Model-Based Approach to Automated Hazard Identification of Chemical Plants

Catherine A. Catino and Lyle H. Ungar
Dept. of Chemical Engineering, University of Pennsylvania, Philadelphia, PA 19104

A prototype hazard identification system, qualitative hazard identifier (QHI), works by exhaustively positing possible faults, automatically building qualitative process models, simulating them, and checking for hazards. QHI matches a library of general faults such as leaks, broken filters, blocked pipes, and controller failures against the physical description of the plant to determine all specific instances of faults that can occur in the plant. Faults may perturb variables in the original design model or may require building a new model. Fault models are automatically generated using the qualitative process compiler and simulated using QSIM. Hazards including overpressure, overtemperature, controller saturation, and explosion are identified in the reactor section of a nitric acid plant using QHI.

Introduction

Automatic generation of chemical plant models and the subsequent simulation of these models can be used as the basis of design, fault diagnosis, and safety and loss prevention systems. Safety and loss prevention, an integral part of the design process, includes identification and assessment of hazards and development of methods to control the extent of a hazard (such as containment of toxic materials) or to limit the likelihood of an undesirable event occurring (such as automatic control systems with alarms) (Coulson and Richardson, 1983). When a hazard or fault occurs in a plant, the model describing the plant may no longer be applicable, in which case, a new model must be built to describe the modified operating conditions. Automatic model builders can construct new models based on the modified conditions.

In this article, a method for automatically building models of chemical plants is used in an algorithm to identify potential process hazards. In this approach to modeling, the computer chooses the relevant physics and chemistry from a predefined library of general phenomena and builds the appropriate model for a given physical system. To describe the full range of behaviors of both normal and faulty chemical process units we use a library of fundamental physical and chemical phenomena such as heat and mass transfer, chemical reaction, and phase equilibrium, not a library of standard pieces of equipment as used in traditional modeling packages such as FLOWTRAN (Rosen and Pauls, 1977; Seader et al., 1987) and ASPEN (Evans et al., 1979).

To automatically generate a mathematical model, two inputs are required as shown in Figure 1. The first is a library of physical and chemical phenomena. Each phenomenon definition in the library consists of a set of preconditions (such as pressure source > pressure destination) required for that particular phenomenon (such as mass flow) to apply and the equations contributed to the model by this phenomenon if it is applicable. The second input is the physical description of

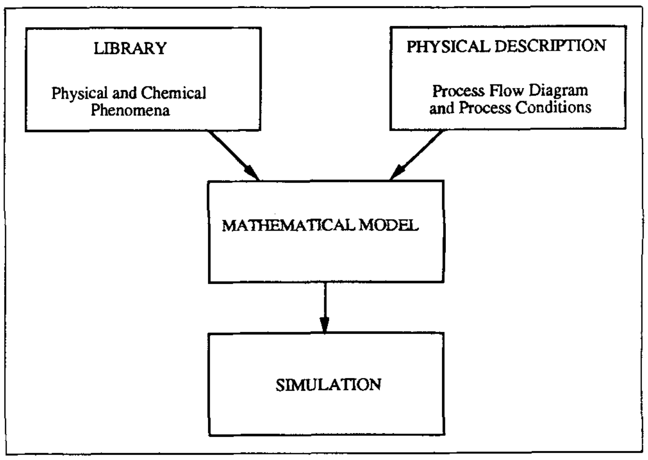
LIBRARY
Physical and Chemical Phenomena
PHYSICAL DESCRIPTION
Process Flow Diagram and Process Conditions
MATHEMATICAL MODEL
SIMULATION

Figure 1. Automatic model generation.

Correspondence concerning this article should be addressed to L. H. Ungar.

the plant including the substances and equipment present, their connectivity, and assumptions about operating conditions. If the preconditions for a particular phenomenon are satisfied by the physical description of the plant, then the equations describing that phenomenon become part of the mathematical model. This process-centered approach to model building provides flexibility, because as long as its preconditions are satisfied, a phenomenon can occur anywhere in the plant, not necessarily just inside a unit in which it is expected to occur. Reaction, for example, is not confined to occur solely in a reactor. Similarly, a fault such as a leak can be defined once, and all possible leak sites can be found automatically using pattern matching.

The procedure of automatic model construction ensures that if an operating limit is reached or if a fault occurs causing the model assumptions to change, a new model will be created. We implement this modeling approach using the qualitative process compiler (QPC) (Crawford et al., 1990; Farquhar, 1993), which generates a set of equations that are abstractions of ordinary differential equations. These qualitative equations are then sent to the QSIM algorithm for qualitative simulation (Kuipers, 1986) and the possible behaviors (qualitative solutions of the model equations) are determined.

Qualitative models were chosen for this study, because they allow predictions to be made about the behavior of the system in the absence of exact quantitative information. Many of the current techniques for identifying hazards are inherently qualitative. The engineers conducting the study attempt to answer a series of what-if type questions about the design: what if the pipe connecting the evaporator to the reactor becomes clogged or what if the inlet temperature is increased? Note that neither of these questions involves numerical values. In order to answer these what-if questions, engineers mentally perform a simulation to determine the consequences of these disturbances. If appropriate quantitative fault models were available, they could be used similarly.

In this article, a prototype hazard identification system, the qualitative hazard identifier (QHI), is introduced which combines the automatic model building features of QPC with the qualitative simulation capabilities of QSIM. QHI identifies possible hazards from faults in chemical plants by exhaustively examining the ways in which a library of faults can occur in the plants. When faults force the model to change or when another chemical plant must be analyzed, QHI builds a new model based on the modified operating conditions and assumptions. Consequences of faults are predicted by qualitative simulation. Thus, QHI works by exhaustively positing possible faults, simulating them, and checking for hazards.

Hazards incorporated in the current implementation include controller saturation, overtemperature, overpressure, catalyst poisoning, and explosion. QHI can evaluate the effects of a number of different faults including leaks, broken filters, partially and completely blocked filters and pipes, and controller failures leading to a valve failing partially or completely closed or to a valve failing open. Faults may perturb variables in the original design model or may require building a new model. An important advantage of the architecture is that all tasks remain decomposed: QPC builds models, QHI applies a set of rules to either modify the physical description or create a disturbance function, QSIM simulates behavior, and QHI interprets behavior.

This article introduces hazard and operability studies, a technique currently used to identify hazards and to explain how the approach used in QHI differs. It also describes how QSIM and QPC work, and details the QHI algorithm. Two case studies using QHI to identify hazards in the reactor section of a nitric acid process are provided, and QHI is compared with other computer aids for hazard evaluation.

Hazard Evaluation Methods

Several techniques exist for evaluating potential process hazards in chemical plants. They include what-if analysis, checklists, fault trees, event trees, failure modes and effects analysis, hazard and operability studies, and so on. These and other methods are described in detail by Lees (1980), Crowl and Louvar (1990), Greenberg and Cramer (1991), and in the AIChE Center for Chemical Process Safety (CCPS) book *Guidelines for Hazards Evaluation Procedures* (1985). The hazard and operability (HAZOP) study is one of the most widely applied of these methods (Chemical Industries Association, 1977; Lawley, 1974; Kletz, 1985).

Originally developed by the Petrochemicals Division of Imperial Chemical Industries (ICI), a HAZOP study is a systematic, methodical procedure for hazard identification. Using design documents describing a facility and a list of guide words, a multidisciplinary team of engineers studies the process flowsheet line-by-line then unit-by-unit using guide words to help generate a list of possible deviations from the intended design conditions that could lead to hazardous events. The possible causes (faults) and consequences (hazards) of these deviations are then identified. Without computer support, a typical HAZOP study of a large process can take many weeks to complete.

