: flow openings, heat transfer openings, work exchange openings, and mass transfer openings. Additionally, the system boundary type is specified as a thermodynamic boundary. The topological system type and topological connector type provide the links needed to connect the various nodes. Each of these elements has a description that allows us to evaluate its functionality. In addition, application specific handlers are also provided by the underlying modeling language. For example, the boundary element in the attribute, *flow-opening*, may have a description that requires an understanding of vapor-phase, liquid-phase, solid-phase, or multiphase transport. Each of these handlers in turn may access additional methods or handlers to facilitate the evaluation process.

The value of the attribute *system-interior-type* specifies the interior system description of the ***node***. The value of this attribute allows us to map the node representation to the process equipment representation. Notice that it is the attributes of the **node** in combination with the features of the underlying specialized modeling language that afford a multilevel, multifaceted view of the process. Hence, we are able to move independently from the node representation to the equipment representation and pursue, as needed, analysis of momentum, mass, and energy interactions of the defining network. It is clear that using the elements of the modeling language, LCR, we also have access to the representation specifying the chemical constituents that are associated with the state. In the sections

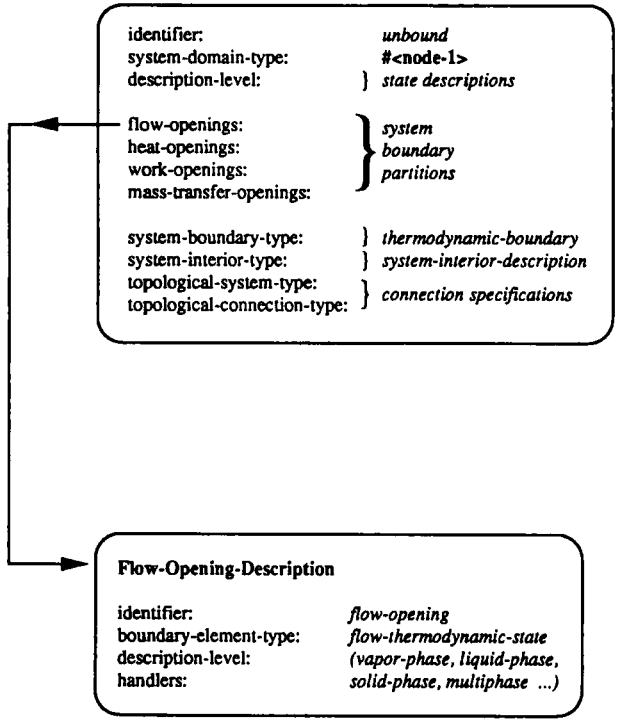

FIG. 4. Select attributes of the modeling objects, ***node***, and ***flow-opening-description***.

that follow, we will show how the information contained in the equipment-based description of a process through the use of MODEL.LA., is accessed by LCR and the chemical species in order to assist in the generation and evaluation of potential reaction alternatives.

An example of a mapping from the equipment representation to the thermodynamic state representation is shown in Fig. 5. It represents an isothermal vertical packed-bed catalytic reactor equipped with temperature and pressure sensors, an explosion vent, and a distributor plate. Notice that the equipment and sensors are not associated with the state representation. They are contained in the base representation and reside in the process description at the equipment level. As discussed earlier, flow, work, heat, and mass interactions are all modeled independently. This allows us to evaluate independently the effect of these processes. Independent evaluation assists in the identification, evaluation, and assessment of event pathways leading to hazardous states.

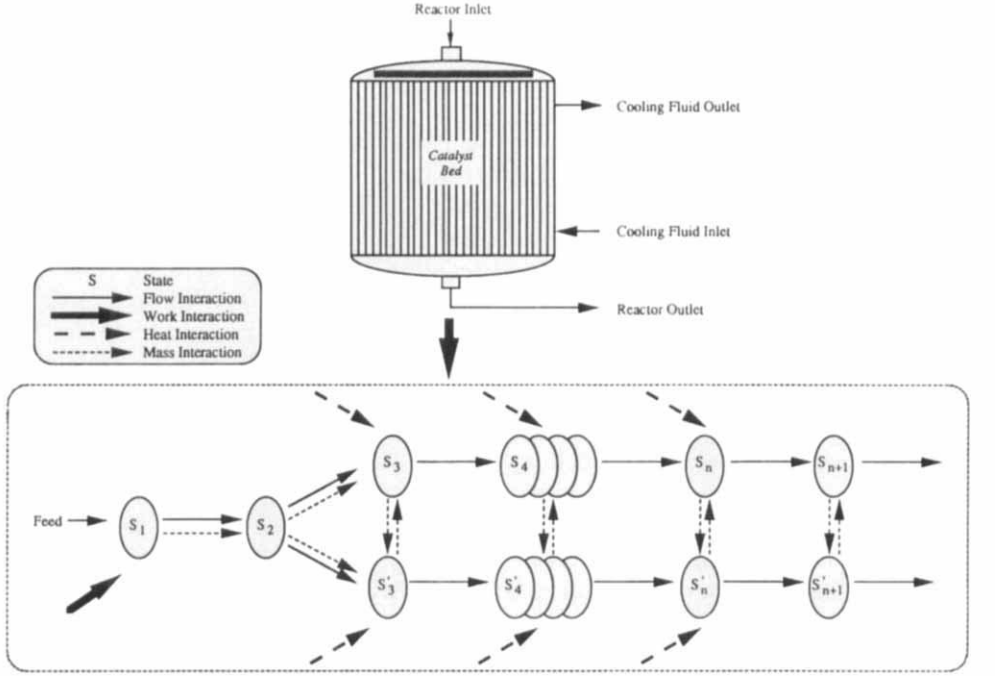

FIG. 5. Mapping of equipment into a thermodynamic state space.

C. GENERATION OF REACTIONS AND EVALUATION OF THERMODYNAMIC STATES

Since new thermodynamic states are created by reactions, it is required that we have a strategy for generating potential reactions between species resident in a particular node of the process description. To describe the reacting species and the reactions they undergo as well as the changes in their molecular structures, we use the language, LCR, which is presented in the first chapter in this volume (for a detailed description of the language, see Nagel, 1991). The language is used extensively in the inductive identification of reaction-based hazards. LCR provides a utility for generating reactions and identifying the resulting pathways, given a set of substrates and a reaction environment. The procedure that provides this utility and the keyword arguments (i.e., arguments with a colon prefix) accepted by this method are given below:

(FIND-ALL-PATHWAYS *:substrates :operators*
:override-environment :initiator-p)

Let us examine the character of the keywords:

- *:substrates*, is the argument that contains the list of chemicals, which are available at a particular process node.
- *:operators*, is the keyword argument that allows the user to specify the types of transformations to be used in order to focus the generation of reaction pathways. To investigate all theoretically feasible pathways subject to encoded preferences the user may supply the keyword argument *:operators* with the value, **K***, a modified composite operator, and allow the generation and evaluation of all pathways having prespecified features ($\Delta G < 10$ kcal/ mol, stereocenters, etc.). Similarly, if there are no preferences, all theoretically feasible reactions can be generated by calling directly FIND-ALL-PATHWAYS with the argument *:operators* having the value $\mathbf{K}_{\text{ab-initio}}$ (Nagel, 1991).
- *:override-environment*, is the argument that lists the operating conditions in which the alternative reaction pathways will be generated. It allows the user (or an automatic procedure) to set alternative operating environments and generate the corresponding alternative reaction pathways.
- *initiator-p*, determines whether an initiator of specific chemical reactions is present or not. It allows the user (or an automatic procedure) to investigate various reaction trajectories without knowing the specifics concerning the mechanisms for initiating reaction pathways.

Using this representation, we can now focus on the identification of the associated thermodynamic states. We assess the likelihood of reactions as well as the inherent instability of each species associated with a node using LCR. The generation of infeasible species is limitated by the following values (i.e., knowledge) embedded in the corresponding attributes;

:override-environment = **reaction-environment**
:operators = **K**, **K***, or $\mathbf{K}_{\text{ab initio}}$.

Let us examine the generation of alternative reactions within the scope of the operating conditions associated with a particular node (or, its corresponding state). For example, suppose that the procedure FIND-ALL-PATHWAYS is applied to a chemical species set (CSS) (i.e., the set of chemicals bound to a specific **state**) composed of three chemicals, **A**, **B**, and **C**, each

Reaction Object	
identifier:	*unbound*
name:	initiation-1
reactants:	(# <Cl_2>)
products:	(#<Cl> #<Cl>)
stoichiometry:	((#<Cl_2> . -1) (#<Cl> . 2))
reaction-environment:	#<reaction-environment-1>
enabling-conditions:	K_f
composing-transformations:	K_t
composing-reactions:	*unbound*
rate-expression:	#<rate-expression-1>
equilibrium-constant:	#<equilibrium-constant-1>
context:	*unbound*

FIG. 6. Description of the modeling object, ***reaction***.

of which is described by an instance of the modeling object, **atom-bond-configuration** (see first chapter). The procedure generates the following set of potentially reactive mixtures:

$$\mathbf{A} \rightarrow ? \qquad \text{(I)}$$

$$\mathbf{B} \rightarrow ? \qquad \text{(II)}$$

$$\mathbf{C} \rightarrow ? \qquad \text{(III)}$$

$$\mathbf{A} + \mathbf{B} \rightarrow ? \qquad \text{(IV)}$$

$$\mathbf{A} + \mathbf{C} \rightarrow ? \qquad \text{(V)}$$

$$\mathbf{B} + \mathbf{C} \rightarrow ? \qquad \text{(VI)}$$

$$\mathbf{A} + \mathbf{B} + \mathbf{C} \rightarrow ? \qquad \text{(VII)}$$

Unique products generated by these reactions are added to the CSS

Pathway Object	
identifier:	*unbound*
name:	pathway-1
reactants:	(# <C_2H_6O> #<Cl_2>)
products:	(#<$C_2H_5Cl_0$>)
stoichiometry:	*unbound*
competing-pathways:	(<pathway-2> #<pathway-3>)
composing-reactions:	(#<initiation-1> #<abstraction-1> #<combination-1> ...)
global-rate-expression:	#<composite-rate-exp-1>
global-equilibrium-constant:	*unbound*

FIG. 7. Description of the modeling object, ***pathway***.

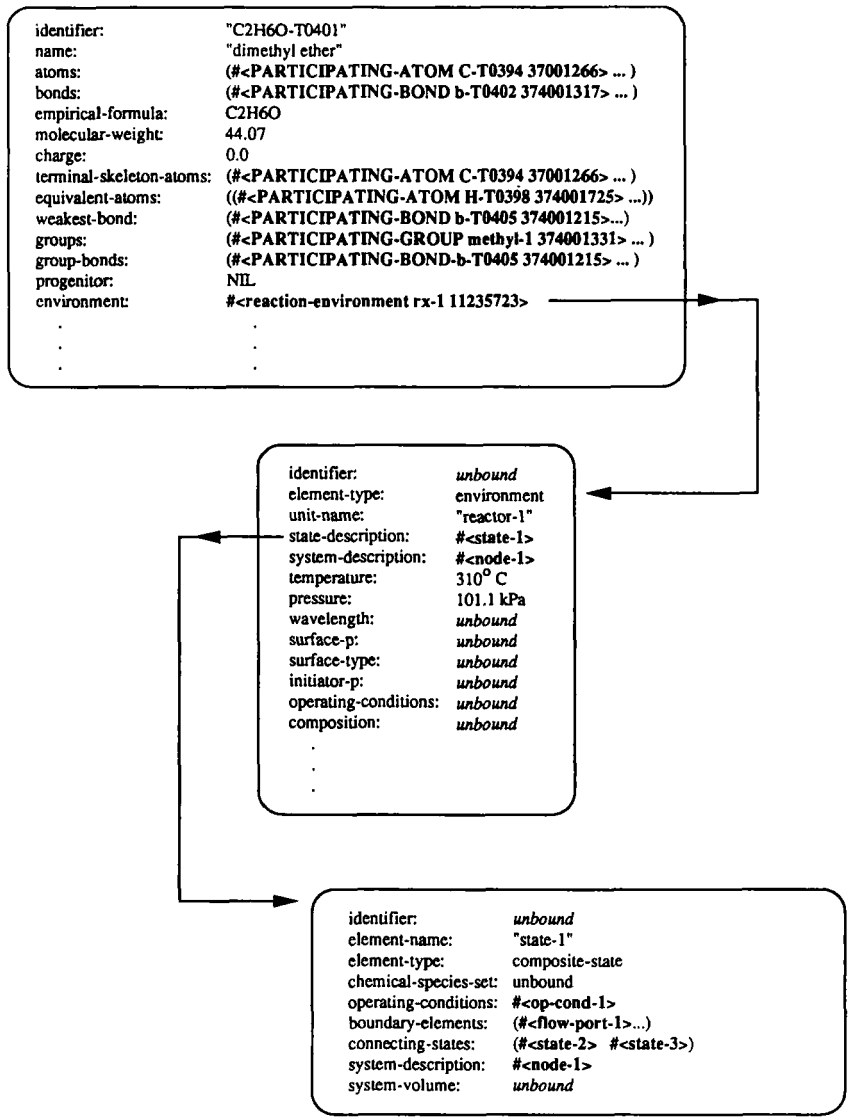

FIG. 8. Accessing process representation from chemical representation.

and the process is repeated. A single call to FIND-ALL-PATHWAYS

(FIND-ALL-PATHWAYS :*substrates* (**ABC**) :*operators* **K***)

generates all possible pathways.

LCR provides modeling objects to contain the information generated by these chemical transformations. For example, the attributes describing the modeling object **reaction** are shown in Fig. 6. These attributes describe not only the reactants and products of the reaction but also contain information on the reaction environment, enabling conditions, composing transformations, reaction stoichiometry, equilibrium constant, rate expression, and identification of competing reactions. Similarly, Fig. 7 shows the attributes describing the object **pathway**, which is made of a network of one-step reactions. Whether any of the above potential reactions, e.g., (I) through (VII), will proceed and develop a hazardous event or not, depends on the value of the operating conditions that characterize the **state** of the corresponding **node**. Therefore, it is imperative that a link be established between the description of the process operations and the **atom-bond-configuration** encoding the information about the available chemicals.

The **atom-bond-configuration** object provides direct access to the **state** description of a **node** through the attribute, *:environment*. Figure 8 shows how the value of the "state-1," describing the conditions in "reactor-1," is transferred to the instance of the **atom-bond-configuration** describing the chemical reactant, "dimethyl ether." Such links allow the transfer of information among different modeling objects in LCR; a mechanism driven by the semantic relationships of LCR (see First chapter).

