OPERATIONS RESEARCH, Vol. 23, No. 1, January-February 1975

Man-Machine Synthesis of Disaster-Resistant Operations

J. Roberto Rivas

Petrocel S. A., Tampico, Mexico

and

Dale F. Rudd

University of Wisconsin, Madison, Wisconsin

(Received October 30, 1973)

Disaster-resistant industrial operations are created to avoid certain classes of events known to be dangerous. While a rapid and accurate synthesis of safe operations is necessary during a disaster, it is hindered by the magnitude of the logic-analysis problems encountered and by the sophistication of the operating goals. This paper develops practical methods for computer-aided synthesis of disaster-resistant operations.

AN INDUSTRIAL disaster can result from the complex interaction of individually innocuous events that cascade with unpredictable results. Many of the events occur by chance, such as the failure of equipment or an error on the part of an operator. Other events link together into deterministic sequences, the course of which is entirely predictable once the sequence is initiated.

After the fact, it is often possible to reconstruct the events that caused a major disaster and identify actions that would have quenched the disaster in its early stages. However, it is much more difficult and important to identify the disaster-quenching actions before the disaster is full blown. The difficulty arises as combinatorial problems form anew as each stochastic event occurs. For example, the chance failure of a valve in an oil refinery presents a multitude of operating policy options, many of which are disastrous, and decisions must be made rapidly and accurately to minimize the disastrous results of this chance event.

We expand on the conjecture that the violent and destructive events so prominent in industrial disasters are to a large extent the effects of an inability to anticipate the long-range effects of current actions. Before the first major disruptions, sequences of seemingly innocuous events occur that force the system into a mode of operation from which it cannot be extricated without disaster. Could the effects of these events be foreseen, actions could be taken to intercept and quench the impending disaster.

An enormously large number of ways exist for manipulating most industrial processes; many are extremely dangerous, and a few achieve useful processing objectives. In the sample process considered in this paper seventeen valves can be opened or closed, leading to $2^{17} = 131{,}072$ final valve positions. Considering that the sequence in which the valves are manipulated during transient operation forms an even larger combinatorial problem, it is hopeless to examine the safety of all possible operating problems that might arise. In practice, only normal operation, start-up, shut-down, and major emergency situations can be examined a priori, leading to

the unfortunate possibility that an extremely hazardous mode of operation may be entered into unknown to the operators as they attempt to contend with other innocuous situations that may arise.

A misleading degree of security is obtained by the a priori development of safety-interlock systems, in which operating procedures are established to avoid hazards in certain preconceived emergency situations; several major disasters have been traced to the use of a priori interlock systems under conditions that were not anticipated. The ability to analyze each situation as it arises is a necessary requirement for developing disaster-resistant operations.

In this paper we summarize some developments in research on man-machine synthesis of disaster-resistant operations for industrial process systems. The basic structure of the industrial process is given to the computer as a connector-node network through which deterministic events propagate, and as general statements of event combinations that can initiate major disasters. When an unpredicted emergency arises, man and machine interact to solve the enormous combinatorial problems that mask the operating policy that best quenches the impending disaster. In matters of seconds these problems can be solved, providing real-time monitoring of emergency situations.

We draw a parallel between the classic test case of artificial intelligence, the computer playing chess, and the problem of concern here, the synthesis of fail-safe operations. Chess pits two minds against each other in a situation so complex that neither can understand it completely, yet the game is sufficiently well defined and the rules so simple that a beginner can follow the play of world champions. The synthesis of large industrial operations takes on many of the troublesome characteristics of a chess game.

Claude Shannon in 1949 proposed that the computer could be programmed to play chess merely by rapid look-ahead to proposed continuations of play to a depth sufficient to identify probable winning moves. The examination of all continuations is not feasible, since there are something like 10^{120} possible continuations and only 10^{16} microseconds in a century to explore them. A sequence of valve operations corresponds to a continuation of play in chess, and in large industrial operations the combinatorial problems are sufficiently difficult to prevent examining even a small fraction of all possible operations in the search for fail-safe operations. The skilled chess player and the skilled process engineer have much in common in their intuitive ability to create useful continuations.

Two distinctly different problems are solved, operating procedure *analysis* and *synthesis*. Analysis involves the computer formulation and solution of the problems in sequential logic that arise in predicting the response of the system to preconceived sequences of operational events; the computer can deduce the response of the system to any given operation policy. Synthesis involves the creation of operational policies to best reach some complex goal in a safe manner. Man and machine interact to solve the problems in inductive reasoning that arise in these synthesis problems, and this requires a hierarchial goal-structure formulation similar to that used in computer-aided chess-playing programs.