A key point in automating any hazard evaluation procedure is deciding where to start in the hazard sequence. Figure 2 shows the pieces of a hazard sequence: an initiating event, the actual fault or malfunction, occurs causing intermediate deviations in process parameters resulting in one or more hazards. For example, in the nitric acid process shown in Figure 3, a clogged air filter leads to the deviation no flow which leads to the hazard high ammonia concentration as ammonia may reach its explosive limit. When engineers perform HAZOP studies,

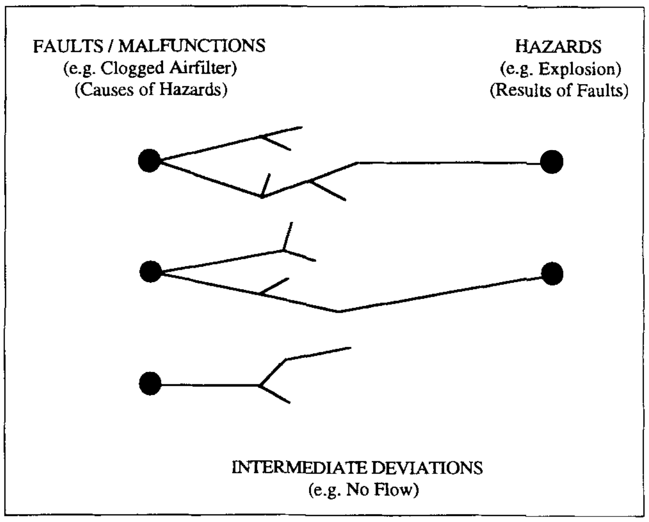

Figure 2. Hazard sequence.

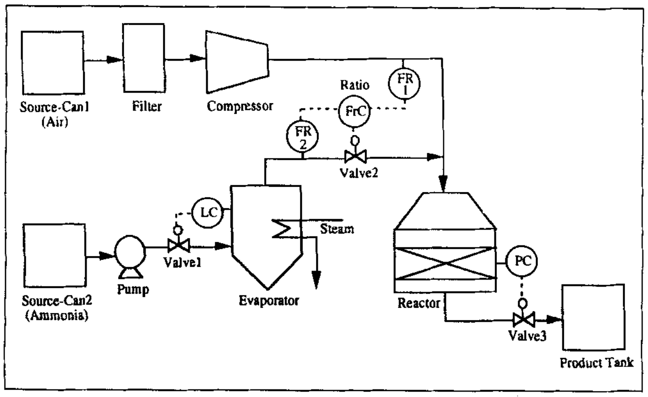

Figure 3. Reactor section of nitric acid plant.
Modified from Coulson and Richardson (1983).

they begin with a deviation such as no flow and reason forward to determine hazards and backward to determine faults, because they are confident that they can generate a complete list of deviations and thus a complete list of faults and hazards. This approach offers advantages in that it can help engineers think of faults that they may not have considered. Also, when different faults cause the same deviation, one does not need to analyze the resulting hazards more than once.

In contrast, the QHI algorithm begins with a fault and simulates forward to determine both intermediate deviations and hazards. This approach requires a library of different classes of faults and "rules" describing the situation in which each may occur. For example, if X is a valve, then it may fail shut or leak. The consequences of faults are systematically explored by (1) finding each location in the plant where each fault type in the fault library could occur, (2) altering the plant model to include the fault, (3) simulating, and (4) checking to see if hazards result. The advantage of altering the model and simulating is that one can detect hazards that may result from complex interactions far downstream from the original fault. One disadvantage of this approach is that one cannot conduct the HAZOP by starting with deviations, since the simulations require having a plant model. As long as the library of faults is complete, starting with a fault is more efficient because only one simulation is required; whereas, starting with an intermediate deviation requires both a simulation to determine the hazards resulting from the deviations and a fault diagnosis system to determine the faults leading to the deviations.

QSIM and QPC Algorithms

The qualitative hazard identifier is based on the ability to automatically build and solve models of chemical plants even at a design stage where exact numerical models and parameter values are not available. To do this, we use QPC and QSIM to build and solve qualitative plant models. This section gives a brief introduction to strengths and weaknesses of this qualitative modeling approach.

The processes that occur in chemical plants, such as reaction and heat transfer, can be described by a set of algebraic and differential equations. To build and solve this set of equations, engineers must specify exact parameter values and mathematical relationships. Depending on the stage of process development, this information may be only partially known or the equations may be highly coupled and nonlinear making them difficult to formulate or solve, even with the help of computers. Yet solutions to these equations are useful for design and safety and loss prevention. The QSIM algorithm provides a framework in which to specify incomplete knowledge about a system in terms of qualitative differential equations (QDEs) and still be able to make predictions about the possible behaviors of the system. The QDEs are abstractions of ordinary differential equations such that infinitely many linear and nonlinear ODEs reduce to the same QDE.

In order to construct a QSIM QDE model, the model builder must specify both the variables of the physical system and the constraints relating variables to each other. Each variable has a quantity space, an ordered set of landmark values specifying the range of values that the variable can have. Landmarks of a variable may include zero, positive infinity, negative infinity, and all qualitative critical values, such as the quantity space of temperature, which may include boiling point and dew point. The QSIM algorithm discovers new critical values as the simulation proceeds, such as noting the point at which a temperature reaches its maximum value, and thus adds new landmarks to the quantity space. At each time point in a simulation, each variable has a qualitative state characterized by its magnitude with respect to the landmarks and its direction of change (increasing, decreasing, or steady). For example, if a variable is negative [on the range $(-\infty\ 0)$] and increasing (inc), then its qualitative state is $[(-\infty\ 0)$ inc].

The constraints consist of qualitative versions of mathematical relationships such as addition, multiplication, and differentiation, along with other constraints asserting that some functional relationship exists between two parameters but only specifying whether the functionality is monotonically increasing ($M+$) or decreasing ($M-$). It is the $M+$ and $M-$ constraints that provide QSIM with the ability to simulate in the absence of complete knowledge. For example, reaction rate increases with reactant concentration irrespective of the order of the reaction.

The model builder may also specify transition information to establish the limits of applicability of a QDE model. Transitions are usually specified in terms of a variable reaching a certain landmark value. For example, when a liquid is heated, it may reach its boiling point. If this critical value is reached and another QDE model is specified as being applicable beyond this boundary, the simulation will continue with the new QDE model. Both models, before and after boiling, must be specified *a priori* by the model builder.

Once a QDE model has been formulated, an initial state is defined by asserting values for certain parameters and letting QSIM complete the initial state by propagating variable values through the constraints. The initial state does not necessarily have to be a steady state; for example, we may be interested in modeling flow through a tank starting with an empty tank in which case the mass in the tank is zero and increasing in the initial state. For the purposes of this research, however, an initial steady state is specified along with a disturbance defining a step change in one of the model parameters. For example, modeling flow through a tank starts with steady-state flow through the tank, and then QSIM may be asked to determine the possible behaviors when the inlet flow rate is increased.

Given the QDE, initial state specifications, and a disturbance function, QSIM identifies the set of possible successor states consistent with the model constraints. If QSIM is unable to reduce either a variable value or its direction of change (derivative) to one result, then all possible values are retained and the state has multiple successors. This procedure of generating and filtering successor states continues until all variables in all states have steady derivatives or until the limit of computer memory is reached, whichever occurs first. A branching tree of states results, and a path through the tree constitutes a behavior of the system. The generation of multiple behaviors vs. one numerical solution is one of the biggest differences between qualitative and quantitative simulation. QSIM has been used to model a wide variety of physical systems including mass and spring systems (Lee and Kuipers, 1988), physiological mechanisms (Kuipers, 1987), and some chemical processes such as reaction and flow through mixing tanks (Dalle Molle et al., 1988; Dalle Molle, 1989). Problems with intractable branching where infinitely many behaviors are produced have limited the complexity and type of models built.

One of the main causes for the proliferation of behaviors is that QSIM is unable to filter all spurious or aphysical behaviors. Spurious behaviors do not correspond to any physically possible solution of the model, and they occur due to the abstractions made in moving from a quantitative to a qualitative model. The QSIM algorithm is guaranteed to generate all behaviors exhibited by the original ODE from which the QDE was derived (Kuipers, 1986). Thus, if QSIM predicts only one behavior, then that behavior is guaranteed to be the correct solution. However, since many different ODEs result in the same QDE, it is possible that in addition to the set of actual behaviors, QSIM will generate spurious behaviors corresponding to no possible solution of the particular situation modeled, but instead describe a slightly different ODE. Much of the recent research associated with QSIM has focused on ways to reduce the number of spurious predictions including higher-order derivative constraints (Kuipers and Chiu, 1987; Kuipers et al., 1991), phase space representations (Lee and Kuipers, 1988), energy constraints (Fouche and Kuipers, 1992), and mixed qualitative-quantitative simulation (Kuipers and Berleant, 1988).