In a similar manner, the semantic relationships among the various modeling objects of MODEL.LA. allow the transfer of information among these objects. When used together, LCR and MODEL.LA. *allow functional and topological information derived at one point of the process to be accessible from any other point in the process.*

III. Inductive Identification of Reaction-Based Hazards

Using these tools and the representations that are built with them, potential hazards are inductively identified by combining various chemical and physical environments of the process in an attempt to generate new process states. The types and numbers of process states are determined by both the process configuration and the operating conditions. Enabling

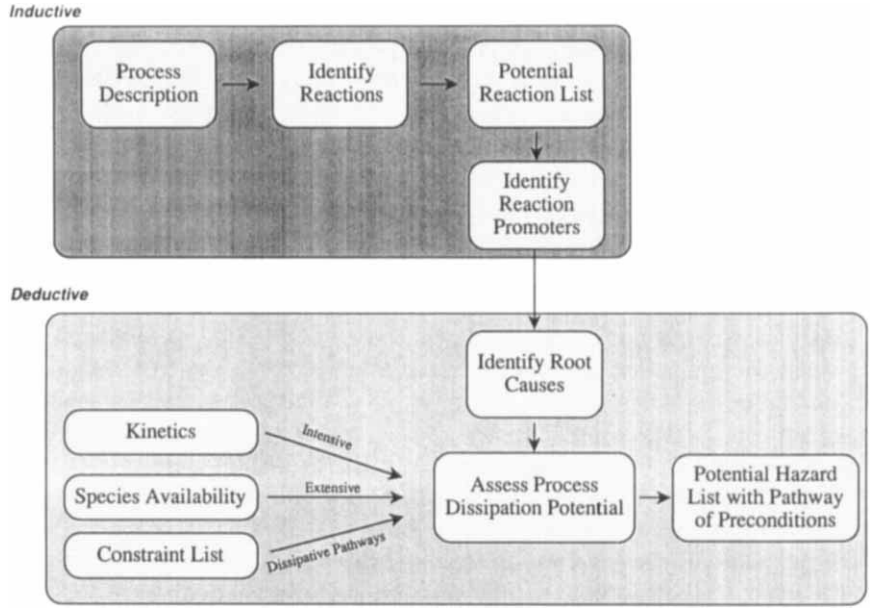

FIG. 9. Overview and implementational elements of the methodology for the identification and analysis of hazards.

conditions of a potential hazard are identified by establishing the conditions specifying the thermodynamic states that precede it. These conditions can be either intended or induced, resulting from changing boundary elements, system descriptions, or reacting states. Normally, they manifest themselves as procedural, material, or boundary changes.

A potential hazard is said to exist when the environment or a sequence of environments cannot be dissipated or prevented by the process. Using this environment as a goal state, we can then deductively identify the sequence of process and/or operational changes that led to its creation. Thus, the root causes and their temporal preconditions (whether they are equipment and/or operational failures) can be identified and associated with the corresponding hazard. Evaluation of the pathways leading to the specific hazard allows the quantification of risk and permits a concrete assessment of the potential hazard in the context of a given design technology.

Information flow during the execution of the hazards identification procedure is shown in Fig. 9. It is composed of two phases:

Phase 1. Inductive identification of reactive states and their assessment as potential hazards.

Phase 2. Deductive identification of pathways leading to those states.

The inductive phase begins with a process description from which reactions are identified using LCR, and the tools described earlier, which allow the identification of potential chemical reacting states. Once all the potential chemical transformations have been identified and the entire chemical species set has been elucidated, a **node** can be fully described and the **states** disseminating from it can be established (i.e., the $n + 2$ independent variable quantities are known). With this description potential physical reactions (e.g., overpressurizing, overheating, overcooling) can be elucidated using classical thermodynamic (open- and closed-system treatment) and transport phenomena analysis around the specific **node**.

Potential physical reactions can only be identified with any assurance of completeness *after* the complete specification of the chemical species set, CSS. This is a result of Postulate 1, $(n + 2)$ *defining variables must be known before a thermodynamic state can be specified*. Since incomplete specification of "n" leads to incomplete specification of potential states, a complete node description is required before physical reactions can be identified completely.

Once potential reactions are identified, they are posted on a potential reaction list. The promoters of each reaction are then identified by tracing the variable values that led to their creation. This is achieved through the links provided in the thermodynamic representation of the **state** and the **atom-bond-configuration** modeling objects. Similarly, by accessing the information contained in **K**, **K*** or $\mathbf{K}_{\text{ab initio}}$, we can also identify the conditions that promoted a particular chemical reaction. These conditions may be chemically induced, as in the case of nucleophilicity, or physically induced, as in the case of high-energy environments enabling radical or photochemistry, or both. These promoters are then used to establish the underlying pathways that enable the top-level event.

In this combined approach, the inductive identification of top-level events and the deductive identification of enabling pathways allow potential hazards and the sequence of events leading to them to be identified completely within the scope of the modeling effort. Moreover, since the top-level event is generated from its underlying states, the pathway of preconditions and the temporal ordering of those preconditions are explicitly spanning only the minimum cut set (see Nagel, 1991).

A. Hazards Identification Algorithm

The algorithms used in this approach are described below using the pseudocode conventions found in Cormen *et al.* (1990). Therein, a *terminal node* is defined as the "first node a feed enters or the last node a

product or byproduct exits" in the process flowsheet. The routine calls for two loops, which is necessary to simulate multiple simultaneously occurring boundary failures.

Algorithm 1 ⟨GLOBAL-HAZARD-IDENTIFICATION⟩

```
input: process-flowsheet
initialize
  process-node-representation ← apply MAKE-PROCESS-TRANSFORMATION
                                to process-flowsheet
  terminal-nodes ← apply FIND-ALL-TERMINAL-NODES to process-node-
                   representation
  potential-hazard-list ← nil
for each node in process-node-representation
   potential-hazard-list ← append (apply IDENTIFY-POTENTIAL-HAZARD
                           to (node process-flowsheet))
                           to potential-hazard-list
  return
for each node in terminal-nodes
   expand node-scope
     until UNIQUE-STATE-IDENTIFIED-P   ;predicate test to identify
                                        unique states
       or EXPANSION-NOT-POSSIBLE       ;empty node-scope
                                        (i.e., terminal node
                                        is reached)
    then
      potential-hazard-list ← append (apply IDENTIFY-POTENTIAL-HAZARD
                              to (node process-flowsheet))
                              to potential-hazard-list
  return
end
return
```

The internal routine, IDENTIFY-POTENTIAL-HAZARD, called from GLOBAL-HAZARD-IDENTIFICATION, returns a potential hazard and the list of the enabling conditions. The algorithm for IDENTIFY-POTENTIAL-HAZARD, expressed in pseudocode, is given below:

Algorithm 2 ⟨IDENTIFY-POTENTIAL-HAZARD⟩

```
input: node              ;starting point-hazards are associated
                          with nodes
       process-flowsheet ;context of the node
initialize
```

```
    potential-reaction-list ← nil
    associated-sates ← nil
    extended-CSS ← nil
    potential-reaction-list ← apply FIND-ALL-PATHWAYS
                              TO CSS of node; identify
                                              potential
                                              chemical
                                              reactions
    extended-chemical-species-set ← collect UNIQUE-CHEMICAL-
                                    CONSTITUENTS from
                                    potential-reaction-list
    associated-states ← apply EVALUATE-STATES
      extended-CSS    to node and ;identify states
                                  associated with each
                                  node due to reaction
  for each state in associated-states
    when UNIQUE-STATE-DESCRIPTION-P
      potential-reaction-list ← apply EVALUATE-PHYSICAL-
                                REACTIONS to state
      return
    return
  for each reaction in potential-reaction-list
    enabling-criteria ← collect FIND-ENABLING-CRITERIA
    classified-influence-paths ← apply CONSTRUCT-VARIABLE-INFLUENCE-
                                 PATHWAYS to (enabling-criteria process-
                                 flowsheet)
    root-causes ← apply IDENTIFY-NONDISSIPATIVE-PATHWAYS to classified-
                  influence-paths
    when root-causes
      potential-hazard ← list reaction enabling-criteria root-causes
      return
    return
  end
  return
```

Notice the interplay between the methodology and the specialized modeling languages and the importance of that interplay. These languages enable the methodological approach. For example, FIND-ALL-PATHWAYS applied to a chemical species set (CSS) utilizes the semantic relationships of LCR to construct the *potential-reaction-list*. Similarly, the representation utilized by LCR allows enabling-criteria to be explicitly associated with each reaction; these criteria can come from the **state** representation

or from the inherent physicochemical properties associated with the chemical constituents themselves. Likewise, EVALUATE-PHYSICAL-REACTION utilizes functionality of the base representation to assess possible effects and to identify enabling criteria. These criteria come from the operating environment of the process equipment.

By interweaving the semantic relationships of the two modeling languages throughout the methodology, we are able to interplay the respective representations to satisfy the representational needs of the task at hand. For example, root causes are identified by propagating (i.e., solving) the enabling criteria through the network of equations lying in the base representation and defining the behavior of the overall process flowsheet. These criteria, however, are located in different representations, at different abstraction levels, to afford the resolution necessary to identify potentially hazardous conditions. The specialized modeling languages provide the tools to map between different representational forms and identify potential hazards with their enabling conditions and underlying root causes in an efficient manner.

Two additional features of Algorithm 1 are worth discussions:

- Interval values, describing ranges of operating conditions, are *dynamically* redefined whenever a condition is found to exceed the existing value range (values may be exceeded in the positive or negative direction).
- The description of a **node** (i.e., the scope of the defining system) can be expanded to simulate common boundary failure or process equipment malfunction.

B. Properties of Reaction-Based Hazards Identification

Let us now consider the relative efficiency of different methods for the identification of hazards (for details and proofs, see Nagel, 1991). Since it is meaningless to compare the efficiency of methods that are incomplete versus those that are complete, for the purposes of this section we consider only complete methodologies. But before beginning, we should point out one additional result that can be derived from the completeness analysis presented in Section II.B.

Corollary. *Equipment-based methodologies for hazards identification can be complete.*

An equipment-based methodology, e.g., HazOp Analysis, begins the identification of hazards with the postulation of specific process variables

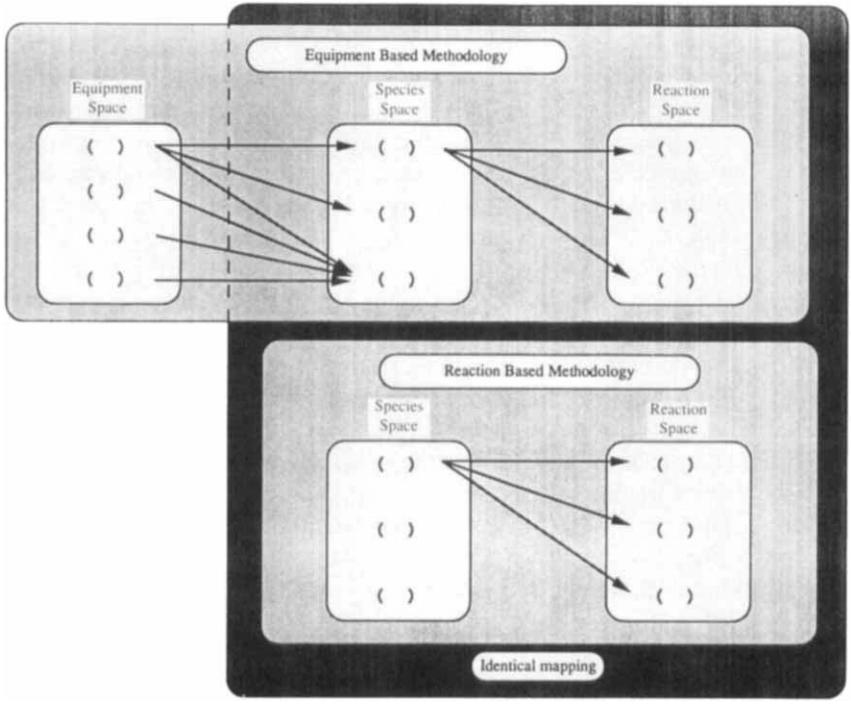

FIG. 10. Mapping of equipment space to species space.

deviations and/or equipment failures. From the point at which the complete chemical species set is generated, an equipment-based methodology is identical to a reaction-based methodology (see Fig. 10), such as the one described in this chapter. The implication of this statement is that the only information generated by the equipment analysis that is used in the identification of top-level events is the state space, which collapses to the species set since temperature and pressure are local variables.

Theorem. *No hazards analysis method exists that is both complete and whose running time is bounded by a polynomial in the number of initial species present in the plant.*

Theorem. *A reaction-based method is more efficient than any equipment-based method.*

The only difference between the equipment and reaction-based methods is in the generation of the chemical species set. Equipment-based methods generate the species set *indirectly* via the equipment state space; a

reaction-based method generates it *directly*. Since both approaches must generate every element in order to supply a complete set of chemical species, the difference in efficiency between an equipment-based method and a reaction-based method must result from the fact that the former must generate the members of the chemical species set more than once, while the latter as we know does not. But, this assertion is trivially satisfied if the size of the equipment space is larger than the size of the species set space (see Nagel, 1991).

However, the equipment-based approach is also engineered to perform a second task, namely, *identification of root causes*. It can be shown (see Nagel, 1991) that if the number of chemical species present in the CSS is S_0, the possible number of subsets of species and thus the maximum number of possible reacting mixtures is 2^{S_0}. Consequently, when the complexity of an equipment-based methodology becomes larger than 2^{S_0}, the added complexity is an effect of the method's intention to also identify the root causes of hazards. Assuming that c_1 represents the number of additional guidewords in the selection of hazards, and c_2 the number of additional process parameters invoked by an equipment-based hazards identification method, then the resulting complexity of such method is given by

$$(2 + c_1)^{(S_0 + c_2)^E},$$

where E represents the pieces of equipment to be searched for the identification of hazards. For a mixture of 2 chemical species the corresponding search space of a method based on the chemical species (as the one presented in this chapter) is $2^2 = 4$. On the other hand, for a simple process with 2 pieces of equipment, $c_1 = 1$ and $c_2 = 2$, an equipment-based hazards identification technique must search a space of $3^{4^2} = 6561$ alternatives.

However, as shown in Section IV, completeness cannot be guaranteed in the identification of root causes by equipment-based techniques, since they cannot guarantee that all reacting mixtures can be identified, and thus not all possible thermodynamic states can be a priori identified. This is because guidewords and variable parameters (e.g., more, less) are designed to trace out *causes*, not generate thermodynamic states. This guarantee only comes when the resolution of the tracing mechanism completely defines the enabling state of the potential hazard.

For example, *to identify a potential hazard that is caused by changing the catalyst shape an equipment-based approach must include surface area as a process parameter to be searched; a piece of information embedded in the phenomenological kinetic rate expression and not explicitly available.* Thus,

as a consequence of its formulation and the uncertainty of conditional operators, the equipment-based approach may not guarantee complete identification of potential hazards and cannot guarantee the complete identification of root causes leading to those hazards.

C. An Example in Reaction-Based Hazard Identification: Aniline Production

In this section, we will demonstrate how the methodology can be used to identify hazards inductively from the set of possible chemical reactions. We will focus on the detection of potential hazards arising from changing operating conditions rather than equipment malfunction or failure; the latter is discussed in Section IV.