It is entirely feasible to provide the following capability to the computer control center of a major industrial process. Given a disruptive event such as a local fire, explosion, or operator error, the computer can be called on to recommend the proper response. The nature of the disruption is given to the computer and it responds in

a matter of seconds with alternate sequences of operating instructions to guide the process operators during the disaster. Further, the computer can monitor the individual operations to lock out events that can lead to levels of operation from which the process cannot be extricated.

GENERAL FORMULATION

Attention is focused on the process industries in which potentially dangerous material moves in a complex manner through systems of pipes, vessels, reactors, and the like. An oil refinery or a chemical-processing center involves acres of complex systems of equipment through which an industrial disaster can propagate.

There are certain events that are known to be extremely dangerous, including:

1. Connecting of certain locations via an open path through the system.
2. The contact of certain combinations of material species.
3. Blocking the flow of certain material.
4. Moving certain material into given regions of the system.
5. Certain local events specific to the process technology.

The fundamental problem is creating operating policies to achieve complex goals while avoiding these dangerous events, especially when the system is already partially disabled from previous events.

Figure 1 is a usual mapping describing the potential routes of material flow in an industrial process; here the catalyst-regeneration system for an oil refinery. This is cast into the connector-node structure shown in Fig. 2, and represented symbolically in Table I. Connectors are defined as regions in the process that contain only a single inlet and outlet for material flow, and nodes are the points at which the connectors join to form the system. Each distinct material species has one or more higher-pressure sources and can be driven through the system toward lower-pressure sinks. The movement of material is controlled by valves, the pressure driving forces, and the structure of the system.

The particular example we see here is of the magnitude encountered industrially. Twenty-seven connectors, seventeen valves, and six species form a sufficiently difficult test case. Industrially, five or ten such systems would be tied together to form a processing complex. However, the present system is sufficient for illustrative purposes.

Hydrocarbons enter the reactors to be upgraded by chemical reactions, which also degrade the catalyst. Over a 24-hour period, the catalyst in a given reactor is regenerated by a complex set of operations involving hydrogen, oxygen, natural gas, inert gas, and carbon tetrachloride. A catalyst-regeneration cycle can involve thirty or forty changes in valve position, the sequence of which must be designed to skirt these dangerous conditions:

1. The hydrocarbon inlet must not be connected by an open path to either hydrogen or natural-gas inlets.
2. The hydrogen inlet must not be connected by an open path to the natural-gas inlet.
3. The hydrocarbons, hydrogen, and natural gas must not mix with inert gas, oxygen, or carbon tetrachloride.