The QPC algorithm (Crawford et al, 1990; Farquhar, 1993) automatically generates the set of qualitative differential equations describing a system. The algorithm decomposes qualitative reasoning into two tasks, a model building task that creates a QDE model, and a qualitative simulation task that predicts possible behaviors resulting from this model. Illustrated in Figure 4, the QPC algorithm for model building and simulation requires two inputs. The user must supply a library of physical and chemical phenomena. The definition of each phenomenon in this library consists of the preconditions necessary for the phenomenon to apply, and the equations contributed to the model if the preconditions are satisfied. For example, the phenomenon mass flow requires the pressure of one vessel to be greater than the pressure of a connected vessel, and if applicable, mass flow contributes an equation describing the flow rate between the two vessels.

The second input to QPC is the physical description of the plant, a symbolic representation of the substances and equipment present, their connectivity, and assumptions about operating conditions. The physical description of the plant is

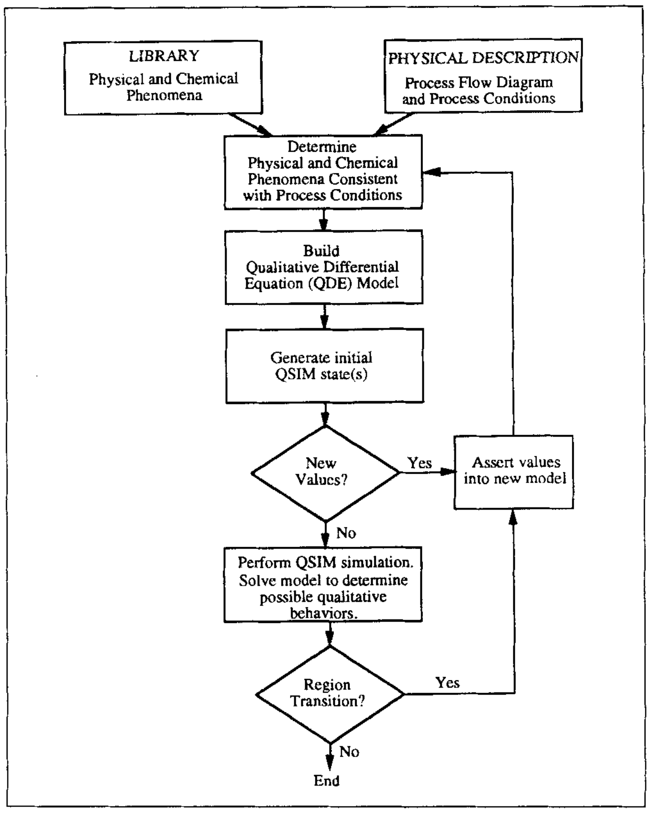
LIBRARY
Physical and Chemical Phenomena
PHYSICAL DESCRIPTION
Process Flow Diagram and Process Conditions
Determine Physical and Chemical Phenomena Consistent with Process Conditions
Build Qualitative Differential Equation (QDE) Model
Generate initial QSIM state(s)
New Values?
Yes
Assert values into new model
No
Perform QSIM simulation. Solve model to determine possible qualitative behaviors.
Region Transition?
Yes
No
End

Figure 4. Model building and simulation using the qualitative process compiler (QPC).

matched against the preconditions of the general phenomenon definitions such as mass flow and chemical reaction. When the preconditions of a process are satisfied, a specified instance of the phenomenon, such as mass flow of water between the reactor and product tank, is identified. This specific phenomenon is said to be *active*, and the equations associated with it become part of the mathematical model, such as an equation describing the flow rate between reactor and product tank is included in the model. The equations from all active phenomena form the model of the entire plant.

After determining the specific instances of phenomena consistent with the physical description and building a QDE model consisting of the equations contributed by these active phenomena, QPC forms an initial QSIM state to specify a qualitative initial value problem. It is possible that in forming the initial state, new variable values will be identified that will activate other instances of phenomena. If this happens, the model must be rebuilt before the simulation can proceed. Once the precise model and initial QSIM state have been generated, a QSIM simulation is performed and the possible qualitative behaviors are identified. If a behavior crosses a region transition, such as water reaches its boiling point, QPC returns to the model building portion of its algorithm so that a new model can be created describing the physics in the next region.

QPC's approach to model building is incremental so that assumptions made and entities present can change as more information is gained from active phenomena. Phenomena themselves can contribute operating assumptions not necessarily specified *a priori* by the model builder in the physical description, and these additional assumptions may cause instances of other phenomena (fluid flow into a vessel can cause a new substance to appear in that vessel, creating a new vapor-liquid equilibrium). Therefore, not only does QPC handle incomplete knowledge about system parameters and functional relationships as qualitative simulators do, but it also handles incomplete knowledge about the entities that exist, the operating conditions that hold, and the physical and chemical phenomena that model the particular physical situation.

A library of physical and chemical phenomena in the QPC representation was developed (Catino, 1993) in which chemical plants are described as a series of containers (vessels that hold material) and the connections (pipes and valves) between them. Containers hold phases which can exist as gases, liquids, or solids, and phases consist of components such as air and ammonia. Other entities incorporated in the library include transport-devices (pumps, compressors) and controllers (pressure, level, and ratio controllers). QPC is a frame-based representation, with inheritance. Liquids, for instance, have certain properties (such as boiling point) different from gases (such as dew point), but both inherit properties associated with phases (such as mass) since both are types of phases.

Qualitative Hazard Identifier (QHI) Algorithm

The QHI algorithm combines the automatic model building features of QPC with the qualitative simulation capabilities of QSIM to identify hazards in chemical plants. Unlike a HAZOP study performed by a team of engineers, where the starting point is an intermediate deviation (such as no flow) developed by combining guide words with design intentions and process parameters, QHI works by exhaustively positing possible faults, simulating the consequential plant behaviors, and checking for hazards. Posited faults are classes of specific faults (such as leaks) that are matched to particular instances of pipes, reactors, and so on.

The basic premise of the QHI algorithm is that there are two types of faults: (1) those that can be described by perturbations of the original design model (for example, a partially blocked pipe can be simulated by a perturbation function that decreases the conductance of the pipe); and (2) those that require building a new model because the original design model is no longer applicable (for example, a leak of water from the shell side of a heat exchanger into its tube side cannot be simulated by the original design model, because water is now unexpectedly present in a second vessel (tube side) and perhaps reacts with the contents of that vessel, and so on). The new model therefore consists of different active phenomena and, hence, different equations.

The idea that certain faults can be analyzed using the original design model while others require building a new model maps onto the HAZOP study's set of guide words. In general, deviations in process parameters developed by the guide words no, more, and less (such as more flow and less temperature) can be analyzed using a single model; whereas, deviations in process intentions developed by the guide words as well as, part of, reverse, and other than require building and analyzing a new model. For example, the guide word "as well as" is used to develop deviations in which all of the original design intentions are achieved along with some additional activity not originally intended. Particles which should have been removed from the air by a filter, for instance, may flow with the air to the rest of the process units.

The QHI algorithm is therefore broken down into two parts, one in which the model does not change and one in which the model changes. Each of these two subparts consists of three processors: a pre-processor that applies a set of rules to determine if a fault is applicable (this pre-processor replaces the guide word approach in HAZOP studies), a processor that calls QSIM, or QPC and QSIM where the model changes, and a post-processor that interprets the behaviors generated and determines whether or not a hazard could have occurred. The following sections describe the details of these processors for each of the two parts of the QHI algorithm.

QHI algorithm: model does not change

Many of the faults that occur in a chemical plant can be described using the original design model and perturbing model parameters. Faults from a library of faults such as a partially blocked pipe or controller failure (valve fails closed) are matched against particular instances of pipes, controllers, and valves in the physical description to develop perturbation functions that perform step changes in parameters of these specific entities. For example, if the conditions for the fault partially blocked pipe match for the pipe connecting the evaporator and reactor, then a perturbation function that performs a step decreases in the conductance (that is, an increase in the resistance) of the pipe connecting the evaporator and reactor is created.