The ability to elucidate competing chemical pathways when changes in control strategies occur is of particular importance in hazard identification. An understanding of the relationship between the reaction pathway topography and design and operating characteristics allows us to mitigate or constrain the offending reaction trajectory. Consider, for example, the catalytic production of aniline from nitrobenzene and hydrogen. Limiting our attention to the reactor, named **reactor-1**, we see that the initialization routine of the procedure GLOBAL-HAZARD-IDENTIFICATION transforms the reactor into a single terminal process node. The identification of hazards within this **node** is carried out by applying the procedure IDENTIFY-POTENTIAL-HAZARD, to this **node**. The procedure, IDENTIFY-POTENTIAL-HAZARD, begins by applying the procedure FIND-ALL-PATHWAYS to the chemical species set. This set, which is bound to the **node**, contains the known chemical species, i.e., CSS = {*nitrobenzene*, *hydrogen*, *Raney nickel*, and *phenol*}. Also, let **reactor-1** be bound to the **node**. The call to the procedure, FIND-ALL-PATHWAYS, and its application to the initial chemical species set, using composite chemical operators involving hydrogenation, is shown below:

$$(\text{FIND-ALL-PATHWAYS} \quad :\textit{substrates}\ \text{CSS} \quad :\textit{operators}\ \boldsymbol{K}_{\textbf{hydrogenation}})$$

Key reactions identified by the procedure, FIND-ALL-PATHWAYS, are

$$\varphi\text{-}NO_2 + H_2 \rightarrow \varphi\text{-}NO + H_2O, \qquad \text{(VIII)}$$

$$\varphi\text{-}NO + H_2 \rightarrow \varphi\text{-}NHOH, \qquad \text{(IX)}$$

$$\varphi\text{-}NHOH + H_2 \rightarrow \varphi\text{-}NH_2 + H_2O. \qquad \text{(X)}$$

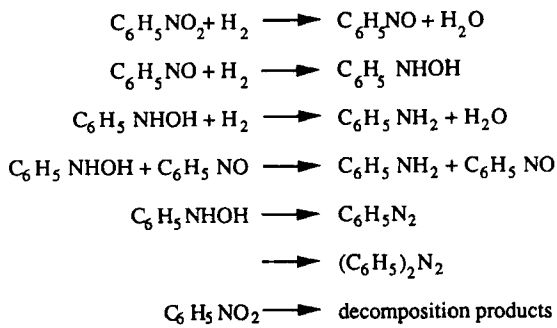

FIG. 11. Potential reactions of nitrobenzene.

These reactions combine to form the following overall reaction:

$$\varphi\text{-}NO_2 + 3H_2 \rightarrow \varphi\text{-}NH_2 + 2H_2O. \qquad \text{(XI)}$$

A less restrictive implementation of the procedure, FIND-ALL-PATHWAYS, on the initial chemical species set, would employ the *:operators* **K** or **K*** and would generate several additional reactions of interest. Among these additional reactions of particular interest are, the decomposition reaction of nitrobenzene and the disproportionation of nitrobenzene with N-phenyl hydroxylamine forming aniline and nitrobenzene:

$$\varphi\text{-}NO_2 \rightarrow \text{products}, \qquad \text{(XIII)}$$

$$AI\varphi\text{-}NO + \varphi NHOH \rightarrow \varphi NH_2 + \varphi - NO_2. \qquad \text{(XIV)}$$

Several of the pathways that emanate from these reactions and are constructed by a recursive application of the procedure, FIND-ALL-PATHWAYS, are shown in Fig. 11. Figure 12 shows how this information is managed by LCR using its modeling elements. This information is unavailable in conventional chemical synthesis programs (e.g., SECS and CYCLOPS) and provides an explanation (using the semantic relationships advanced by LCR) as to why the various reactions were activated and the specific intermediates were formed. For example, **pathway-1**, representing the transformation of nitrobenzene into aniline, "knows" that it ***is-disaggregated-in*** three separate individual reactions, denoted by **hydrogenation-1, hydrogenation-2**, and **hydrogenation-3**. Similarly, it "knows" that it ***is-abstracting*** the more general global reaction, **reaction-pathways**. It also "understands" which pathways are in competition; these in turn can be abstracted or disaggregated using the semantic relationships of LCR. In this way, the chemical sequence of events that led to a specific reaction can be traced all the way back to the reaction conditions that enabled it.

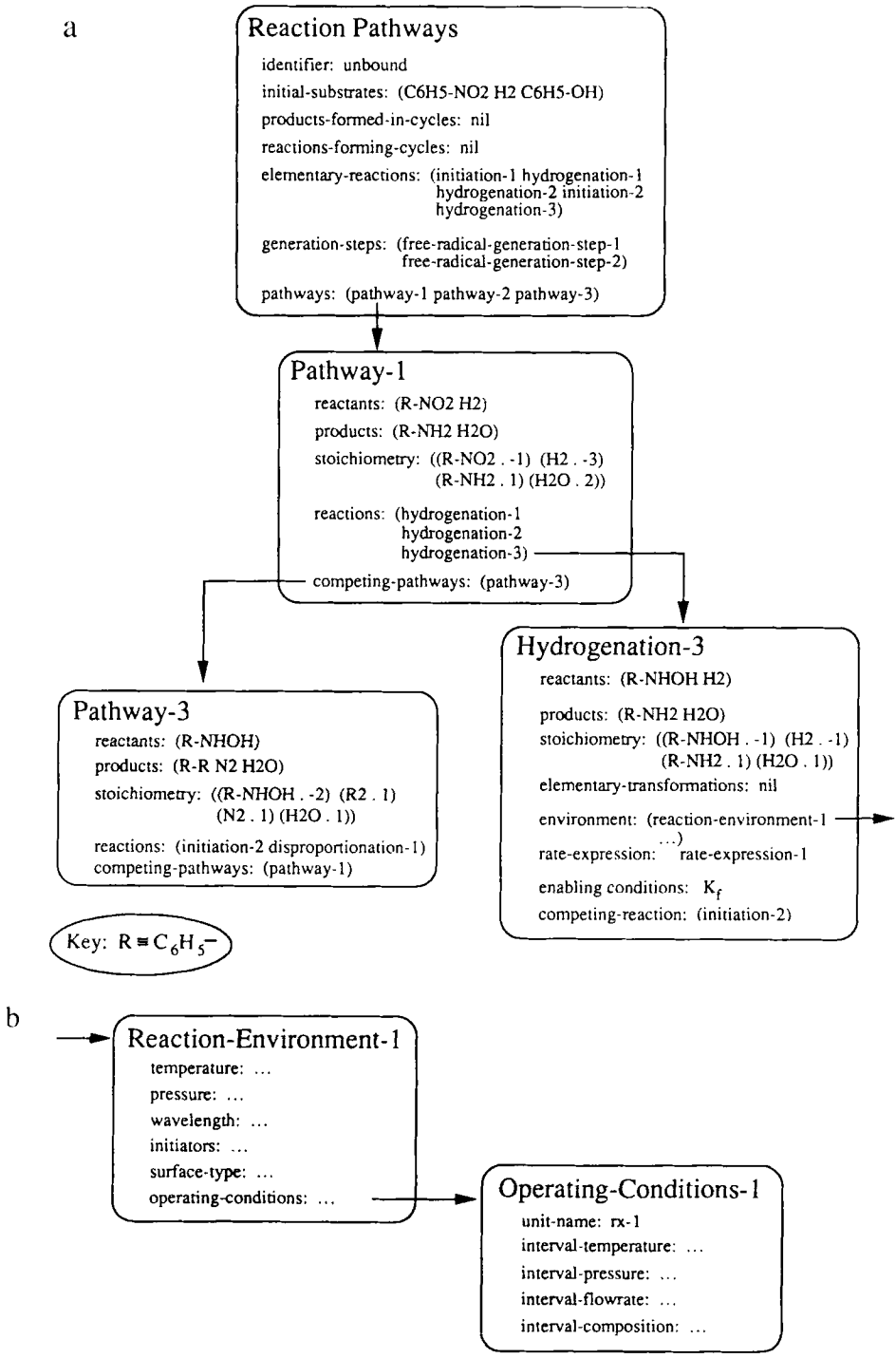

FIG. 12. Linkages between (a) reactions and pathways, (b) reactions and their operating conditions.

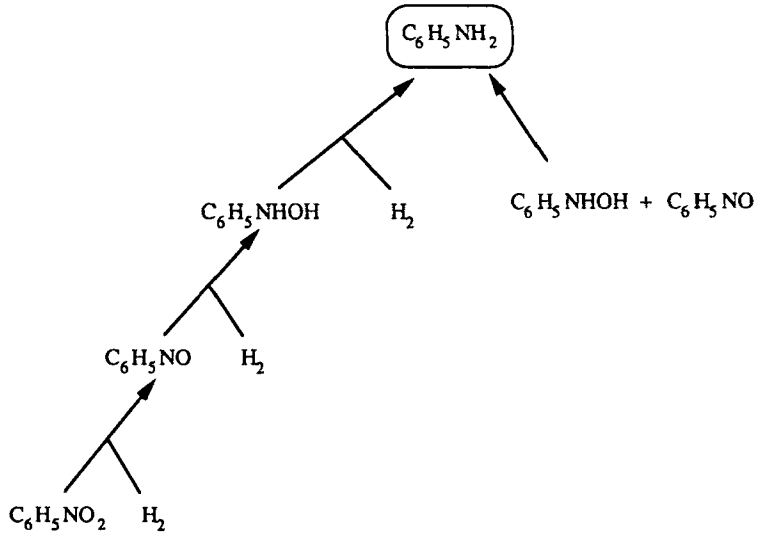

FIG. 13. Reaction path to aniline.

These enabling conditions can be related, in turn, to the operating conditions of the particular processing unit(s), using the *parent-equipment* attribute of *system-description* (an attribute of **node**). Through this mechanism the procedure FIND-ENABLING-CRITERIA traces out the sequences of events that lead to the reaction of interest. Figure 13 shows the path that leads to the formation of aniline. Because of the information contained in the LCR modeling objects that constitute this path, we know precisely the conditions that led to the creation of each element contained in the path. For example, by accessing the *heat-of-formation* attribute associated with the modeling element **reaction**, we identify that heat liberation is non-uniform during hydrogenation. This implies that an accumulation of the intermediate, φ-NHOH, could lead to an uncontrollable energy release.

More importantly, the tree spanning the generation pathways for aniline production (see Fig. 13) makes explicit the fact that hydrogen injection is an insufficient means for controlling reaction temperature. This results from the disproportion reaction identified by the procedure, FIND-ALL-PATHWAYS. As a consequence, a reduction in the hydrogen injection rate and regulation of the cooling fluid rate are necessary to control the heat released by the reaction. The realization that the heat released by the reaction cannot be controlled through hydrogen addition, alone, brings out an important parameter for the safe operation of the reactor, because the experimental detection of the *N*-phenyl hydroxyl-

amine disproportionation with nitrosobenzene is difficult without prior knowledge of its occurrence (Stoessel, 1989).

Since LCR specifies the preconditions associated with each reaction generated by the procedure, FIND-ALL-PATHWAYS, the conditions that enabled a specific chemical transformation can be easily identified. This knowledge allows us to make explicit such information as the decomposition temperature of φ-NO_2, the disproportionation temperature of φ-NHOH, and the preconditions for the exothermic formation of such products as azobenzene and diazobenzene. By expanding the node description to include the atmosphere around the reactor (e.g., air), the initial CSS can be expanded to include *oxygen*. Then, the procedure, FIND-ALL-PATHWAYS, can identify additional reactions involving oxygen, e.g., combustion reactions with hydrogen, phenol, and Raney nickel as well.

The analysis of the inductively synthesized chemical pathways, described above, indicates that, if one started with only an understanding of the overall reaction, i.e., the hydrogenation of nitrobenzene to form aniline and water, the analyst may not have recognized the need to control reaction temperature, while moderating both cooling rates and hydrogen injection rates. Lacking this information, the loss of recirculation or the need to provide emergency cooling to satisfy additional duty requirements resulting from *N*-phenyl hydroxylamine accumulation, or potentially the need for redundant recirculation pumps may not have been understood and appreciated.

IV. Deductive Determination of the Causes of Hazards

The fundamental premise in all of the approaches that attempt to mitigate or control hazards lies on the assumption that they have the ability to both (1) identify accurately and (2) pinpoint precisely the location of a potential future hazard. We have shown that, although understanding the set of enabling conditions is essential for safe plant operation, the identification of the entire set of enabling conditions is an intractable task. It is impossible to completely identify the set of enabling conditions, leading to a hazardous state for the following reasons. The physicochemical conditions (e.g., presence of certain chemicals within predefined ranges of compositions, temperatures, pressures, flowrates within certain ranges of values) enabling the activation of particular chemical transformations form a set whose members possess variable values. When these variable-valued conditions are substituted into the

equations representing the behavior of a chemical plant, the variables defining the conditions for the occurrence of a top-level event (TLE) (e.g., mixing of two chemicals forming an explosive mixture) achieve values necessary for the occurrence of a hazard. Since the variable-valued preconditions for a hazard are drawn from a range of, theoretically, infinite real-valued alternatives, we conclude that it is intractable to deduce all possibilities leading to a hazard. [*Note:* Expressed in different words, the analytic intractability in determining all conditions leading to hazards comes from the following weakness: There are no analytic (numerical) procedures that can take a set of equalities (e.g., the equations modeling the behavior of a plant) and a set of inequalities (e.g., preconditions for the activation of chemical pathways), solve them together, and produce a consistent set of new inequalities.]

Shifting to an interval representation, where we bracket the variable-valued preconditions into arithmetic intervals, over which the behavior is safe or unsafe, is only a partial solution for two reasons:

- First, the boundaries of the intervals, defining the conditions for the occurrence of a TLE are essentially functions of the values that the preconditions for the accomplishment of a reactive pathway can take on.
- Second, if the equations modeling the dynamic behavior of a plant display chaotic solutions (inherent in many non-linear systems), then we cannot be certain that the process behaves the same way when the preconditions defining a reactive pathway take on any values within given arithmetic intervals of values.

As a result, we have focused on the interpretation of the pathway *leading to a hazardous state* and its topography, and how these relate to the inherent safety of the process design technology rather than on the elucidation of pathways *leading to top-level events*. In the absence of a methodology for the complete identification of all conditions enabling the occurrence of a TLE, such an approach is essential if the safety of a chemical operation is to be enhanced.

A. Methodological Framework

In the previous section we discussed how an inductive approach can be used to generate all the chemical reaction pathways and the associated thermodynamic states, which lead to top-level hazardous events. A potential hazard is said to exist when the thermodynamic state or sequence of thermodynamic states leading to the hazard cannot be prevented, or the

impact of these states cannot be dissipated by the design or the operating capabilities of the process. Using the precursor thermodynamic state(s) as a goal state(s), we can then deductively identify the sequences of changes in the structure of the processing system and/ or its operation that led to the creation of the hazardous state. Consequently, root causes and their temporal preconditions, whether equipment and/ or operational failures, can be identified and associated with the known hazard. Evaluation of the paths, leading to the hazard, permits the quantitative assessment of the top-level event and affords the risk assessment of the potential hazard in the context of the process design technology. In this section, we will show how the structure of the path itself provides a metric for assessing the inherent safety of the process design technology being evaluated.

The deductive identification of paths leading to a top-level event is accomplished through a *recursive tracing of the variable-influence links*, which describe the transition from state to state on the way to the hazardous event. These *variable-influence* links describe how the various physiochemical variables affect each other, and are generated from the network of modeling relationships describing the behavior of a plant. By propagating the values of the enabling/ promoting conditions through the network of modeling relationships, one can assess the ability of the process to dissipate potentially hazardous effects. Obviously, the completeness and correctness of the results obtained through such an approach depend on the completeness and correctness of the modeling relationships. The modeling languages, LCR and MODEL.LA., are used to capture the requisite modeling relationships, which in addition to the first-principles-based equations, may include heuristics, empirical knowledge, experimental correlations, and design decisions. For example, LCR is used to capture the modeling of chemical kinetics, description of chemical structures for the computation of physical properties, whereas MODEL.LA. is used to capture the topology of the unit operations in a plant, the material and energy balances around the unit operations, and the description of operating variables (e.g., temperature, flow, pressure, composition).