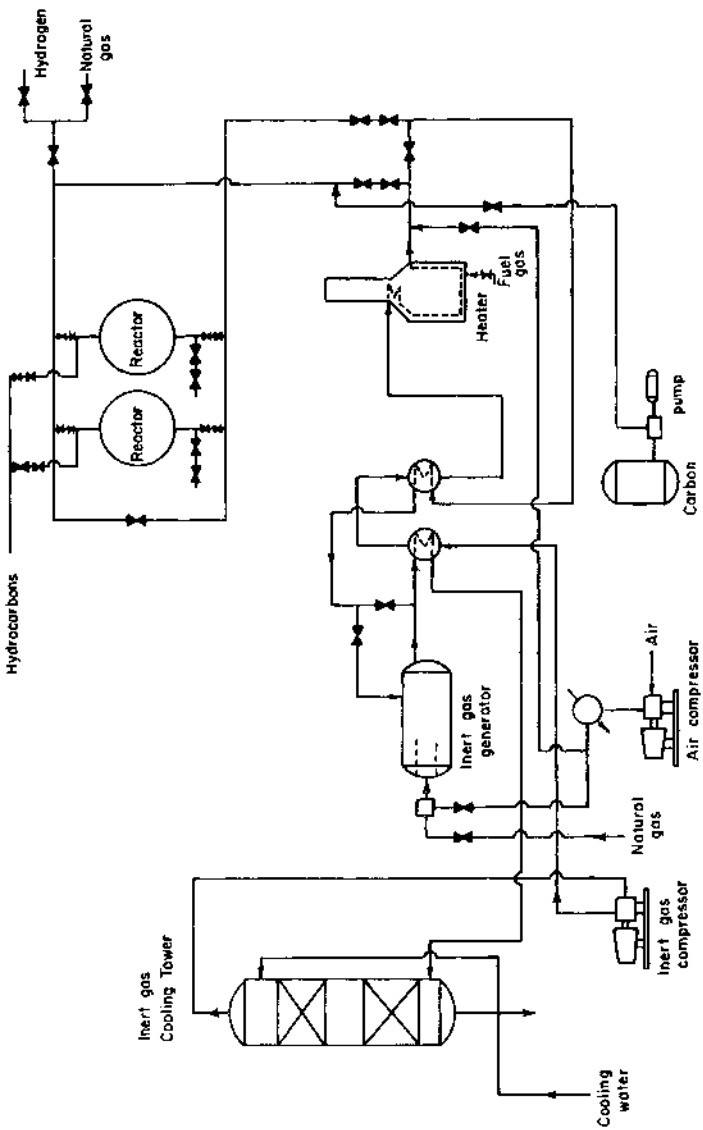

Fig. 1. The catalyst-regeneration system for an oil refinery.

4. The flow of hydrocarbons and inert gas must never be blocked.
5. Oxygen and carbon tetrachloride must never flow to the hydrocarbons outlets.
6. Hydrocarbons must never appear in the upper and lower reactor headers.
7. Hydrogen and natural gas must never go into the lower reactor headers.

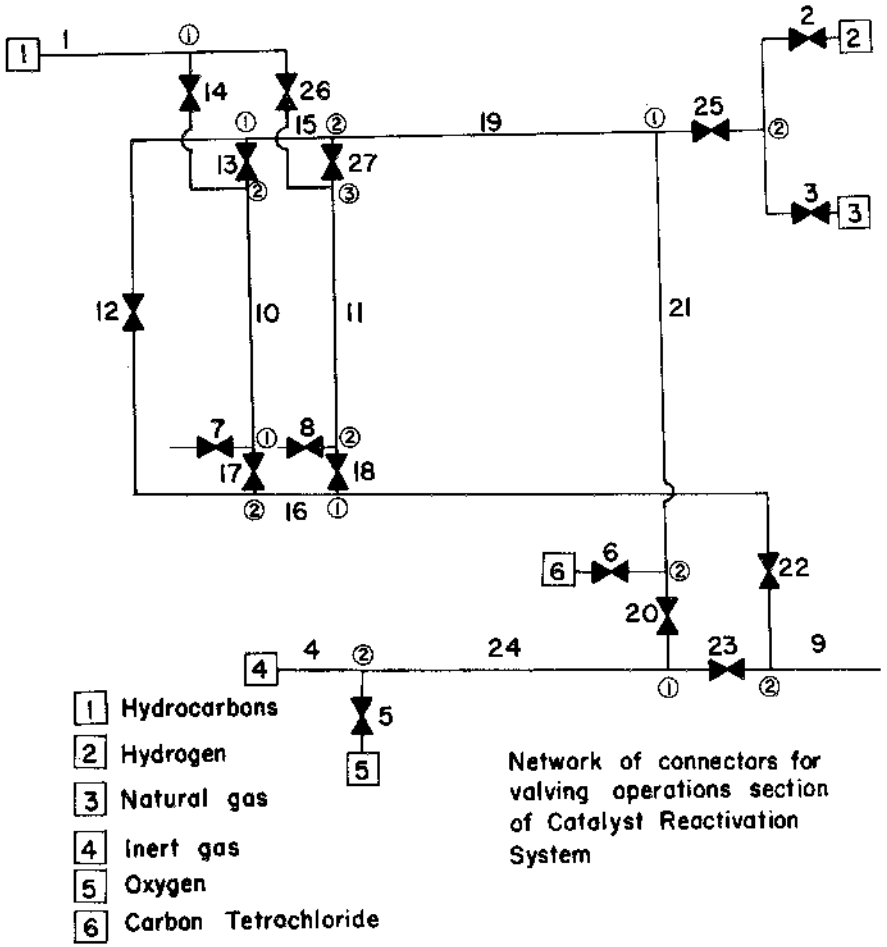

Fig. 2. The connector-node structure for the catalyst-regeneration system.

These safety constraints are shown symbolically in Table II, in which the species and connectors are numbered as in Fig. 2.