An overview of the QHI algorithm is provided in Figure 5. First, a QDE model of the plant as it was designed to operate is automatically generated by QPC from a library of phenomena and then the physical description of the plant, a symbolic representation of the process units present, their connections, and operating conditions. This plant model consisting of the

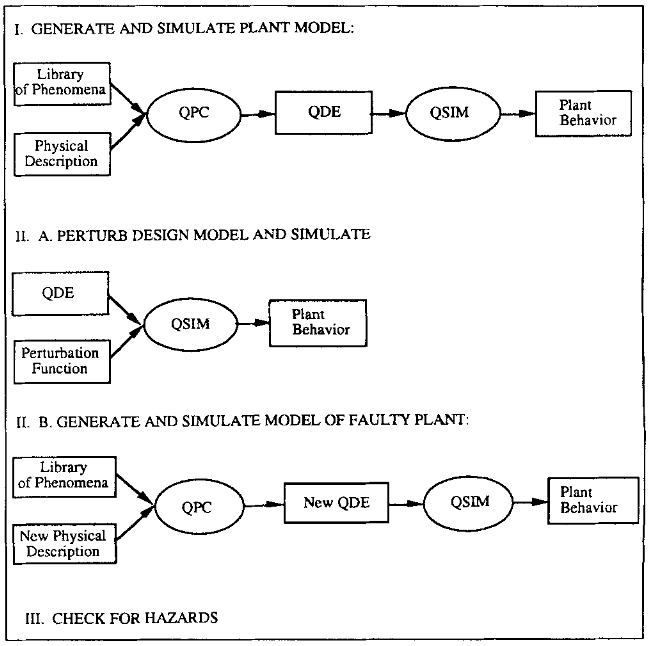

Figure 5. Overview of qualitative hazard identifier (QHI) algorithm.

Use II A when model does not change and II B when the model changes.

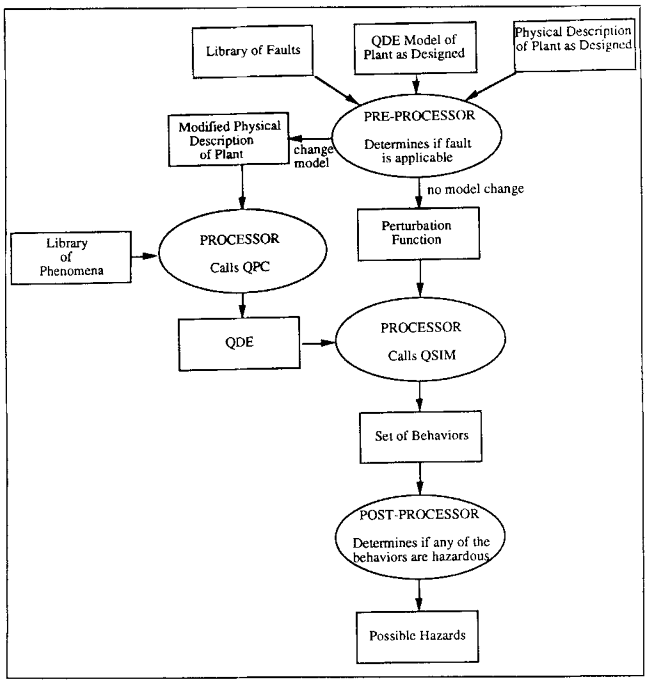
Library of Faults
QDE Model of Plant as Designed
Physical Description of Plant as Designed
Modified Physical Description of Plant
change model
PRE-PROCESSOR Determines if fault is applicable
no model change
Library of Phenomena
PROCESSOR Calls QPC
Perturbation Function
QDE
PROCESSOR Calls QSIM
Set of Behaviors
POST-PROCESSOR Determines if any of the behaviors are hazardous
Possible Hazards

Figure 6. Inputs and outputs of QHI processors.

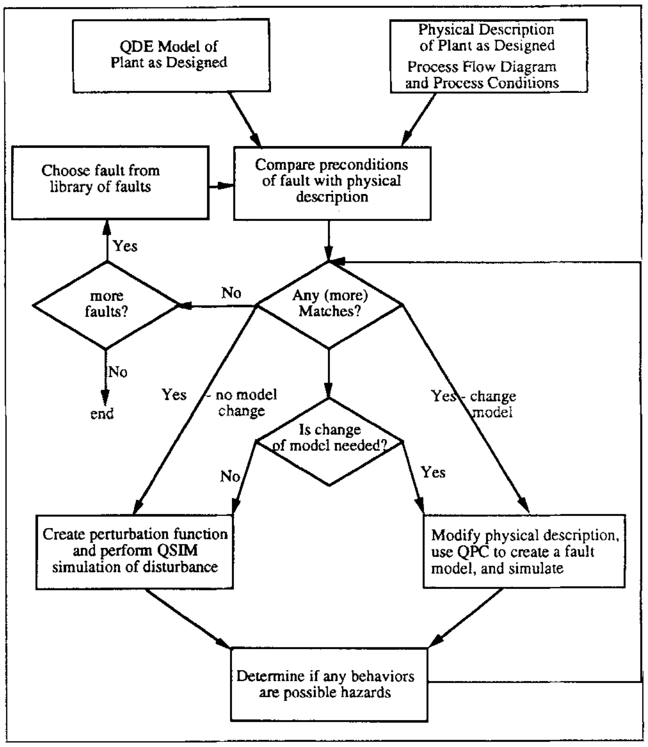
QDE Model of Plant as Designed
Physical Description of Plant as Designed
Process Flow Diagram and Process Conditions
Choose fault from library of faults
Compare preconditions of fault with physical description
Yes
more faults?
No
Any (more) Matches?
No
end
Yes - no model change
Yes - change model
Is change of model needed?
No
Yes
Create perturbation function and perform QSIM simulation of disturbance
Modify physical description, use QPC to create a fault model, and simulate
Determine if any behaviors are possible hazards

Figure 7. QHI algorithm.

equations contributed by the active phenomena is then simulated by QSIM to ensure that the original equations are self-consistent. Next, in the case that the model does not change, a perturbation function is created: a fault is introduced in the system, and the design model is simulated by QSIM to determine the behavior of the plant in response to the fault. Finally, QHI checks all behaviors for hazards.

More specifically, Figure 6 illustrates the inputs and outputs of the QHI processors while a more detailed flowsheet is shown in Figure 7. In the current implementation, certain modifications are hand-encoded into the QDE model once it has been generated by QPC. Most of these changes are made in order to algebraically simplify the equations in the model and to limit ambiguity in its simulation and should become unnecessary as QPC and QSIM are improved.

Given the QPC description and QDE model of the plant as it was designed to operate and a set of rules describing general faults, the pre-processor determines a particular instance of a fault and outputs the perturbation function that can be used to simulate the behavior of a plant with this fault. Each rule in the library of faults consists of the name of the fault (partially blocked pipe), the preconditions necessary for the fault to apply including the objects that must exist and the operating conditions that must hold, and the variable to be perturbed (decrease conductance of pipe) if the preconditions are found to exist in the physical description. Thus, these rules link faults (partially blocked pipe) to their resulting parameter deviations (such as decrease in conductance of pipe).

The scope of faults that have been implemented in QHI when the model does not change is shown in Table 1. The third rule, for example, states that in order to have a blocked filter, a container that is a filter and is connected to another container via a path must exist in the physical description. If a container that satisfies these conditions is found in the physical description, then it is possible for the container (filter) to become blocked, and the fault is implemented by performing a step decrease to zero in the conductance of the path connecting the two containers. Note that the list of rules is not a list of specific faults applicable only to the nitric acid plant. Rather, the faults included in this library are general enough to apply to a variety of plants and also to a number of instances within a single plant. Thus, it is possible for the pre-processor to find more than one set of matching conditions for any given rule. For example, a blocked pipe only requires that a path exists and therefore applies to all pipes in the plant. When more than one match is found, QHI loops through each set of matches individually, creating the associated perturbation function and sending it on to the QHI processor.