Although the knowledge required to assess the potential for the prevention or dissipation of hazards is often held by different abstractions of the overall representation, the semantic relationships of the modeling languages afford efficient access to this information. The effect of protective processes, equipment restraints, sensors and control systems, emergency procedures, etc., are captured as constraints. Constraints may be embedded in the underlying representation as equations, or be associated directly to it via a constraint list, i.e., a collection of explicit process restraints. These restraints may be *passive*, such as materials of construction, or

active, such as protection processes, sensors and control systems, and operating procedures. *Constraints limit the variable value of an enabling condition. They may achieve this task in one of two ways: demand mitigation or demand prevention.*

Mitigation requires no corrective action to limit a variable value. It is accomplished through the appropriate design of an equipment that (1) leads to the restraint of the top-level event, given an enabling condition, or (2) limits the value that an extensive (e.g., flow, total inventory of a material) or intensive variable (e.g., composition, temperature) can achieve. For example, the value of an intensive variable can be limited by the physical phenomena that are allowed to occur within a process. Orchestration of these phenomena, or where and how they occur, can place a restriction on the type of species that may be generated, the rates at which reactions occur, the separation of constituents into phases, etc. Similarly, extensive variable values such as equipment volume, maximum accumulation, or total mass, can be limited by process equipment design. Since the equations describing a process are a manifestation of these phenomena, select modification of the *variable-influence links* (i.e., of the cause-and-effect pathways) can maximize the inherent safety of the design technology. More importantly, *modification of the topology of the variable-influence links or limitation of the achievable variable values by the enabling preconditions are the only means by which the inherent safety of a process design technology can be altered.*

Preventing the occurrence of an enabling condition requires the undertaking of a corrective action. These actions are designed to limit the variable value of an enabling condition; however, they do so through active participation of protective equipment (e.g., release vents), control loops, or operating procedures. Like the mitigation, the prevention of enabling conditions necessitates an understanding of the *variable-influence links* (i.e., cause-and-effect links) and consequently, of the underlying physicochemical phenomena. We begin the elucidation of the *variable-influence links*, leading to a potentially hazardous states, by first identifying the $(n + 2)$ independent variables that describe the potentially hazardous state. Since process design technologies are describable by the network of process modeling relationships (i.e., topology of the process flowsheet, material and energy balances, chemical and phase equilibrium relationships, kinetic and transport rate expressions, constitutive equations) that define the interactions of various variable quantities, we can trace out the influence of each variable that is associated with the state preceding the TLE. A trace of the variable-influence links defines the pathway of cause-and-effect interactions, which lead to the TLE.

B. Variables as "Causes" or "Effects"

The variable-influence pathway leading to the top-level event is constructed from the structure (Boolean) form of the incidence matrix, representing the network of process modeling relationships. But, the variable-influence pathways encompass certain information on directional causality. Consequently, it is important to identify the role of any variable in the set of modeling relationships: is a variable an input (cause), an output (effect), or of indeterminate directionality? In order to assign a role to each variable, we have developed a specific methodology that will be described in the following paragraphs. Consider the Boolean form of the incidence matrix that represents the modeling relationships, determining the behavior of a specific plant. These relationships capture all available knowledge about the plant; i.e., they are not limited to first-principles modeling equations, but include qualitative, order-of-magnitude, empirical correlations, etc. The fact that the matrix is in a Boolean form allows the simultaneous presence of relationships with inhomogeneous variables (i.e., real-valued, interval-valued, qualitative, or logical variables).

An *input-variable*, i.e., a variable that indicates the influence of the surrounding world on the process is clearly a potential *cause* and never an *effect* of the process' behavior. Typical examples of such input variables are design specifications, setpoints of control systems, characteristics of process feeds. The values of the input variables are established by factors external to the process. Input variables are generally extensive variables. An exception occurs when invariant intensive variables are associated with a feed. For example, the oxygen concentration in air may be considered invariant, when it is being used as a feedstock (e.g., formaldehyde production from air and methanol). Similarly feedstocks delivered to the process with concentration specifications can become inputs (e.g., 100% methanol). Manual settings by operators are input variables. These include manual valve manipulations, setpoints of controllers, structure of control loops, and starting or shutting of pumps. Furthermore, *all potential failures must be characterized as input variables*. This result occurs because potential failures are sources in the set of process equations that drive the causality in a particular direction. Therefore, we can use the following rules (see also Nagel, 1991) for the unambiguous characterization of certain variables as input variables (causes):

Rule 1. All influences of the surrounding world on a particular process are characterized as input variables, and are considered as causes of subsequent evolutions in the thermodynamic state of the process.

Rule 2. Process design specifications, and manual setting of operating variables and controller parameters are considered to be input variables, i.e., causes of subsequent events.

Once the set of input variables, associated with the influence of the surrounding world, has been identified, a systematic procedure examines the remaining variables in an effort to determine their unambiguous role as inputs or outputs. It should be clear at the outset of the subsequent discussion that *the unique and unambiguous characterization of all the process variables as inputs (causes) or outputs (effects) is in general impossible, and corresponds to an undecideable proposition.*

The assignment of a variable as *output* variable is always associated with a particular modeling relationship and implies that the variable takes on its value from the solution of the corresponding equation. Thus, it is identical to the concept of an output variable within the scope of the *input/output set assignment* for the solution of a set of nonlinear algebraic equations. But, unlike the solution of algebraic equations, an *output* variable within the scope of the deductive identification of hazards indicates a *physical consequence*, i.e., an *effect*, resulting from a specific set of causes (i.e., input variables). Consequently, we cannot use the variety of algorithms which have been developed for the identification of input/output set assignments, but we can employ some of the same ideas. Here are some of the rules which guide the selection of output variables (for detailed discussion, see Nagel 1991):

Rule 3. A variable occurring in a relationship, whose remaining variables have been characterized by Rules 1 and (or) 2 as input variables, is an output variable. It represents an effect of the process' behavior and can never be a cause.

Rule 4. A variable can be an output from only one relationship. Therefore, a variable already assigned as the effect (i.e., output variable) of a particular relationship, will be treated as cause (i.e., input variable) in subsequent relationships.

Rule 5. When a variable is the only variable in a particular relationship, then it must be characterized as output.

Rule 6. When a variable occurs in only one relationship and has not been characterized as input variable by Rules 1 and (or) 2, then it must be an output variable.

It is clear from the Rules 1–6 that the assignment of a variable as a *cause* or *effect* is guided by strict and unambiguous physical causality arguments. For example, if all the variables except one in a physical relationship have

been characterized unambiguously as causes, the remaining variable *must* be the effect of the particular relationship. Also, it is obvious that if a particular variable has been characterized as the effect of a particular relationship, it can only be a cause in subsequent relationships. Finally, the variables that specify the state preceding a top-level event must be output variables.

It is also clear that Rules 1–6 do not resolve the character of all variables appearing in a set of modeling relationships. Thus, after a repeated application of Rules 1–6, two or more relationships with the corresponding unassigned variables, remain to be characterized. Such subsets of relationships should be "solved" simultaneously and can produce a number of alternative sets of variables, which could have been assigned as output. Clearly, any of these alternative assignments is arbitrary and does not reflect any unambiguous physical causality. What is more important is the fact that *the determination of unambiguous cause-and-effect links among the variables of the remaining relationships corresponds to an undecideable proposition, which can only be resolved through additional independent knowledge.*

Under conditions of incomplete assignment and ambiguity on the role of various variables in the cause-and-effect links, we have adopted a conservative attitude, exploring all potential causalities that may be produced from the modeling relationships.

C. Construction of Variable-Influence Diagrams

Let us now see how the ideas of the previous section can be used to construct the variable-influence diagram, that defines the paths leading to top-level event. Consider a set of four modeling relationships, represented by the structural matrix shown below.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1			x		x		x
2		x			x	x	x
3	x	x	x	x		x	
4		x		x		x	

The columns represent variables in the defining process relationship and rows represent the relationships themselves. In accordance with the definitions given above, construction of the variable-influence path proceeds

as follows:

1. Assignment of inputs:
   Step 1—assign constants as inputs (Rule 2)
   Step 2—assign invariant intrinsic properties of feeds as inputs (Rule 1)
   Step 3—assign setpoints as inputs (Rule 2)
2. Assignment of outputs
   Step 1—assign variables occurring in only one equation as outputs (Rule 6)
   Step 2—assign single variables in equations as outputs (Rules 3, 5)
   Step 3—eliminate assigned variables and corresponding equations (Rule 4)
   Step 4—assign dependent variables of scientific equations and definitions as outputs
   Step 5—repeat

In the structural incidence matrix given above, there are seven variables (*columns:* terms on abscissa, i.e., on horizontal axis; viz., x_1–x_7) and four relationships (*numbered rows:* terms on ordinate, i.e., on horizontal axis; viz., 1–4). Therefore, three variables must be assigned as inputs to specify the system. They are x_2, x_3, and x_5. Rewriting the structural matrix by moving the input variables to the right-hand side and delineating them with a vertical line, we have

	x_1	x_3	x_6	x_7	x_2	x_4	x_5
1		x		x			x
2			x	x	x		x
3	x	x	x		x	x	
4			x		x	x	

,

where the left-hand side of the structural matrix represents outputs and the right-hand side represents inputs. Using the definitions shown above, the following variables are characterized as effects (i.e., outputs): x_1 appears only in relationship 3 and therefore receives its value from that equation; similarly, equation 4 contains a single variable on the left hand side of the structural matrix, therefore x_6 takes its value from relationship 4. After the elimination of rows and columns associated with each assigned output variable, x_3 is identified as taking its value from relationship 1, and x_7 taking its value from relationship 2. Thus, x_3 and x_7 are characterized as effects (i.e., outputs) from relationships 1 and 2, respec-

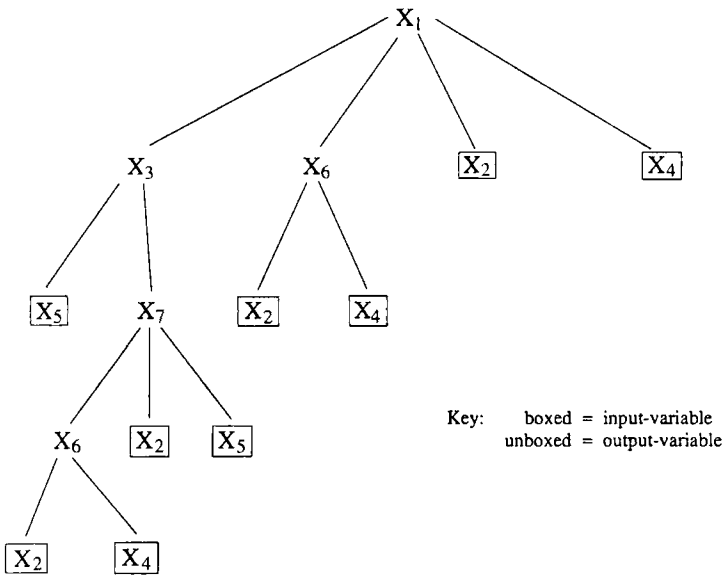

FIG. 14. Variable-influence pathway.

tively. The following structural incidence matrix indicates all assignments:

	x_1	x_3	x_6	x_7	x_2	x_4	x_5
1		**x**		x			x
2			x	**x**	x		x
3	**x**	x	x		x	x	
4			**x**		x	x	

The variable influence path leading to x_1 is shown in Fig. 14. As a consequence, x_1 can only be influenced by variables that appear on the path; causality is established by the input variables. This occurs because the output set assignment given to the set of relationships in the structural matrix is unique, as it is made evident by rearranging the tabulation order of relationships and variables comprising the structural matrix:

	x_6	x_7	x_3	x_1	x_2	x_4	x_5
4	**x**				x	x	
2	x	**x**			x		x
1		x	**x**				x
3	x		x	**x**	x	x	

Unique output set assignment occurs when the structural incidence matrix is triangular, such as the above. As a consequence, the variable-influence path leading to the top-level output variable (e.g., x_1) is also unique and can be affected only by inputs that occur along the path.

Alternatively, the network of relationships may not be represented by a triangle structural matrix. In this instance, the structural matrix may not have a unique output set assignment, and there may be alternative paths leading to the variable that describes the state preceding the top-level event. Alternative paths are created when loops exist in the set of process modeling relationships necessitating simultaneous solution. For example, in the structural matrix shown below there are three distinct output set assignments, leading to three distinct potential variable-influence paths. (*Note:* Underlined entries in bold indicate the assigned outputs to each relationship.)

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
1		**X**	x		x				x			x
2					**X**		x		x	x		
3				**X**			x	x	x		x	
4			**X**	x		x		x	x		x	
5					x	**X**				x		x
6					x	x	**X**					x
7								**X**				
8									**X**	x		x
9	**X**	x		x	x		x				x	

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
1		**X**	x		x				x			x
2					x		**X**		x	x		
3				**X**			x	x	x		x	
4			**X**	x		x		x	x		x	
5					**X**	x				x		x
6					x	**X**	x					x
7								**X**				
8									**X**	x		x
9	**X**	x		x	x		x				x	

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
1		**X**	x		x				x			x
2					x		**X**		x	x		
3				**X**			x	x	x		x	
4			**X**	x		x		x	x		x	
5					x	**X**				x		x
6					**X**	x	x					x
7								**X**				
8									**X**	x		x
9	**X**	x		x	x		x				x	

Alternatively output set assignments, and consequently variable-influence paths, result from the block structure established by variables x_5, x_6, and x_7. Thus, variable x_5 can take its value from any one of the three relationships, since there is no way that x_5 can be uniquely and unambiguously assigned as an effect to any one of these three relationships.