Once the perturbation function has been created, the QHI processor calls QSIM to perform a qualitative simulation of the disturbance and to ultimately determine the behavior of the plant as a result of the specific fault. In general, these QSIM simulations for the nitric acid plant contain approximately 250 variables and 320 constraints, most certainly the largest QSIM models built and functioning to date. In order to obtain meaningful results from these large, complex models, the perfect controller, and pseudo-steady-state assumptions, forms of time-scale abstraction are used (Catino, 1993, Kuipers, 1987). The perfect controller assumption ignores all transients associated with the controller variables so that control is instantaneous and faster than all other processes in the plant. The pseudo-steady-state assumption forces all transients to be ignored, allowing comparison of qualitative values between two steady states. Therefore, the plant starts at one steady state, a fault occurs leading to a parameter disturbance, and then a new steady state is reached in which certain parameter values differ from the first steady state.

Table 1. Rules Linking Faults to Parameter Disturbances Without Model Change

Fault	Entities in Physical Description	Relations in Physical Description
Partially Blocked Pipe	Path	Step decrease in conductance of path
Blocked Pipe	Path	Step decrease to zero in conductance of path
Partially Blocked Filter	Container	Step decrease in conductance of path
Blocked Filter	Container	Step decrease to zero in conductance of path
Controller Failure (Valve Fails Open)	Valve	Step increase in valve position of valve
Controller Failure (Valve Fails Partially Closed)	Valve	Step decrease in valve position valve
Controller Failure (Valve Fails Closed)	Valve	Step decrease to zero in valve position of valve
Partially Blocked Heat Path (Fouling)	Heat path	Step decrease in thermal conductance of heat path
Blocked Heat Path (Fouling)	Heat path	Step decrease to zero in thermal conductance of heat path

The QSIM simulation produces the set of all possible next steady states, and these behaviors are sent to the QHI post-processor to determine if any of them are hazardous. The list of possible hazards includes both generic hazards such as over-pressure, over-temperature, and controller saturation, all of which can occur in any plant, and specific hazards such as explosion which are particular to the materials and design involved. For example, if the concentration of ammonia in the reactor of the nitric acid plant reaches a critical value, the mixture becomes explosive. Comparative analysis (Weld, 1987) is used to identify hazards. A brief description of the post-processor mechanism follows.

The generic hazards are each associated with a variable (such as pressure, temperature, valve position). The QHI post-processor selects one of these variables associated with a hazard, such as pressure, and finds all pressure parameters in the model, such as evaporator-pressure, reactor-pressure, and so on. For each of these model-specific pressure parameters, the post-processor checks all behaviors to see if that parameter has increased from the original steady state to the new steady state. If the pressure of a process unit increases in any of the behaviors, then it is feasible for the hazard over-pressure to occur in that unit as a result of the fault. The same procedure is carried out for temperature, valve-position, and mass fraction of ammonia. Note that QHI identifies controller saturation only by testing whether or not valve-position has increased rather than testing to see if it is greater than or equal to a critical value. In a qualitative model, if a variable increases and a landmark larger than the original value exists, it will always be possible for the variable to reach that landmark. If approximate quantitative knowledge were available, this could be used to determine whether or not saturation actually occurs (Catino, 1993).

QHI algorithm: model changes

Some faults cannot be simulated by perturbing the original design QDE model but instead require the creation of a new model. This half of the QHI algorithm shows the strength of incorporating automatic model building capabilities into a computer aid for hazard identification. This section describes the QHI algorithm for automating the case in which faults change the model describing the plant.

An overview of the algorithm when the model changes is presented in Figure 5, parts I and IIB. The first part of the procedure is identical to the case when the model does not change: the design case model is automatically generated by QPC and subsequently simulated by QSIM. Instead of developing a perturbation function, the second step alters the design case physical description of the plant consisting of assumptions about operating conditions and process units and substances present. This new physical description, along with the original library of phenomena, is then input to QPC and a new QDE model describing the fault scenario is automatically generated, simulated, and checked for hazards. As before, Figure 6 shows the inputs and outputs of the QHI processors, while a more detailed flowsheet is presented in Figure 7. Note that the same names are used for the processors in this half of the QHI algorithm as were used when the model did not change, but their functions are slightly different.

Given the physical description of the plant as it was designed to operate and a set of rules describing general faults, the pre-processor determines particular instances of faults and outputs the modified physical description of the plant representing the faulty conditions. Each rule in the library of faults consists of the name of the fault, the preconditions necessary for the fault to apply, including the entities that must exist and the operating conditions that must hold, and a list of modifications to be made to the physical description if the preconditions are found to exist in the physical description. Thus, these rules link faults to their respective physical description modifications. Faults that have been implemented when the model changes include leaks and broken filters.

Once the new physical description has been created, the QHI processor calls QPC to generate a new QDE model and calls QSIM to perform a qualitative simulation. Finally, the post-processor interprets the behavior(s) of the qualitative simulation and determines if any hazards could possibly occur. When the model changes, hazards are identified by comparing the list of active phenomena with QHI's list of hazardous phenomena which may include catalyst poisoning or mass flow to the vessel "atmosphere," that is, leak. If any of the active phenomena match a hazardous phenomenon, then QHI records that phenomenon as a possible hazard.

Results of the QHI algorithm

QHI is able to evaluate the effects of a number of different faults including leaks, broken filters, partially and completely blocked filters and pipes and controller failures leading to a valve failing partially or completely closed or to a valve failing open. Some specific faults can be simulated while others do not complete before exhausting the 224 megabytes of memory on a Sun SparcStation ELC. Run times vary from seconds to days. Numbers of behaviors range between ten and 1,600, with most between 40 and 100. Table 2 presents the specific instances of faults that have been simulated and the resulting hazards that were identified.

There are several reasons why some faults exhaust available computer memory before completing the simulation and identifying hazards. The biggest problem lies in the inefficiency of the constraint satisfaction algorithm used by QSIM to determine and filter all possible variable values given the extremely large number of variables and constraints. While efficiency problems are hardly noticeable when simulating small QDE models, they become a major obstacle when large models like the one describing the nitric acid plant (250 variables and 320 constraints) are simulated. Improvement of the constraint satisfaction algorithm is currently being examined by the Qualitative Reasoning Group at the University of Texas at Austin and at the University of Pennsylvania.

Simulations can also be made faster by rearranging and simplifying the models generated by QPC. Furthermore, as will be demonstrated in the first case study reported here, the type of disturbance made has a definite effect on the simulation time. Often, a result can be obtained more easily when a variable value is perturbed to an extreme (such as zero) rather than simply decreasing it to another positive value. This technique of simulating a disturbance by perturbing to some extreme has been employed by other researchers (Weld, 1988).

Two other problems arise from the QHI algorithm when the model does not change: determining which physical description to use to build the design case QDE model and determining which variables to inherit following a disturbance. First, the basic problem in deciding which physical description to use lies in determining which controllers should be included as part of the process description. When perfect control is assumed, some disturbances lead to contradictions, indicating that the controller cannot compensate for the disturbance. To obtain a consistent solution, the simulation must be rerun with the controller saturated, that is, fixed at its maximum value.

The second problem that arises from the QHI algorithm is determining which variables should be inherited, that is, remaining constant, following a disturbance. At least three factors are involved in making this decision: the controllers included the QDE model, the variable to be perturbed and its "location" in the plant, and the layout of the plant. Some generic inheritance rules can be written; for example, because the perfect controller assumption is used, the set point and the controlled variable are always inherited for each controller. Beyond this, the decision about which variables to inherit becomes trickier, because it depends on the layout of the plant and, hence, is difficult to generalize by developing rules that will apply to a variety of designs. For this reason, variables to be inherited in the current implementation of QHI are provided *a priori* for all possible disturbances rather than determined automatically using a set of rules.

Case study 1: model does not change

Consider, once again, Figure 3 depicting the reactor section of a nitric acid plant. The design has passed preliminary analyses and has been sent to QHI for a hazards review. Faults that can be simulated by making perturbations to the original design model are considered first.

A completely blocked filter is selected off the list of rules linking faults to parameter disturbances (Table 1), and its preconditions are compared with the entities and operating conditions in the physical description of the nitric acid plant. One match is found: the air filter container exists, and it is a filter connected to another unit, the compressor, via the path path-af-comp1. Since the preconditions for a completely blocked filter have been satisfied, QHI creates a QSIM perturbation function that decreases the conductance of path-af-comp1 to zero. The QDE model of the plant is then simulated with this perturbation function to determine the effects of a blocked filter. Inherited variables include air filter temperature, evaporator pressure, and the set points and controlled variables of the three controllers. The simulation takes less than a minute to complete on a Sun SparcStation ELC, and QSIM predicts 72 possible steady states following the disturbance.

All 72 behaviors exhibit the following characteristics: a step decrease to zero in the conductance of the pipe connecting the air filter and the compressor causes the flow rate between the air filter and reactor to decrease to zero. The pressure increases in the filter until it reaches the pressure of source-can1. When the pressures becomes equal, the flow rate into the filter decreases to zero. The ratio controller notices that the flow rate between the filter and reactor has decreased to zero, so the controller closes valve2 to force the flow rate between the evaporator and reactor to zero, maintaining the ratio at its set point. In order for the flow rate out of the evaporator to go to zero, the vaporization rate must decrease to zero. In order for the vaporization rate to decrease to zero, the heat flow rate must also go to zero, which means that the evaporator temperature increases until it reaches the temperature of the steam unit (the heat source). With the flow rates into the reactor decreased to zero, the reactor contents react until one of the reactants disappears. Therefore, the mass fractions of products increase while the mass fractions of reactants decrease. Finally, since the flow rates of the feed streams to the reactor have decreased to zero and these streams cool the reactor contents and since the reaction is exothermic, the reactor temperature increases.

Next, the post-processor examines all 72 behaviors. The description above provides the chain of reasoning that leads QHI to predict the following possible hazards: over-pressure in the air filter and over-temperature in the evaporator and in the reactor.

It is also interesting to examine the results from QHI if the system is designed without a ratio controller. A step decrease to zero in the conductance of the pipe connecting the air filter and compressor again causes the flow rate between the air filter and reactor to decrease to zero. The pressure increases in the filter until it reaches the pressure of source-can1, at which point, the flow rate into the filter decreases to zero. Since the air filter pressure increases from one steady state to the next, QHI identifies the hazard over-pressure in the air filter. There is no flow of air into the reactor; however, since there is no ratio controller in this design, the flow rate of ammonia into the reactor and the evaporator temperature remain unchanged. Since the mass fraction of ammonia in the reactor increases, QHI identifies the hazard explosion in the reactor. Explosion can occur when the mass fraction of ammonia in the reactor reaches a critical value. In this case, the reactor temperature decreases because the ammonia feed acts to cool the reactor, and the reaction rate goes to zero since there is no air left in the reactor to react.

Once models can be automatically generated, engineers can propose changes to the design, rebuild the model, and use QHI to determine if the change results in a hazard. If the nitric acid plant is designed without a ratio controller, QHI predicts an explosion as a possible hazard. When a ratio controller is incorporated in the design, QHI no longer predicts the explosion, and a safer design results.

It is also interesting to note that when a partially blocked filter is simulated using the same QDE model as in the case of a completely blocked filter, that is, the conductance is decreased to a smaller positive number rather than zero, 42 behaviors result, and processing time increases to 25 hours. Perturbing a variable to an extreme value, in this case zero, can greatly reduce simulation time. When a partially blocked filter is simulated, QHI predicts an extra hazard, an explosion, in addition to the same three hazards identified for the completely blocked filter, over-pressure in the air filter, and over-temperature in the evaporator and reactor. Although a ratio controller maintains the ratio of ammonia to oxygen entering the reactor, the flow rates of ammonia and oxygen into the reactor change, and explosion is predicted because QSIM is unable to determine whether or not the mass fraction of ammonia increases, decreases, or remains the same following the disturbance.

This computer-generated result was compared with a HAZOP study performed by a team of engineers (Coulson and Richardson, 1983). For the deviation less flow in the line connecting the air filter and compressor, the engineers predict only one possible hazard caused by a partially blocked filter concerning possible dangerous increase in ammonia concentration, whereas QHI predicts four possible hazards. QHI similarly checks all other potential faults that do not change the model. These results are shown in Table 2.

Case study II: model changes

QHI continues the hazard review of the nitric acid plant by considering next the faults that are simulated by rebuilding the original design model. A broken filter is selected off the list of rules linking faults to physical description modifications, and its preconditions are compared with the assumptions in the original design physical description of the nitric acid process. One match is found: the air filter container exists, it is a filter, and it contains a solid particle phase.

Since the conditions for a broken filter have been satisfied, QHI modifies the physical description as per the rule for a broken filter. The assumptions that ignore influences on the derivatives of the mass fraction of particles in the solid phase in the air filter and of the total mass of the solid phase in the air filter are removed since these describe a correctly functioning filter. This fault also requires finding all process units downstream of the filter because particles will now be present in these vessels. QHI discovers that the air filter is connected to the compressor, which is connected to the reactor which is connected to the product tank. Since the compressor is a transport device and is assumed not to contain phases, the following assumptions about the reactor and product tank are added to the physical description: (contains-unexpected-state reactor air filter) (contains-unexpected-state product-tank reactor), (unexpected reactor air filter particles), and (unexpected product-tank reactor particles).

This new physical description, along with the original library of processes, is sent to QPC, and a QDE model describing a broken filter is built and simulated. The additional assumptions involving "contains-unexpected-state" cause a new solid phase to be created in both the reactor and product-tank. The existence of the new phases combined with the two "unexpected" assumptions above instantiates two new phenomena asserting that particles are now expected in the reactor and product-tank which in turn instantiates two new contained component phenomena, as well as two new solid-phase phenomena in addition to forced flow of particles between the air filter and reactor via compressor and mass flow of particles between the reactor and product tank. More importantly, however, QPC activates the phenomenon catalyst-poisoning because both a solid phase and a catalyst exist in the reactor.

QPC simulates this new QDE model and determines that there are three possible steady states. Ideally, one state should be created, but QSIM branches on the difference in mass fraction of particles in the solid phase in the reactor and mass fraction of particles in the solid phase in the product tank, allowing this difference to be negative, zero, or positive. In this case, the mass fraction of particles in the solid phase in both containers is one as the solid phase consists solely of particles. A generic rule could be added which would reduce the number of behaviors to one. Finally, the QHI post-processor examines the QDE fault model and determines that catalyst poisoning is a possible hazard of a broken filter, precisely the hazard predicted by the team of engineers. QHI similarly checks all other potential faults that change the model.

Comparison With Other Hazard Evaluation Methods

Now that the QHI algorithm has been explained in detail, we can compare QHI with other work that has been done to provide computer support for hazard identification. The basic principle behind the development of QHI is that there are two requirements for a hazard identification aid. First, automatic model building is essential to rebuild models when faults force the model to change and in order to build a general system applicable to a variety of facilities. Secondly, a simulation is necessary to predict the results of faults. Methods for auto-

Table 2. Faults Simulated Using the Qualitative Hazard Identifier (QHI) and the Resulting Hazards

Class of Fault Specific Fault	Hazards	Identif. Time
Partially Blocked Pipe		
Source-can1 and air filter	Overtemperature in evaporator Overtemperature in reactor Explosion	76 h
Air filter and compressor	Overpressure in air filter Overtemperature in evaporator Overtemperature in reactor Explosion	25 h
Reactor and valve3	Overpressure in evaporator Level controller saturation Ratio controller saturation Pressure controller saturation	4 h
Source-can2 and pump	Overpressure in evaporator Level controller saturation Ratio controller saturation	1.3 h
Evaporator and valve2	Overpressure in evaporator Level controller saturation Ratio controller saturation	1.3 h
Blocked Pipe		
Source-can1 and air filter	Overtemperature in evaporator Overtemperature in reactor	20 s
Air filter and compressor	Overpressure in air filter Overtemperature in evaporator Overtemperature in reactor	50 s
Reactor and valve3	N/A—Memory exceeded	
Source-can2 and pump	Overtemperature in evaporator Overtemperature in reactor	47 s
Evaporator and valve2	Overpressure in evaporator Overtemperature in evaporator Overtemperature in reactor	33 s
Partially Blocked Heat Path		
Steam unit and evaporator	Overpressure in evaporator Level controller saturation Ratio controller saturation	1.4 h
Blocked Heat Path		
Steam unit and evaporator	N/A—Memory exceeded	
Partially Blocked Filter		
Air filter	Overpressure in air filter Overtemperature in evaporator Overtemperature in reactor Explosion	25 h
Blocked Filter		
Air Filter	Overpressure in air filter Overtemperature in evaporator Overtemperature in reactor	50 s
Broken Filter (Filter Fails to Remove Particles)		
Air filter	Catalyst poisoning in reactor	2 h
Controller Failure (Valve Fails Partially Closed)		
Valve1	Ratio controller saturation	4 h
Valve2	Overpressure in evaporator Overtemperature in evaporator Overtemperature in reactor Explosion	6.2 h
Controller Failure (Valve Fails Closed)		
Valve2	Overpressure in evaporator Overtemperature in evaporator Overtemperature in reactor	35 s
Valve3	Overpressure in air filter Overpressure in reactor Overtemperature in reactor Overtemperature in evaporator Ratio controller saturation	13 min
Controller Failure (Valve Fails Open)		
Valve1	Overpressure in evaporator	45 min

matically generating fault trees are becoming established, and decision support systems and check lists are commercially available, but applying simulation of process models to hazard identification is quite new.