Construction of the variable-influence diagrams, indicating the paths of the cause-and-effect links, which lead from the external causes to the variables describing the potentially hazardous state, is achieved by using the procedure CONSTRUCT-VARIABLE-INFLUENCE-PATHWAYS, a procedure called from the procedure, IDENTIFY-POTENTIAL-HAZARD. This method traces the influence of various variables defining the state preceding the top-level event. It constructs variable-influence paths from the set of process modeling relationships contained in the structural incidence matrix. The algorithm used for the generation of variable influence pathways is given below:

Algorithm 3: ⟨CONSTRUCT-VARIABLE-INFLUENCE-PATHWAYS⟩.

```
input:                    process-flowsheet
    enabling-criteria     ;output from IDENTIFY-POTENTIAL-HAZARD
initialize
process-flow-equations ← apply IDENTIFY-PROCESS-FLOW-EQUATIONS
                          to process-flowsheet
VIM ← apply IDENTIFY-INFLUENCE-MATRIX to process-flow-equations
TLE-vars ← apply IDENTIFY-TLE-VARS to enabling criteria   ;identify top-
investigation-list ← TLE-vars                             level event
for each tree in investigation-list                       variables
if next-available-node ← apply IDENTIFY-NEXT-EXPANDABLE-NODE to tree
    then
    expansion-list
        ← apply EXPAND-NODE to VIM, tree, next-available-node
    endif
return
for each expansion in expansion-list
  apply CLASSIFY-BRANCH to next-available-node   ;classify
                                                  according to
                                                  technology-type
  if (apply CONTINUE-EXPAND-BRANCH-P to expansion,
    next available-node)
  then              ;predicate test to evaluate
                     branch expansion
    (append tree to investigation-list)
  endif
return
```

Whenever the procedure CONSTRUCT-VARIABLE-INFLUENCE-PATHWAYS has to generate a number of alternative variable-influence paths, it employs a breadth-first search strategy. The procedure, EXPAND-NODE, is used to establish the expansion list through the following algorithm:

Algorithm 4: ⟨EXPAND-NODE⟩.

```
input: VIM, tree, var
initialize
    expansion-list ← nil
apply CLEAR-ALL-ASSIGNMENTS to VIM
for each node in (apply PATHWAY-NODES to tree, var)
    apply ASSIGN-OUTPUT-VARIABLE to VIM, node
    return
for each equation in (apply EQNS-CONTAINING-VAR to VIM, var)
    if (apply VALID-ASSIGNMENT-P to VIM, eqn, var)
       then
          new-vars ← remove var from (apply VARS-IN-EQN to eqn)
          apply CLASSIFY-VARS to new vars
          new-tree ← apply APPEND-TREE to tree, node, new-vars
          append new-tree to expansion-list
       endif
    return
return
```

D. Characterizing of Variable-Influence Pathways

The procedure CONSTRUCT-VARIABLE-INFLUENCE-PATHWAYS applies to each node of the variable-influence pathway the procedure, CLASSIFY-BRANCH, which in turn classifies each branch of the pathway according to the type of the technology, which can be used to control the variable-value specifying the node. The algorithm of this procedure is given below:

Algorithm 5 ⟨CLASSIFY-BRANCH⟩.

```
input: node
for each child-node in (apply CHILDREN to node)
select case for child-node
    terminal-variable
is apply TAG-NO-EXPANSION to child-node      ;no node expansion
    endcase
    already-in-pathway
```

```
        is apply TAG-NO-EXPANSION to child-node
        endcase
    endselect

    select case for node
        inherent-controllability
        is apply TAG-TYPE-1-TECHNOLOGY to node      ;see definition below
        endcase
        Type-2-controllability and process-modification-applied
        is apply TAG-TYPE-2-TECHNOLOGY to node
        endcase
        Type-3-controllability and control-loop-applied
        is apply TAG-TYPE-3-TECHNOLOGY to node
        endcase
    endselect
    return node
```

Nodes are classified in this manner in order to make explicit the protective system responsible for mitigating a disturbance and its relationship to the top-level event. The procedure CONSTRUCT-VARIABLE-INFLUENCE expands each node of the tree and classifies each branch according to the technology type that could be used to control the variable-value, using the following definitions:

1. A *Type-1 technology* is capable of reducing the number of TLEs associated with the specific design, or is capable of modifying the topology of the path leading to the TLE through a change in process chemistry or the structure of the designed process flowsheet, provided design changes do not involve the introduction of *Type-2* or *Type-3 technologies*. Examples include reactant substitution, catalyst introduction or substitution, byproduct reduction, chemical character modification, solvent substitution, and intermediate chemicals elimination. Industrial processes with *Type-1 technology* changes include the following: (a) substitution of the reactant raw materials, hydrogen cyanide and acetylene, with propylene, ammonia, and air; (b) production of butyl lithium in dilute solutions, which prevents spontaneous combustion in the presence of air; and (c) direct hydroxylation of ethylene forming ethylene glycol, rather than oxidation of ethylene forming ethylene oxide, which in turn is hydrated to form ethylene glycol, thus eliminating ethylene oxide as an intermediate. Changes in the pathway topology, leading to the TLE, require design modification of the unit operations, changes in the piping, substitution of unit operations, or layout modification.

2. A *Type-2 technology* limits the value that a variable, which is involved in the pathway leading to a TLE, can achieve without requiring an action. Examples involving *Type-2 technologies* include the following:

(a) reduction of inventories, (b) minimization of operating (process condition) extrema, (c) reduction of intermediate accumulation. Industrial processes incorporating *Type-2 technologies* include the following: utilization of in situ reactions to limit intermediate accumulation (e.g., methyl isocyanate consumption via in situ reaction reduces the need for storage), utilization of continuous processes to minimize process hold up (e.g., continuous nitration processes eliminate the need for batch manufacturing of nitroglycerine), and reduction in the intensity of processing conditions can lead to smaller material and energy releases (e.g., catalyst improvements have enabled the Oxo process to produce aldehydes from syngas and olefins at lower pressure).

3. A *Type -3 technology* limits the value that a variable, involved in the path leading to a TLE can achieve *and require an action to do so. Type-3 technologies* involve active control of process variables in a direct or indirect manner, in order to limit the value that they can achieve. Examples include the following: (a) manipulation of feed rates, (b) use of cooling water, (c) temperature control and pressure control, (d) catalyst activity assessment through the monitoring of conversion and selectivity.

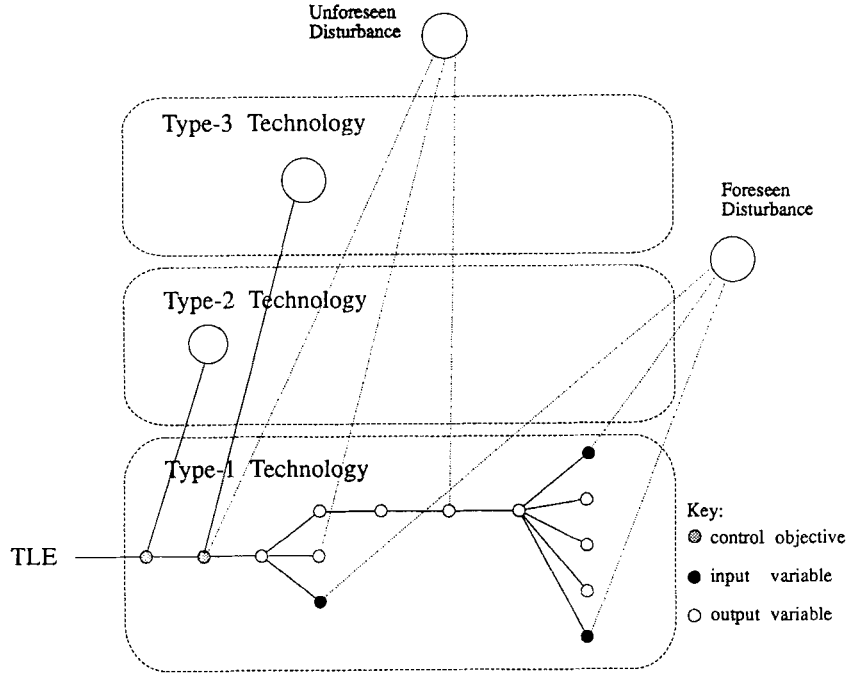

FIG. 15. Mitigation of disturbances through the specification of the control points.

The result of tagging the paths with the type of hazards-preventive technology is shown in Fig. 15. Having such a diagram one can then associate the different technology types with various parts of the variable-influence diagram, for the purpose of directly or indirectly controlling the variable values, composing the variable-influence pathway to the top-level event. Such an approach not only provides an understanding on how to control the values of each variable along the variable-influence path, but, also permit the designer to evaluate the feasibility of the process and its safety devices with respect to various operational criteria.

E. Assessment of Hazards-Preventive Mechanisms

The strategy we use for the specification of the most attractive hazards-preventive control objectives is based on *closeness*, i.e., hazards-preventing control objectives, which affect a variable that is at a minimum distance from the top-level event is preferred over those that affect more distant variables. However, the point on the variable-influence path, where the actual manipulation (e.g., design modification, controller or safety device) takes place, remains unspecified; it depends on whether the origin, type and intensity of disturbances are known ahead of time or not.

For disturbances that can be *foreseen*, it is preferred that the hazards-controlling objectives be placed near their origin (i.e., entry point of an input), in order to mitigate the effect of the disturbance before it has the opportunity to be amplified (see example in the diagram of Fig 15). *Foreseen disturbances* enter the process through the set of external variables or inputs. These include pump failures, valve failures, controller malfunctions, etc.

Notice, though, that the greater the distance between the location of the hazards-controlling objective and the top-level event, the greater the opportunity (i.e., the higher the probability) for the appearance of an unknown (*unforeseen*) disturbance along the pathway that connects the control objective to the TLE. *Unforeseen disturbances* can enter the process anywhere along the pathway and often change the pathway leading to the top-level event. Since the thermodynamic state preceding a top-level event is specified by the procedure, IDENTIFY-POTENTIAL-HAZARD, any disturbance, foreseeable or unforeseeable, must act as an input to these variables, if the TLE is to be enabled. Consequently, the only effective mechanism for the mitigation of unforeseen disturbances is the establishment of control objectives at *level 1*; the last defensive line before enabling the top-level event. This specification is translated into a series of conditional statements on the values of the variables at *level 1* of the variable-influence diagram (see Fig. 15). The methodology used by the

procedure, GLOBAL-HAZARD-IDENTIFICATION, enables the identification of TLEs independently of the pathways that lead to them. Additional TLEs are identified by the procedure GLOBAL-HAZARD-IDENTIFICATION as it expands the scope of the process description in its search for potential hazards. Expansion of the process description includes the reformulation of the boundary defining the process, thus allowing the incorporation of disturbances (e.g., as new inputs) that were "*unforeseen*" by the earlier formulation of the process description.

Type-2 technologies can be particularly effective in mitigating unforeseeable disturbances, because they do not require active control of the disturbance. Nevertheless, whenever possible, *Type-1 technologies* offer the best mechanisms for the mitigation of unforeseeable disturbances. This is accomplished through the addition or subtraction of relationships from the structural incidence matrix; operations that describe the recommended changes in chemistry, materials, type of unit operations, or structure of the process flowsheet. On the other hand, control of foreseeable disturbances is better handled by *Type-3 technologies*. Each variable along the variable-influence pathway is subsequently characterized by the type of hazards-controlling technologies, i.e., *Type-1, Type-2*, or *Type-3 technologies*, which could be employed for controlling the value of the specific variable. Such characterizations allow fast and effective screening of the potential disturbance mitigation alternatives.

The procedure IDENTIFY-NONDISSIPATIVE-PATHWAYS constructs the set of enabling conditions that lead to a top-level event. It takes as its input the variable-influence pathways, which it obtains from the procedure, CONSTRUCT-VARIABLE-INFLUENCE-PATHWAYS. TLE variables are then identified, and each variable contained in the set is traced to identify potentially feasible roots for causing the TLE. When an achievable root is identified (i.e., an input disturbance, controlled or uncontrolled, which can enable the top-level event), it is collected into root causes and returned. The algorithm used by this procedure is shown below:

Algorithm 6: ⟨IDENTIFY-NONDISSIPATIVE-PATHWAYS⟩.

```
input: classified-influence-paths
initialize: TLE-variables ← (collect-TLE-variables)
for each variable in TLE-variables
    when (unique-path-p classified-influence-paths)   ;predicate test for
                                                        identifying unique
                                                        paths
        (apply CONSTRUCT-FEASIBLE-ROOTS                ;constructs path-
            to TLE-variables classified-                ways or trees
            influence-paths)                            which support
```

```
                                                    enabling criteria
        and collect into root-causes
        return
    else
        for each path in classified-influence-paths
        to (apply CONSTRUCT-FEASIBLE-ROOTS          ;constructs a
            to TLE-variables path)                   feasible root to
        and collect into root-cause                  the goal state
        return                                       (TLE variable)
                                                     path
    return goal state (TLE)
end
```

Figure 16 shows how the procedure IDENTIFY-NONDISSIPATIVE-PATHWAYS has identified a potential path, starting from a given TLE and deducing a specific (*foreseen*) disturbance as a potential cause for the given TLE. Furthermore, in Fig. 16 we can also see the type of the preventive technologies (as identified by the procedure, CLASSIFY-BRANCH) that are appropriate for each variable on the pathway, which leads from the foreseen disturbance to the specific TLE.

The procedure IDENTIFY-NONDISSIPATIVE-PATHWAYS calls the procedure CONSTRUCT-FEASIBLE-ROOTS, which assesses the feasibility of a particular variable trace to enable the TLE, given a disturbance as its input. The

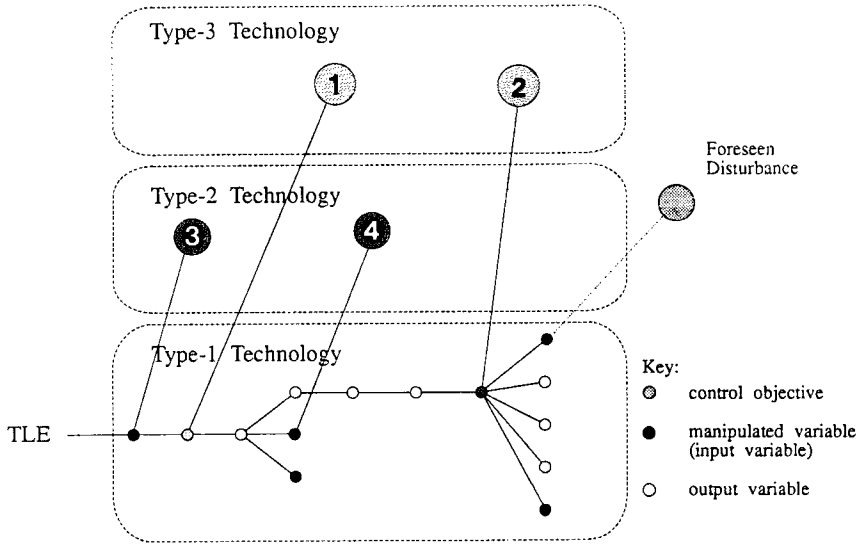

FIG. 16. Placement of technologies for the prevention of a specific TLE.

procedure CONSTRUCT-FEASIBLE-ROOTS also identifies the control mechanism responsible for the mitigation of the input disturbance. When a controlling mechanism is not associated with the input, the variable and its pathway are collected directly into roots. The algorithm defining the logic of the procedure, CONSTRUCT-FEASIBLE-ROOTS is shown below:

Algorithm 7: ⟨CONSTRUCT-FEASIBLE-ROOTS⟩.

```
input: variable-set
       pathway
when (enabling-feasibility-p variable-set pathway)
     for each variable in variable-set
          when (controlled-variable-p variable pathway)
               controller ← (get-controller variable pathway)
               collect variable, pathway, and controller into roots
               return
          else
               collect variable into roots
          return
     return
return
end
```

F. Fault-Tree Construction

The construction of the variable-influence pathways and the classification of its nodes in terms of the type of technology that can be used to mitigate the propagation of a hazard, constitute the essential basis for the generation of topological fault trees, a tool that can guide the evaluation of hazards preventive mechanisms. The procedure CONSTRUCT-TOPOLOGICAL-FAULT-TREE is at the core of this construction. It takes as its input the top-level event, the associated hazardous state preceding the TLE, and the root causes for each variable defining the hazardous state. Using these inputs, it determines the reactions responsible for the hazardous state and the enabling criteria of the reaction. With this information, it constructs a *level-1-gate:* a logical gate immediately preceding the top-level event. This gate establishes the demands (i.e., inputs) of the top-level event. By looping through the set of demands associated with the *level-1-gate*, we can construct qualitative logical gates from a given demand input and its root causes. Since each qualitative gate has associated with it an input and an output, these can be linked together to form a tree. By linking the tree to the *level-1-gate* and in turn appending the *level-1-gate*

to the TLE, a topological fault-tree can be constructed. The algorithm used is as follows:

Algorithm 8: ⟨CONSTRUCT-TOPOLOGICAL-FAULT-TREE⟩.

```
input: TLE
       hazardous-state
       root causes
initialize: hazard-variables ← (get-hazardous-variables hazardous-state)
            reaction ← (get-reaction hazardous-state)
            enabling-criteria ← (get-enabling-criteria reaction)
            level-one-gate ← (construct-level-one-gate
                                 hazard-variables enabling-criteria)
for each input in level-one-gate
  gates ← (CONSTRUCT-TOPOLOGICAL-GATES input   ;gate construction
(get-root-causes ;input))
  tree ← (LINK-GATES input gates)              ;connects gates
  append tree to level-one-gate                 via their logical
  return                                        outputs
append TLE to level-one-gate
return
end
```

Basic gates are constructed using the procedure, CONSTRUCT-TOPOLOGICAL-GATES. This procedure builds *or-gates*, *and-gates*, and *special-gates*. *Special-gates are based on first-order predicate logic and require a certain amount of quantitative analysis prior to Boolean assignment* (*i.e., and-gates, or-gates, and special-gates*). The procedure receives an input variable and the root causes associated with that variable. Then, it uses the variable trace associated with the root causes to establish the pathway leading to the external input and the protective devices associated with that input. By collecting the controllers associated with the various controlled variables, the procedure identifies the necessary *and-gates*, *or-gates*, and *special-gates*. The algorithm used for the implementation of the procedure, CONSTRUCT-TOPOLOGICAL-GATES, is shown below:

Algorithm 9: ⟨CONSTRUCT-TOPOLOGICAL-GATES⟩

```
input: variable
       root-causes
initialize: and-gates ← nil
            or-gates ← nil
            special-gate ← nil
```

```
            initial-internal-input
                ← (get-internal-input variable   ;network
                    root-causes)                  starting point
    for each variable in (get-output-assignment root-causes)
      up from initial-internal-input
      when (unbranched-node-p variable)          ;predicate test for
          controller                              determining if node
              ← (get-controller variable          is branched
                  root-causes)
          when controller                        ;constraint
            collect (construct-and-gate          ;identification
                    variable controller          of variable
            (get-input variable))
              into and-gates
            and
            collect (construct-or-gate variable controller (get-input
                    variable))
              into or-gates
            return
          collect (construct-special-gate variable (get-inputs variable))
          into special-gates
          return
      gates ← append and-gates or-gates special-gates
      return
    return
    end
```

Complex topological fault trees are constructed using the gates identified by CONSTRUCT-TOPOLOGICAL-GATES. Since each logical gate has a set of inputs and an output, all the gates can be linked together by matching gate outputs to gate inputs.

Quantitative analysis of the pathway, leading to the TLE, is required to establish the logical basis for converting *special-gates* into structures (i.e., trees) containing *or-gates* and *and-gates*. By associating averaged probability and failure rate data with the resulting topological fault tree (i.e., with the variables and the technology type of preventive mechanisms), the latter can be transformed into a conventional fault-tree. Construction of the fault tree in this manner has particular utility because it is complete; all pathways leading to the TLE will be identified within the scope of the relationships describing the particular process design technology. This prevents incompleteness in the pathways leading to the TLE and mini-

mizes errors in the frequence of that event, often by more than three orders of magnitude (ICI, 1988). (*Note*: ICI has shown that incompleteness can result in estimates of the TLE that are off by as much as three to five orders of magnitude; whereas errors in probability and failure rate data lead to estimates that are often within a factor of 2–5.)

G. An Example of Reaction-Based Hazard Identification: Reaction Quench

Consider a process that consists of a reactor used for the processing of a highly unstable chemical that is sensitive to small increases in temperature. The reactor is equipped with a quench tank to protect the system against a runaway reaction and is monitored by two temperature sensors (see Fig. 17): T_1 and T_2. Sensor T_1 automatically activates the quench tank outlet valve when it detects a temperature rise above the specified upper limit. Sensor T_2 sounds an alarm in the control room to alert the operator to the process upset. When the alarm sounds, the operator closes the reactor inlet valve. The operator also pushes a quench tank valve button in the control room in case the quench valve fails to open. Note that A is the reactant; B, the product; and C, the quench.

The analysis for the deductive determination of root causes begins by constructing the set of relationships that describe the process flowsheet.

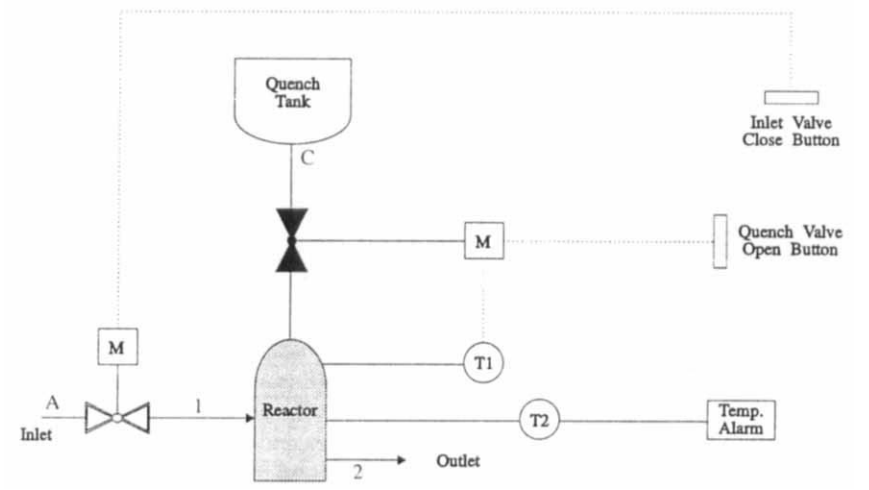

Fig. 17. The elements of the reactor-quench example.

These are shown below:

$$F_2 = f_2^{\mathrm{A}} + f_2^{\mathrm{B}} + f_2^{\mathrm{C}}, \tag{1}$$

$$f_2^{\mathrm{C}} = Q + F_1^{\mathrm{C}}, \tag{2}$$

$$f_1^{\mathrm{A}} = f_2^{\mathrm{A}} + r_{\mathrm{A}}, \tag{3}$$

$$f_1^{\mathrm{B}} = f_2^{\mathrm{B}} - r_{\mathrm{A}}, \tag{4}$$

$$F_1 = f_1^{\mathrm{A}}, \tag{5}$$

$$f_1^{\mathrm{B}} = 0, \tag{6}$$

$$f_1^{\mathrm{C}} = 0, \tag{7}$$

$$r_{\mathrm{A}} = k[C_{\mathrm{A}}]V_{\mathrm{RE}}, \tag{8}$$

$$k = A * \exp(-E^*/RT_{\mathrm{R}}) = f(T_{\mathrm{R}}), \tag{9}$$

$$[C_{\mathrm{A}}] = f(f_1^{\mathrm{A}}, Q, r_{\mathrm{A}}), \tag{10}$$

$$Q_{\mathrm{r}} = f(\Delta h_{\mathrm{r}}, r_{\mathrm{A}}), \tag{11}$$

$$0 = f(Q, T_0, F_1, T_F, Q_{\mathrm{r}}, F_2, T_{\mathrm{R}}), \tag{12}$$

$$Q = f(SP_1), \tag{13}$$

$$SP_1 = f(T_{\mathrm{R}}, T_1), \tag{14}$$

$$F_1 = f(SP_2), \tag{15}$$

$$SP_2 = f(Op_2), \tag{16}$$

$$OP_2 = f(T_{\mathrm{R}}, T_2), \tag{17}$$

where F denotes a feed, f denotes a molar flowrate, Q denotes the quench feed, r_{A} is a reaction rate expression, V_{RE} is the volume of the reactor, SP denotes valve stem position, T denotes temperature, and Op denotes an operator. Complex relationships involving these variables show only the corresponding functional dependence [e.g., relationships (9) through (17)]. The *structural matrix* constructed from this set of equations and relationships is shown in Fig. 18. Each numbered row describes the relationship between the variables that are contained in it. The variables are represented by the columns of the matrix. The occurrence of a variable in a particular relationship is signified by a nonzero entry (e.g., x in Fig. 18) at the intersection of the column representing the variable with the row representing the relationship.

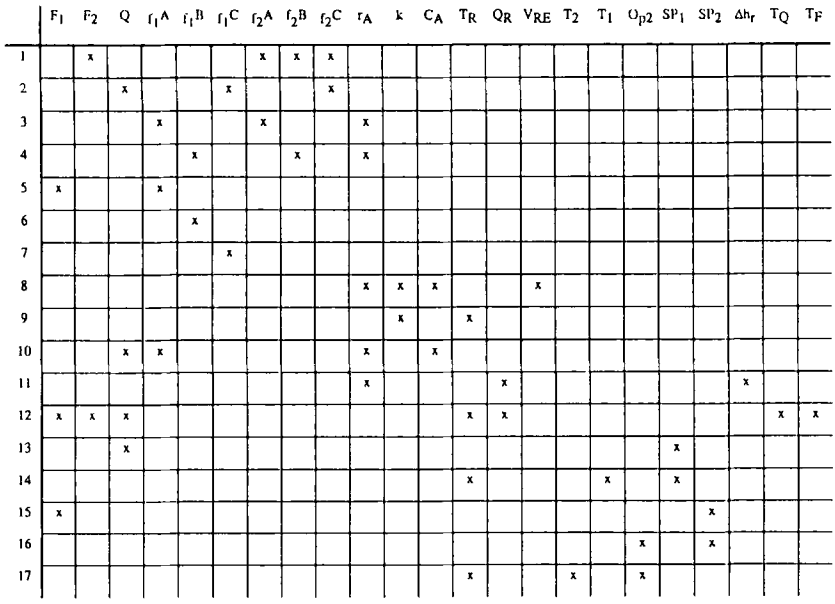

FIG. 18. The structural incidence matrix for the reactor-quench example.

The process of constructing the variable-influence pathway, which leads to the state preceding the top-level event, begins with the identification of the potential input variables. Following the procedures outlined in Section IV.B, the assignment of the input variables is initiated. But, the identification of the *input specifications* is intricately related to the scope of the assumed *system's boundary.* Thus, by restricting our attention to the reactor only (see shaded unit in Fig. 17), examination of the process modeling relationships identifies two system constants: Δh_r (the heat of reaction) and V_{RE} (the reactor volume); these become input variables in the structural matrix. Similarly, if the scope of the process' boundary is expanded to include external feeds, then the boundary of the system encloses all the shaded area of Fig. 19. This expansion identifies the invariant intensive properties describing the inlet feed (T_F) and the quench feed (T_Q) as external or input variables. Step 3 of the input assignment procedure (see Section IV.B) assigns the setpoints T_2 and T_1 as input variables, thus further expanding the scope of the process boundary (see Fig. 20).

Once inputs are assigned, output assignment begins using the method described in Section IV.B. Using this procedure, f_1^B is identified as taking its value from Eq. (6) and f_1^C is identified as taking its value from Eq. (7).

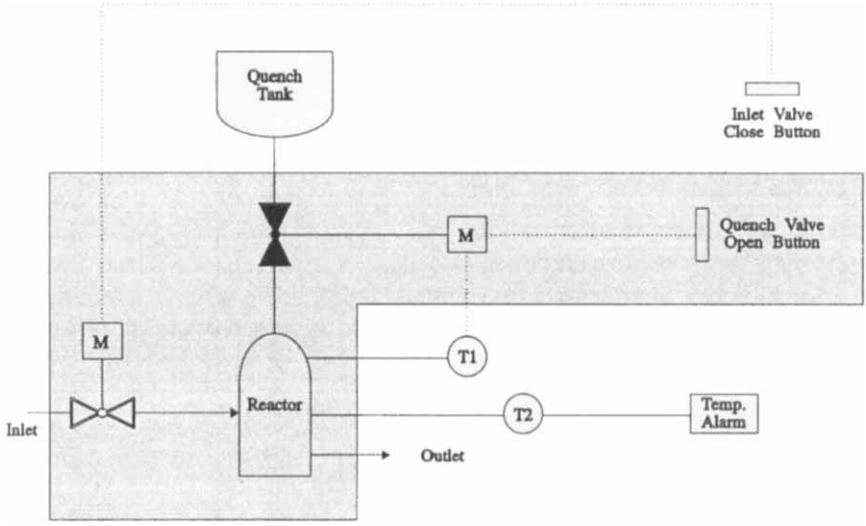

FIG. 19. The system boundary defining the expanded input specifications for the reactor-quench example.

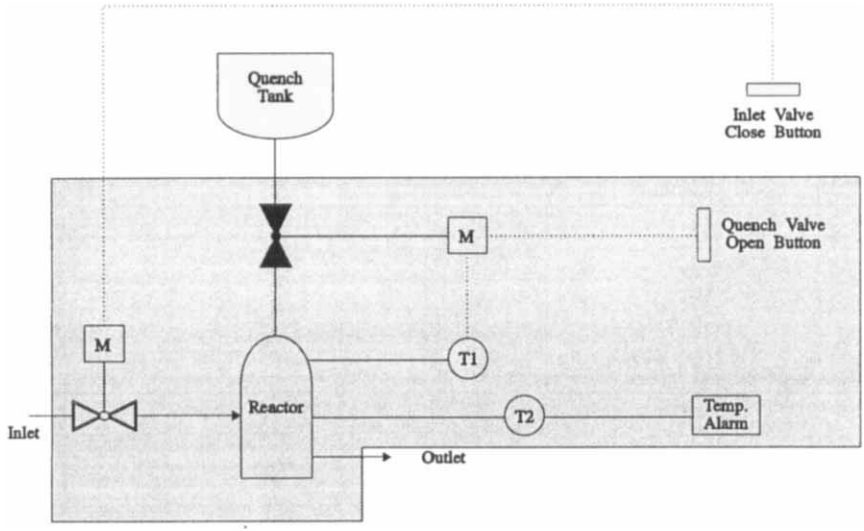

FIG. 20. The final system boundary indicating further expansion of the input specifications.

The respective rows and columns of the structural matrix are then eliminated. The value of the reaction rate constant can be given only by the definitional relationship (9), and thus it is assigned as output from Eq. (9). On elimination of the row and column corresponding to Eq. (9) and variable k, no other output assignments can be made. The block structure that results allows the remaining variables to take their output from any one of several equations.

To break the loop of cause-and-effect relationships, we make the unqualified assumption that Op_2, the variable denoting the human operator, obtains its value from relationship (17); i.e., the operator responds to the temperature in the reactor (T_R) and not vice versa. This assumption is then catalogued, as *Assumption-1*, so that it may be retracted at a later stage, as required. Such retraction allows the causality that emanates from the set of process equations to be modified. Once we have eliminated Eq. (17) and the column corresponding to variable Op_2, we can repeat the steps of the output set assignment (see Section IV.B), and find that F_1 obtains its value from Eq. (15). The set of input and output assignments, made so far, establish the causality between

- Reaction temperature and human operator
- Human operator and feed valve stem position
- Feed valve stem position and feed rate.

Elimination of the respective rows and columns identifies further that $f_1^{\wedge}$ takes its value from Eq. (5).

At the end of the preceding assignments, a block structure of "*simultaneous relationships*" still remains. Again, though, an unqualified assumption can be made regarding the causality of variables in Eq. (14). The assumption is that SP_1, *the variable describing the feed valve stem position, is driven by reactor temperature.* This assumption is catalogued and denoted as *Assumption*-2.