Of all the hazard evaluation techniques, fault trees have been the subject of the majority of computer aids. Fault trees are graphical models that show how various combinations of equipment failures and faults can lead to a hazard. Fault trees are not used to identify hazards; rather, hazards are identified first by some other means, and then fault trees are developed to identify the faults that lead to those hazards. Algorithms for converting causal models to fault trees have been developed. Fault trees can be generated from signed directed graph (digraph) models (Lapp and Powers, 1977). More recent work by AI researchers, such as Forbus (1984, 1990), has furnished methods for automatically creating the digraphs. Alternatively, mini-fault trees can be created from propagation equations relating inputs and outputs of a unit and from statements linking faults and hazards to particular parameter disturbances (Kelly and Lees, 1986). These mini-fault trees are combined to create the main fault tree for the plant, a rather complex procedure that uses rules to avoid inconsistencies. Also, an algorithm has been developed to convert influence diagrams, which specify the chain of events leading to an accident event, into a fault tree (Nagel, 1991).

Several researchers have developed hazard identification aids. Waters and Ponton (1989) share some similarity to the QHI approach in that they use qualitative simulation for fault propagation. However, they do not provide an algorithm for identifying hazards or their initiating faults. Moreover, they build their models by hand rather than automatically generating them, and apparently they have only looked at flow and not at reaction.

Parmar and Lees (1987) do provide an algorithm for identifying hazards and their causes. Unlike QHI, which starts by systematically examining faults, this algorithm begins by reviewing parameter deviations as in a conventional HAZOP study. Unit models consisting of rules are automatically generated from user-supplied propagation equations relating inputs and outputs of units. For example, the user must provide the propagation equation flow out = f(flow in), which the program automatically converts to the rule "if flow in is high (low), then flow out may/will be high (low)." An enhancement to their approach would be the ability to automatically generate the propagation equations. In addition, a device-centered library of unit models is used which means that only events that have been anticipated *a priori* can occur in a process unit, such as reaction can occur only in units specified by the model builder. The approach, therefore, lacks the flexibility and robustness of QHI, which takes full advantage of a process-centered library of phenomena to automatically build process unit models in which unanticipated events may occur.

Another model-based methodology has been used to determine hazardous chemical and/or physical reactions (Nagel, 1991). Variable influence diagrams are developed from the set of process equations describing a flowsheet by starting with a top-level hazard and chaining back through the events to determine the root causes. This approach is much more detailed than QHI. Exact stoichiometric reactions and quantitative values (such as pressure, temperature) are used to eliminate or to confirm potential hazards. The method does not provide for automatic model generation; however, fault trees can be automatically generated. Finally, Vaidyanathan and Venkatasubramanian (1992) are beginning work on an object-oriented framework for automating HAZOP analysis.

Commercial computer aids for hazard analysis have been less ambitious in helping to identify hazards, but instead have focused on database management and on helping the user choose which hazard identification technique is best suited for his or her purpose. Lihou Technical and Software Services, Inc. (1991) has designed HAZOP version 2.0 to operate on IBM-compatible personal computers. The software combines a word processor for data entry with database tools such as information sorting and retrieval. Similarly, HAZOP-PC uses checklists to help users identify hazards (Primatech, Inc., 1990). Arthur D. Little, Inc. (1992) has developed HAZOPtimizer, another PC-based software for documenting and reporting HAZOP studies, preliminary hazard analyses, what-if/checklist analyses, and failure modes and effects analyses. Battelle has created SOPHIE, a rule-based expert system for the selection of hazard evaluation procedures. Other HAZOP systems have been written in Prolog in the Australian and British academic communities. For example, see Weatherill and Cameron (1989).

Conclusions and Recommendations

This article has explored an innovative methodology for identifying hazards in a continuous chemical process. Unlike previous attempts at providing computer support for hazard identification, QHI takes advantage of recent advances in automatic model building and simulation in order to predict hazards in a nontrivial chemical plant design. QHI postulates possible faults, simulates them, and checks for hazards. Rules are used to match general faults with specific instances within the physical description of the plant. The effects of multiple faults, such as controller saturation and blocked pipe, can be simulated. QHI draws on the simulation and model-building power of QSIM and QPC to reason about a large number of process variables including flow rate, temperature, pressure, mass, mass fraction, and to automatically rebuild models when faults force the model to change.

The QHI algorithm and code is entirely general and independent of the plant being analyzed. It requires as input a library of QPC model fragments describing all processes in the plant, as well as of all faults and hazards that can occur in the plant, including rules to allow QHI to recognize situations where the faults are relevant. Many of these processes, faults, and hazards such as heat transfer, leaks, and overpressure are general. Others, such as the chemistry and some dangers of explosion, are specific to the plant being studied. QHI also requires a description of the plant in a form that QPC can use to produce a QDE model of the plant under normal operation. The overall analysis is, of course, limited by the completeness or incompleteness of the above libraries.

The current library of faults is incomplete. However, faults not currently included, such as power outages, loss of steam pressure, sensor failures, and addition of water, all fit the QHI framework nicely. For example, power outages would result in pumps not functioning (zero flow rates) and sensors reading zero. Power outages would perturb multiple variables, but would require no changes to QSIM or the QHI algorithm.

Similarly, addition of water to a unit would force the model to change as additional phenomena in the library would be activated to model the interactions between water and all other chemical species in the system, but could be handled by the current algorithm.

The other major limitation of QHI comes from the ambiguity inherent in qualitative simulation, which limits the size and complexity of the systems that can be simulated. The ambiguity, however, does not play as significant a role in hazard identification as it does in fault diagnosis, because it is sufficient to know that a hazard *may* occur. Since one qualitative simulation captures the behavior of an infinite number of numerical simulations, ambiguity is advantageous, because with one simulation the engineer can determine if there are any combinations of variable values that lead to a hazardous event. Engineers do not use much quantitative information when performing an initial HAZOP study, but instead, identify hazards first and then filter some as less likely or less significant after obtaining additional information. The fact that engineers filter the hazards generated in a conventional HAZOP study and pass on only a fraction for further assessment explains why QHI often predicts more hazards than the engineers. In general, QHI errs in predicting possible hazards that may not exist, because a spurious result was produced or because the specific combination of variable values cannot possibly lead to a hazard. This feature is certainly better than missing an important hazard and follows from QSIM; all possible physical behaviors are guaranteed to be generated, but other behaviors that do not correspond to any physically possible solution may also be predicted.

The development of QHI has helped to identify areas where more research is required to make QSIM and QHI viable industrial tools. First, if qualitative simulation is to be used on larger problems, QSIM must be more efficient so that intermediate calculations do not require excessive memory. Our results are somewhat discouraging for qualitative reasoning: although in principle QHI should work seamlessly and easily, in fact, minor tweaking of the models is required to make them computationally tractable. Worse, although we have full dynamic models for the plant use in this study, they require far more computing power than we have available. Improvements to QSIM will increase the number of fault cases that QHI can analyze and evaluate. Secondly, once more cases can be analyzed more quickly, more rules need to be developed to decide which QDE model to use to simulate a fault case and to identify which variables to inherit or maintain constant following a disturbance. Thirdly, in the current version of QHI, hazards such as over-pressure and over-temperature are examined only when the model does not change by comparing the value of variables within the same quantity space. To identify these types of hazards when the model changes, a more advanced comparative analysis method, such as Grantham and Ungar (1991), is required since the value of the variable found by simulating the original design model will no longer be directly related to the value of the variable found by simulating the fault model.

In the shorter term, it would be possible to use the exact structure of QHI using purely quantitative models for the simulation. Related work on fault diagnosis in dynamic systems indicates that methods based on automatic model modification can be applied using quantitative as well as qualitative simulation (Vinson and Ungar, 1993). Results were comparable on small systems, but the quantitative simulation scales much better to larger, more densely connected plants.

Future research could also be directed toward increasing the scope of QHI. For example, HAZOP studies are also conducted for startups and shutdowns of chemical plants. Extension of QHI to this area would exploit QPC's incremental model building capabilities and QSIM's ability to do dynamic simulation. Another feature that could be added to QHI is the ability to predict the likelihood of the occurrence of various faults or hazards. With these additional capabilities, QHI would be an integral, contributing member of an industrial HAZOP team.

Acknowledgments

We thank Adam Farquhar and Ben Kuipers for providing QPC and QSIM and supporting us in their use. We also acknowledge NSF Grant CBT 86-57899 for partial support of this project.

Literature Cited

Catino, C. A., "Automated Modeling of Chemical Plants with Application to Hazard and Operability Studies," PhD Diss., Dept. of Chemical Engineering, Univ. of Pennsylvania (1993).

Center for Chemical Process Safety, *Guidelines for Hazard Evaluation Procedures*, AIChE, New York (1985).

Chemical Industries Association, *A Guide to Hazard and Operability Studies*, London (1977).

Coulson, J. M., and J. F. Richardson, *Chemical Engineering*, Vol. 6, *An Introduction to Chemical Engineering Design*, R. K. Sinnott, ed., Pergamon Press, New York (1983).

Crawford, J., A. Farquhar, and B. Kuipers, "QPC: A Compiler from Physical Models into Qualitative Differential Equations," *Proc. of Amer. Assoc. of Artif. Intell.*, 365 (1990).

Crowl, D. A., and J. F. Louvar, *Chemical Process Safety: Fundamentals with Applications*, Prentice Hall, Englewood Cliffs, NJ (1990).

Dalle Molle, D. T., "Qualitative Simulation of Dynamic Chemical Processes," PhD Diss., Dept. of Chemical Engineering, Univ. of Texas at Austin (1989).

Dalle Molle, D. T., B. J. Kuipers, and T. F. Edgar, "Qualitative Modeling and Simulation of Dynamic Systems," *Comp. and Chem. Eng.*, **12**, 853 (1988).

Evans, L. B., J. F. Boston, H. I. Britt, P. W. Gallier, P. K., Gupta, B. Joseph, V. Mahalec, E. Ng, W. D. Seider, and H. Yagi, "ASPEN: An Advanced System for Process Engineering," *Comp. Chem. Eng.*, **3** (1979).

Farquhar, A., "Automated Modeling of Physical Systems in the Presence of Incomplete Knowledge," PhD Diss. available as technical report UT-AI-93-207, Univ. of Texas at Austin (May, 1993).

Forbus, K. D., "Qualitative Process Theory," *Artif. Intell.*, **24**, 85 (1984).

Forbus, K. D., "The Qualitative Process Engine," *Readings in Qualitative Reasoning about Physical Systems*, Morgan Kaufman, San Mateo, CA, p. 220 (1990).

Fouche, P., and B. J. Kuipers, "Reasoning about Energy in Qualitative Simulation," *IEEE Trans. on Syst., Man, and Cybernetics*, **22** (1992).

Grantham, S. D., and L. H. Ungar, "Comparative Analysis of Qualitative Models When the Model Changes," *AIChE J.*, **37**, 931 (1991).

Greenberg, H. R., and J. J. Cramer, eds., *Risk Assessment and Risk Management for the Chemical Process Industry*, Van Nostrand Reinhold, New York (1991).

Kelly, B. E., and F. P. Lees, "The Propagation of Faults in Process Plants: Parts 1-4," *Reliab. Eng.*, **16**, 3, 39, 63, and 87 (1986).

Kletz, T. A., "Eliminating Potential Process Hazards," *Chem. Eng.*, **92**(7), 48 (1985).

Kuipers, B. J., "Qualitative Simulation," *Artif. Intell.*, **29**, 289 (1986).

Kuipers, B. J., "Abstraction by Time-Scale in Qualitative Simulation," *Proc. of the Amer. Assoc. of Artif. Intell.*, 621 (1987).

Kuipers, B. J., and D. Berleant, "Using Incomplete Quantitative

Knowledge in Qualitative Reasoning," *Proc. of Amer. Assoc. of Artif. Intell.*, St. Paul, MN, **324** (1988).

Kuipers, B. J., and C. Chiu, "Taming Intractable Branching in Qualitative Simulation," *Proc. of Tenth Int. Joint Conf. on Artif. Intell.* (1987).

Kuipers, B. J., C. Chiu, D. T. Dalle Molle, and D. R. Throop, "Higher-Order Derivative Constraints in Qualitative Simulation," *Artif. Intell.*, **51**, 343 (1991).

Lapp, S. A., and G. J. Powers, "Computer-Aided Synthesis of Fault Trees," *IEEE Trans. on Reliab.* (1977).

Lawley, H. G., "Loss Prevention: Operability Studies and Hazard Analysis," *Chem. Eng. Prog.*, **70**(4), 45 (1974).

Lee, W. W., and B. J. Kuipers, "Non-Intersection of Trajectories in Qualitative Phase Space: A Global Constraint for Qualitative Simulation," *Proc. of Amer. Assoc. of Artif. Intell.*, St. Paul, MN, **324** (1988).

Lees, F. P., *Loss Prevention in the Process Industries*, Butterworths, London (1980).

Lihou Technical and Software Services, Inc., "Hazop Version 2.0," *Chem. Eng. Prog.*, **87**(10), 72 (1991).

Arthur D. Little, Inc., "HAZOPtimizer," *Chem. Eng. Prog.*, **88**(8), 9 (1992).

Nagel, C. J., "Identification of Hazards in Chemical Process Systems," PhD Diss., Dept. of Chemical Engineering, Mass. Inst. of Technol. (1991).

Parmar, J. C., and F. P. Lees, "The Propagation of Faults in Process Plants: Hazard Identification," *Reliab. Eng.*, **17**, 277 (1987).

Primatech Inc., *HAZOP-PC*, Columbus, OH (1990).

Rosen, E. M., and A. C. Pauls, "Computer Aided Chemical Process Design: The FLOWTRAN System," *Comp. Chem. Eng.*, **1**(1) (1977).

Seader, J. D., W. D. Seider, and A. C. Pauls, *FLOWTRAN Simulation—An Introduction*, 3rd ed., Ulrich's Book Store, Ann Arbor (1987).

Vaidyanathan, R., and V. Venkatasubramanian, "A Knowledge-Based Framework for Automating HAZOP Analysis," AIChE Meeting (1992).

Vinson, J. M., and L. H. Ungar, *The Automatic Process Evaluator*, *Proc. Int. Conf. on FOCAPO*, Rippin et al., CAChE, p. 443 (1993).

Waters, A., and J. W. Ponton, "Qualitative Simulation and Fault Propagation in Process Plants," *Chem. Eng. Res. Des.*, **67**, 407 (1989).

Weatherill, T., and I. T. Cameron, "A Prototype Expert System for Hazard and Operability Studies," *Comp. Chem. Eng*, **13**(11/12), 1229 (1989).

Weld, D. S., "Comparative Analysis," *Proc. of Tenth Int. Joint Conf. on Artif. Intell.*, p. 959 (1987).

Weld, D. S. "Exaggeration," *Proc. of the Amer. Assoc. of Artif. Intell.*, p. 291 (1988).

Manuscript received Aug. 2, 1993, and revision received Jan. 18, 1994.