The variable-influence diagram resulting from the set of the two assumptions, i.e., *Assumption*-1 and *Assumption*-2, is shown in Fig. 21. This diagram makes explicit the pathways leading to T_R and illustrates how disturbances can propagate through the network of relationships and enable the occurrence of the TLE. If we now retract *Assumption*-1, and replace it by its opposite (which will be labeled *Assumption*-3), i.e., the operator does not react to the temperature of the reactor but *the value of the reactor temperature is defined by the actions of the operator*, then the variable-influence diagram changes and is now represented by that of Fig. 22. By collecting the entire set of graphs that describe the resulting set of cause-and-effect relationships, we can guarantee complete identification of pathways, leading to the TLE, within the scope of the modeling effort,

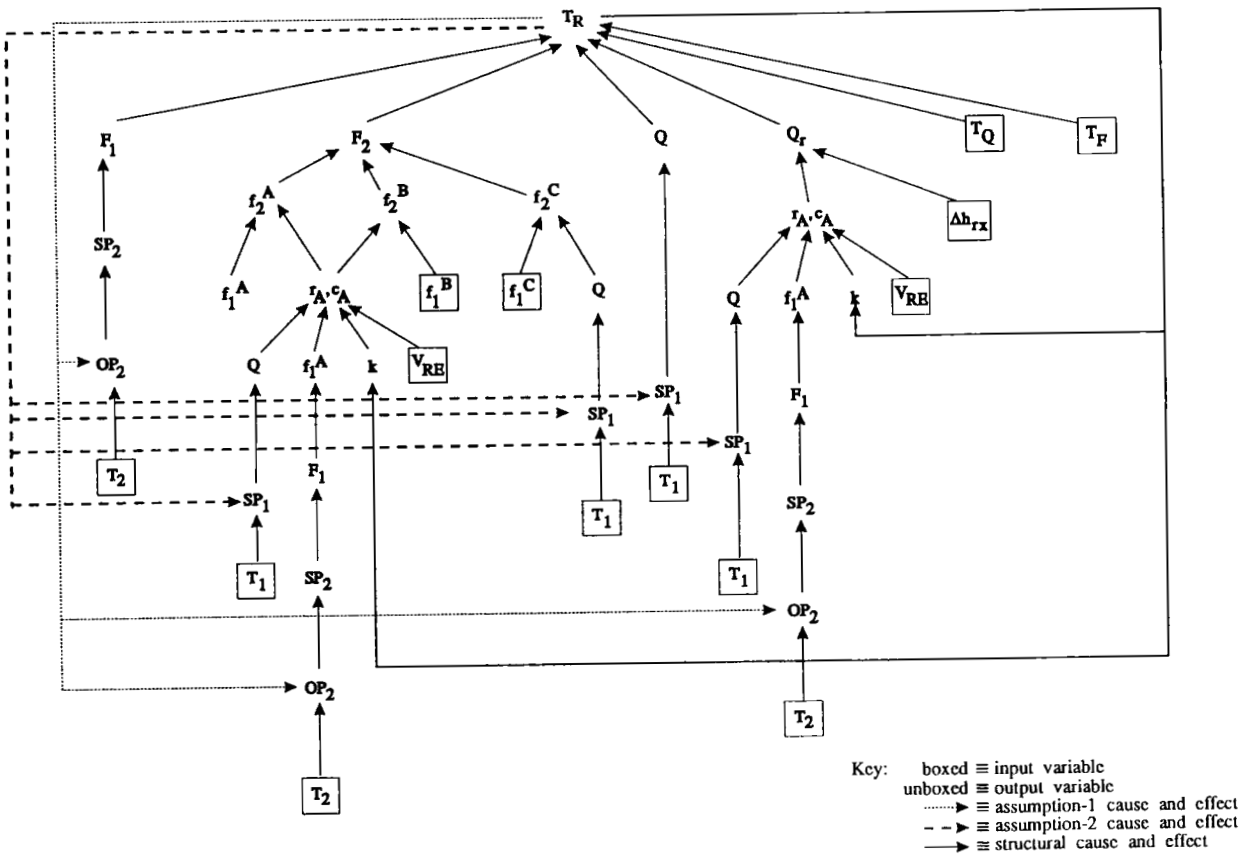

FIG. 21. The variable-influence diagram resulting from *Assumption-1* and *Assumption-2*.

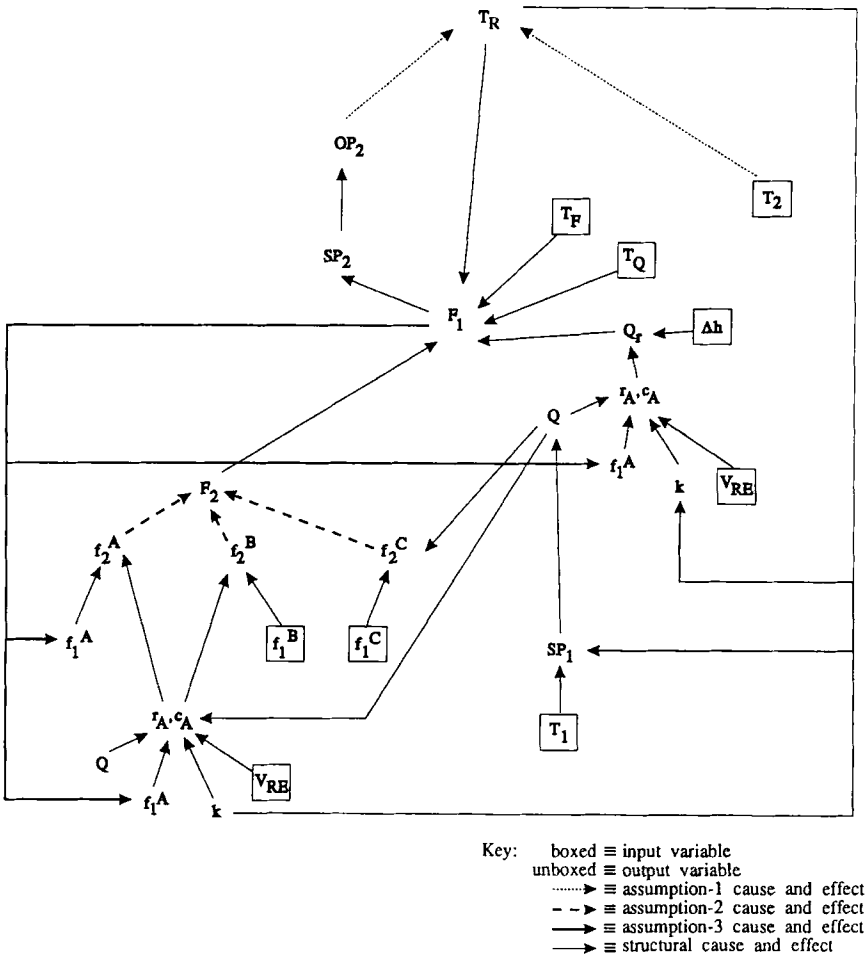

FIG. 22. The variable-influence diagram resulting from *Assumption-2* and *Assumption-3*.

in conjunction with the set of the assumptions made. By associating the variables, contained in the variable-influence diagram, with the appropriate types of hazards-preventing technologies (i.e., *Type-1*, *Type-2*, *Type-3 technologies*), the pathway of root causes is constructed. Figure 23 illustrates the association of the variables with specific *Type-1*, *Type-2*, and *Type-3 technologies*, for the case where the variable-influence diagram has been constructed assuming *Assumption-1* and *Assumption-2* to be true. Note that the heat of reaction Δh_r is the only Type-1 technology associated with the TLE. *This implies that a potential pathway for mitigating the*

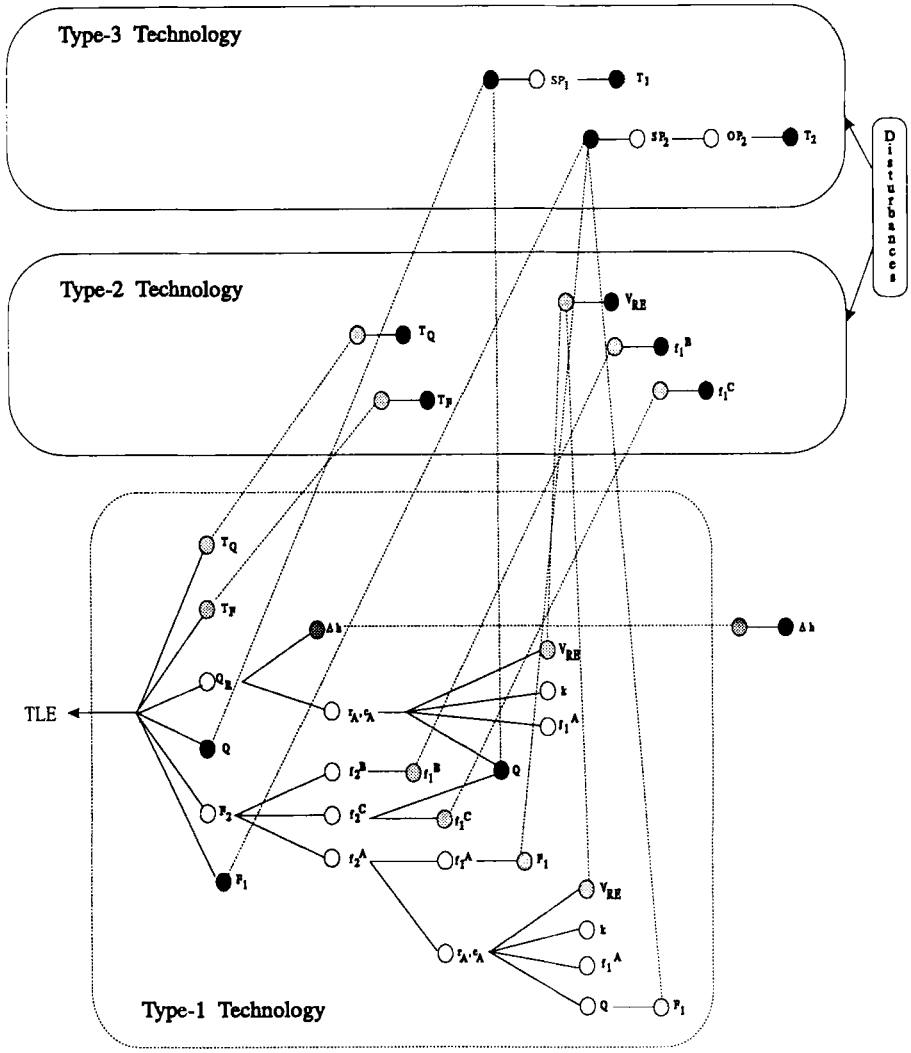

FIG. 23. Construction of root cause diagram.

TLE is to change the heat of reaction and obviously implies a change in chemistry. Furthermore, the variable-influence diagram makes explicit that top-level event prevention necessitates the placement of control objectives on every variable that immediately precedes the top-level, i.e., variables $T_Q, T_F, Q_R, Q, F_2, F_1$, should come under control.

Notice, also, that *the value of the variables described by Type-2 technologies are generally design specifications*; they affect the magnitude of the

top-level event. For example, V_{RE} specifies the volume of the reactor. Similarly, feed and quench temperatures vary the rate at which a TLE is achieved; likewise, so may f_1^B and f_1^C as the values that describe the molar flowrates of the product and the quench, respectively, contained in the feedstream, vary. This is particularly so if the reaction described is autocatalytic. Trace amounts of product in the feed can catalyze the reaction leading to a thermal disturbance that ultimately could trigger a reaction runaway.

Type-3 technologies require active control to mitigate a disturbance and consequently are protective systems associated with the process flowsheet. Notice that certain control actions cannot be modeled easily without overspecifying the system such as the manual override of the quench valve by the operator. For this reason, the constraint list, as described earlier, is associated with each piece of process equipment. This allows us to associate in an *a posteriori* manner additional control structures that are available to the process.

How the different technology types are used to identify potential root causes that can enable the top-level event is illustrated in Fig. 23. This illustration makes explicit the fact that disturbances, which enter into the system, can enter only through the inputs identified by Type-2 and Type-3 technologies. These disturbances can either be known a priori (*foreseen*) or not (*unforeseen*). Modeled disturbances can affect the process only through its input variables and are therefore identified by the pathways leading to the top-level event. Unmodeled disturbances can act in one of two ways: they can affect the value of a variable through an unmodeled relationship, or they can change the casuality described by the pathway, negating the assumption set from which it was built. However, since various assumption sets are used to construct the alternative pathways leading to the top level event, the impact of unmodeled disturbances that enabled some pathways can be contained. Furthermore, because of the manner in which the top-level event has been identified, it can be enabled only through the set of variables that describe the state immediately preceding it. Since any disturbance must eventually pass through that state, if it is to enable the top-level event, control objectives designed around these variables can mitigate foreseen and unforeseen disturbances that lead to a top-level event. Using the knowledge associated with different technology types, and the pathways that lead to a top-level event, we can construct a topological fault tree. This is illustrated in Fig. 24. *Note the need for a special gate.* This type of gate is required because without quantitative assessment, it cannot be determined whether the gate is a single *and-gate*, a single *or-gate*, or a structure containing *and-gates* and *or-gates*.

The topological fault tree shown in Fig. 24 suggests that the top-level event can be enabled through a change in the input of T_Q (e.g., this could imply no quench). Using the algorithm presented in Section IV.C, the

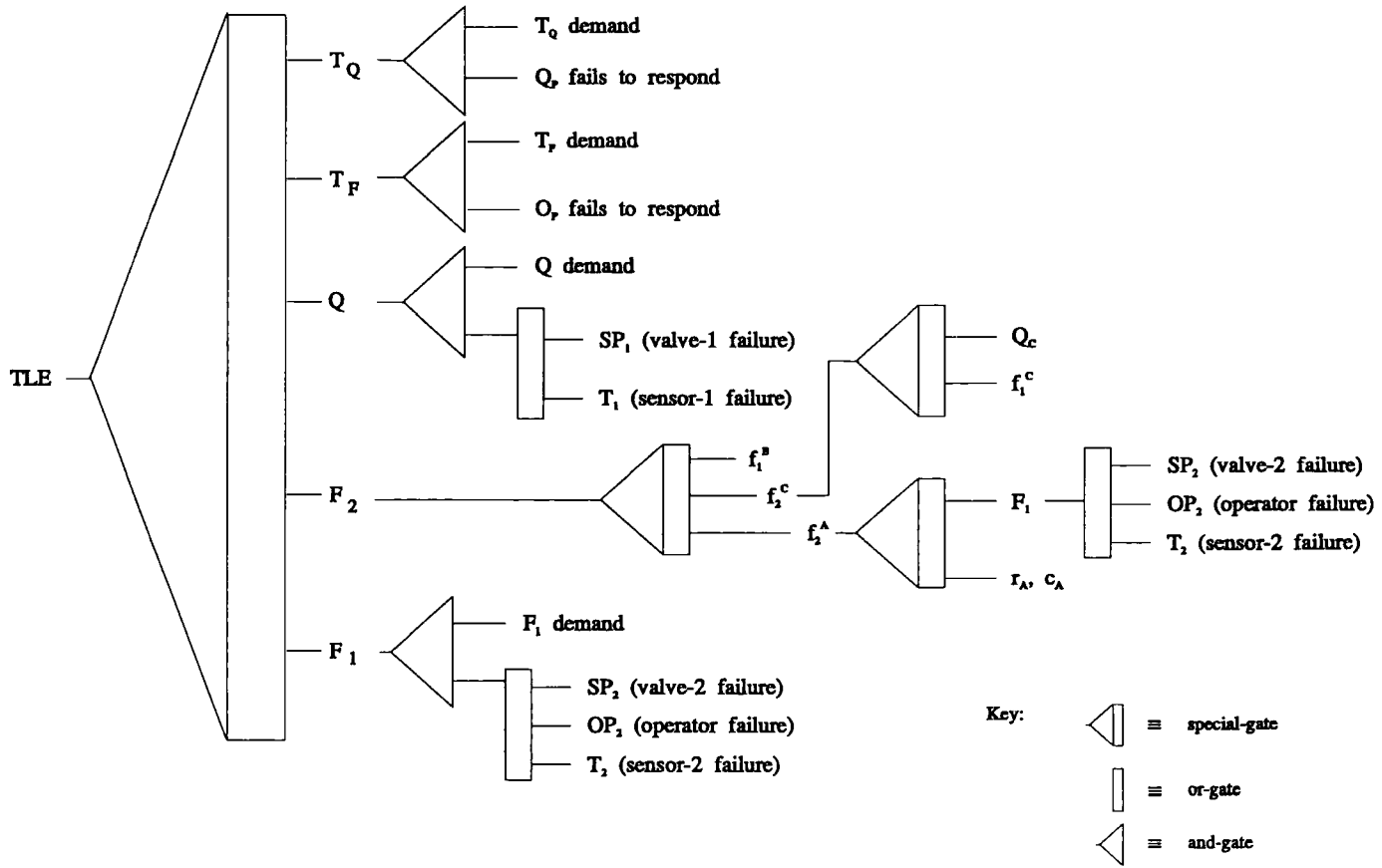

FIG. 24. The topological fault tree resulting from *Assumption-1* and *Assumption-2*.

logical gate constructed from this input variable is an *and-gate*, the result of a controlled variable involving an unbranched node. T_F, feed temperature, is the next variable with a pathway to a top-level event. Similarly, an *and-gate* is constructed for feed temperature, T_F, representing a thermal deviation demand in the feed and the failure of the operator to notice the upset, or take the required action. Quench flow rate, Q, also has a pathway to the top-level event (see Fig. 23). Following the algorithm proposed, construction of logical gates associated with this pathway illustrates that a demand by Q can be enabled in one of two ways: (1) the stem position of valve 1 fails to obtain the necessary position (i.e., the valve fails to close), or (2) the sensor setpoint T_1 is in error.

Unlike T_2, T_F, and Q, the outlet feed, F_2, requires a special gate to model the logical consequences of root causes passing through it. This gate takes as its input f_2^B, f_2^C, and f_2^A. Figure 23 illustrates that f_2^B obtains its input directly from f_1^B, a Type-2 technology. Consequently, any disturbance in f_1^B could potentially enable F_2. Since f_2^C and f_2^A are branched nodes, they in turn require construction of special-gates. The special-gate constructed from f_2^C takes as its input Q, and f_1^C. Similar to f_1^B, f_1^C is a Type-2 technology that takes its input directly from the

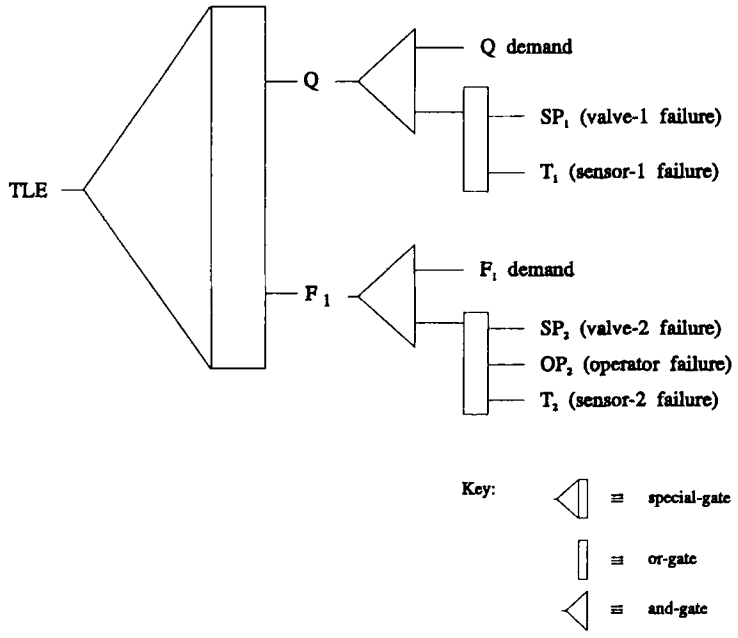

FIG. 25. The topological fault tree resulting from *Assumption-2* and *Assumption-3*.

external world. Therefore, any disturbance in the external surroundings that can lead to a change in the value of f_1^C could potentially also enable the top-level event.

Variable f_2^A, which is a branched node, also requires the construction of a special-gate. This gate requires as its input F_1 and $\{r_A, C_A\}$. Although the expansion for this node is not shown, what is important to recognize is the fact that with the exception of V_{RE}, each of the inputs leading to this set have been covered previously. The implications of changing V_{RE}, a Type-2 technology, is that a change in reactor volume could enable a top-level event. Although the mechanism for this process has not been modeled, it could be the result of an improper design or the accumulation of material internal to the reactor. *What is important is that the effect of V_{RE} has been made explicit.*

The final variable with a pathway leading to the top-level event is F_1, an unbranched node. Since F_1 has associated with it a Type-3 technology as its protective system, a demand on F_1 can be mitigated by the proper positioning of the stem controlling valve 2, proper action by the operator, and a proper setpoint (i.e., T_2) on the temperature controller. A con-

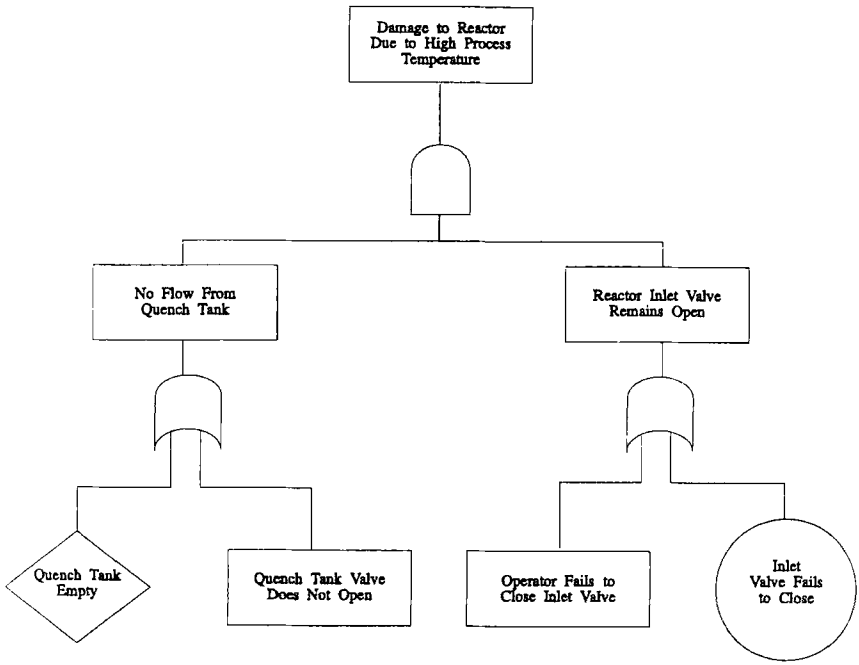

FIG. 26. The industrial implementation of a fault tree for the reactor-quench example.

densed version of the topological fault tree shown in Fig. 24, involving only Type-3 technologies, is shown in Fig. 25. It identifies two principal pathways to the top-level event: Q and F_1. Notice that a special-gate is still needed to connect these inputs to the output, T_R. The necessity of the special-gate will remain until *quantitative* analysis can determine whether the top-level event can be enabled by either Q or F_1, or whether both are required. This depends on the dynamics of the system: reaction rates, time requirements for runaway, heat release rates, heat removal rates, reactor volume, quench rates, etc.

The fault tree cited in literature for this process is shown in Fig. 26 (Battelle, 1985). Notice the similarity between Figs. 26 and 25, particularly in the structure of the two trees, and recognize that as a result of quantitative analysis, Fig. 26 has an *and-gate* as its top-level gate. More importantly, recognize that without complete quantification of the root causes, the fault tree given in Figure 26 may be incomplete.

V. Conclusion

By using the domain-specific modeling languages, LCR and MODEL.LA., we have developed a process-based methodology to identify potential hazards in a chemical process and generate mechanisms for the prevention or mitigation of their effects, through the identification and correction of inherent design weaknesses, or the adaptation of the operating procedures. The methodology is based on an interplay of inductive and deductive reasoning. *Inductive* reasoning has been used in order to identify (1) all potential chemical reactions, which could lead to a hazard, and (2) the requisite conditions that would enable the occurrence of these reactions. *Deductive* reasoning has been used to convert the enabling conditions needed for a reaction into process design or operational "faults" that would constitute the causes for the occurrence of a hazard. The methodology is more efficient, complete, and cost-effective than are current hazard analysis approaches. It contributes three important achievements: (1) formalization of the hazards identification problem, (2) systemization of hazard analysis through all phases of the design process, and (3) construction of a methodology that completely identifies all potential hazards within the scope of the modeling efforts. Furthermore, it establishes a formal strategy for the integration of safety into a design technology at any point in the design process and provides a means for discriminating among design alternatives with respect to disturbance mitigation. Finally, the methodology also provides the basis for the

optimization of a design technology with respect to the parameters that describe its inherent safety.

References

Abelson, H., Sussman, G. J., and Sussman, J., "Structure and Interpretation of Computer Programs." MIT Press, Cambridge, MA, 1985.

Atallah, S., Assessing and managing industrial risk. *Chem. Eng.*, September, p. 8 (1980).

Batstone, R., *in* "Proceedings of the International Symposium on Preventing Major Chemical Accidents," February, p. 5.126. AIChE, Washington, DC, 1987.

Battelle, "Guidelines for Hazard Evaluation Procedures." AIChE Press, Washington, DC, 1985.

Boykin, R. F., and Kazarians, M., Quantitative risk assessment for chemical operations. *In* "Proceedings of the International Symposium on Preventing Major Chemical Accidents," February, p. 1.87. AIChE, Washington, DC, 1987.

Brachman, R. J., and Levesque, H. J., "Readings in Knowledge Representation." Morgan Kaufmann Publishers, Los Altos, CA, 1985.

Brannegan, D. P., "Hazards Evaluation in Process Development," Chemical Process Hazards Review, p. 18. American Chemical Society, Washington, DC, 1985.

Bretherick, L., "Bretherick's Handbook of Reactive Chemical Hazards." Butterworth, London, 1990.

Carson and Mumford, Analysis of incidents involving major hazards in the chemical industry. *J. Hazard. Mat.* **3**, 149 (1979).

Cormen, T. H., Leiserson, C. E., and Rivest, R. L., "Introduction to Algorithms." MIT Press, Cambridge, MA, 1990.

Cox, R. A., An overview of hazard analysis. *In* "Proceedings of the International Symposium on Preventing Major Chemical Accidents," February, p. 1.37. AIChE, Washington, DC, 1987.

Culbertson, T. L., and Searson, A. H., "Exxon Facility Design Assessment and Control of Hazards," Exxon internal publication. Exxon, 1983.

Dale, S. E., Cost effective design considerations for safer chemical plants. *In* "Proceedings of the International Symposium on Preventing Major Chemical Accidents," p. 3.79. AIChE, Washington, DC, 1987.

de Groot, J. J., and Van der Elst, F. H., Thermal properties of peroxides. *Inst. Chem. Eng. Symp. Ser.* **68**, 3/ V:1 (1981).

Haastrup, P., Design errors in the chemical industry. *Inst. Chem. Eng. Symp. Ser.* **80**, J15 (1983).

Hendrikson, J. B., *J. Am. Chem. Soc.* **108**, 6748 (1986).

Hoffmann, J. M., Chemical process hazard review. *Am. Chem. Soc.*, p. 1 (1985).

ICI, "Process Safety," Course Notes. ICI, 1988.

Kletz, T. A., Make plants inherently safe. *Hydrocarbon Process.*, September, p. 72 (1985).

Kritikos, T., A model for process design automation. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA (1991).

Lakshmanan, R., and Stephanopoulos, G., Synthesis of operating procedures for complete chemical plants. *Comput. Chem. Eng.* **12**, 985 (1988).

Lees, F. P., "Loss Prevention in the Process Industries." Butterworth, London, 1980.

Lees, F. P., Hazards warning structure: Some implication and applications. *Inst. Chem. Eng. Symp. Ser.* **80**, J1 (1983).

Lowe, D. R., and Solomon, C. H., Hazards identification procedures. *Inst. Chem. Eng. Symp. Ser.* **80**, G8 (1983).

Maher, M. L., *in* "Expert Systems in Engineering" (D. T. Phan, ed.). IFS Publications/ Springer-Verlag, Berlin, 1988.

Mosleh, A., Bier, V. M., and Apostolakis, G., A critique of current practice for the use of expert opinion in probabilistic risk assessment. *Reliab. Eng. Syst. Saf.* **20**, 63 (1988).

Nagel, C. J., Identification of hazards in chemical process systems. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA (1991).

Ozog, H., Hazard identification analysis and control. *Chem. Eng.* (*N.Y.*). February 18, p. 161 (1987).

Ozog, H., and Bendixen, L. M., Hazard identification and quantification. *Chem. Eng. Prog.*, April, p. 55 (1987).

Perkins, J. D., and Barton, G. W., Modelling and simulation in process operation. *In* "Foundations of Computer-Aided Process Operations" (G. V. Reklaitis and H. D. Spriggs, eds.). CACHE Corp., Austin, TX, and Elsevier, New York, 1987.

Sheil, B., Power tools for programming. *Datamation*, February, p. 131 (1983).

Sheil, B., The artificial intelligence tool box. *In* "Artificial Intelligence Applications in Business" (W. Reitman, ed.), p. 113. 1984.

Slater, C., and Pitblado, "Major Industrial Hazards Project Report." The Warren Centre for Advanced Engineering, University of Sidney, Sidney, Australia, 1987.

Sriram, D., and Maher, M. L., *in* "Applications of Artificial Intelligence in Engineering Problems" (D. Sriram and R. Adey, eds.), Vol. 1. Southhampton University, UK 1986.

Stephanopoulos, G., The future of expert systems. *Chem. Eng. Prog.*, September, p. 44 (1987).

Stephanopoulos, G., Johnston, J., Kritikos, T., Lakshmanan, R., Mavrovouniotis, M., and Siletti, C., Design-kit: An object-oriented environment for process engineering. *Comput. Chem. Eng.* **11**, 655 (1987).

Stephanopoulos, G., Johnston, J., and Lakshmanan, R., An intelligent system for planning plant-wide process control strategies. *Journal A* **29**(3), 81 (1988).

Stephanopoulos, G., Henning, G., and Leone, H., MODEL.LA. A modeling language for process engineering. Part I. *Comput. Chem. Eng.* **14**, 813 (1990a).

Stephanopoulos, G., Henning, G., and Leone, H., MODEL.LA. A modeling language for process engineering. Part II. *Comput. Chem. Eng.* **14**, 847 (1990b).

Stoessel, F., Experimental study of thermal hazards during the hydrogenation of aromatic nitro compounds. *Proc. Int. Symp. Loss Prev. Saf. Promot. Process Ind., 6th*, p. 77-1 (1989).

Walling, C., "Free Radicals in Solution." Wiley, New York, 1957.