The analysis and synthesis of operational policies for trivially simple systems are now considered as an introduction to the general capability required in the far more complex systems of industrial interest. Figure 3 illustrates a system of six connectors, a source of species A, a source of species B, and an outlet; Table III contains the operating procedure to be analyzed. Initially all six valves are closed and no material is trapped in the system. When valves 1, 3, 5, and 6 are opened,

TABLE I

STRUCTURE OF THE CATALYST-REACTIVATION SYSTEM

Connector i	Node j	Node l	Connectors joined to node j	Connectors joined to node l	Species	Valves
1	1		14, 26		1	
2	2		3, 25		2	x
3	2		2, 25		3	x
4	2		5, 24		4	x
5	2		4, 24		5	x
6	2		20, 21		6	
7		1	10, 17			x
8		2	11, 18			x
9		2	22, 23			
10	1	2	7, 17	13, 14		
11	2	3	8, 18	26, 27		
12	1	2	13, 15	16, 17		x
13	1	2	12, 15	10, 14		x
14	1	2	1, 26	10, 13		x
15	1	2	12, 13	19, 27		
16	1	2	18, 22	12, 17		
17	1	2	7, 10	12, 16		
18	1	2	16, 22	8, 11		
19	1	2	21, 25	15, 27		
20	1	2	23, 24	6, 21		x
21	1	2	19, 25	6, 20		
22	1	2	16, 18	9, 23		x
23	1	2	20, 24	9, 22		x
24	1	2	20, 23	4, 5		
25	1	2	19, 21	2, 3		x
26	1	3	1, 14	11, 27		x
27	2	3	15, 19	11, 26		x

Note. There were no trapped species initially.

TABLE II

SAFETY CONSTRAINTS FOR THE CATALYST-REGENERATION SYSTEM

Must not mix	Must not connect inlets	Must not block flow	Must not flow at outlets 7 and 8	Must not be at reactor headers
1, 4	1, 2	1	4	1 (upper and lower headers)
1, 5	1, 3	4	5	
1, 6	2, 3		6	
2, 4				
2, 5				2, 3 (lower header)
2, 6				
3, 4				
3, 5				
3, 6				

Note. The species nomenclature is from Fig. 2.

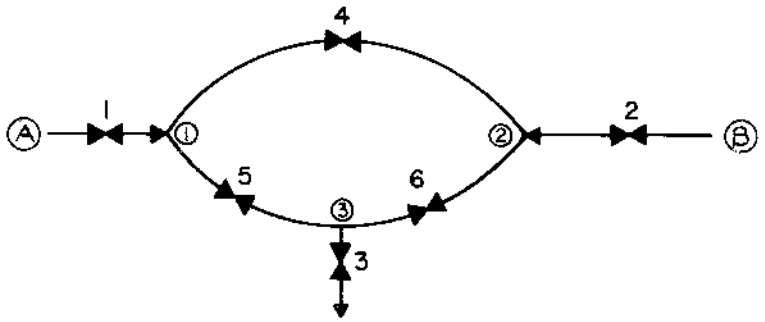

Fig. 3. A small network of connectors.

species A flows through connectors 1, 5, and 3, and reaches sides 2 and 3 of connector 6, side 2 of connector 2, and sides 1 and 2 of connector 4. When valve 2 is opened and valves 1 and 5 closed, species B flows through connectors 2, 6, and 3, and reaches side 2 of connector 4 and side 3 of connector 5. Species A is trapped in side 1 of connectors 1, 4, and 5, in side 2 of connector 4 and side 3 of connector 5. Species A is flushed out of connector 2, 6, and 3 by the flow of species B. After valve 4 is opened and valves 2 and 3 closed, the two species are trapped in all connectors.

As seen, the analysis of a proposed operating policy is merely a bookkeeping problem. Unfortunately, the problem becomes excessively difficult for industrially large systems, and we are required to develop a logical framework of analysis to handle these important problems.

Figure 4 illustrates an operating-policy synthesis problem. In this example, hydrogen flow is to be initiated through a system initially filled with air. The valving sequence sought is one in which air and hydrogen are not in contact. Below is a summary of the mental processes we use to solve this problem intuitively.

The only difference between the initial and final valve positions in Fig. 4 are the positions of valves 1, 3, and 5. However, the operating procedure [open 1, close 3, open 5] is decidedly hazardous. The operating objective "initiate hydrogen flow to the low-pressure outlet" is too far removed from the identification of valving sequences. We need a sequence of less ambitious goals to bridge the gap.

"Evacuate air" may be a reasonable place to begin. To reach this goal we must reach two goals in this sequence "stop air flow" and "open system to low-pressure outlet."

"Stop air flow" is accomplished by [close 3] and "open system to low pressure outlet" by [open 5].

TABLE III

Operating Procedure for the Network of Figure 3

Order of operation	Open valves	Close valves
1	1, 3, 5, 6	
2	2	1, 5
3	4	2, 3

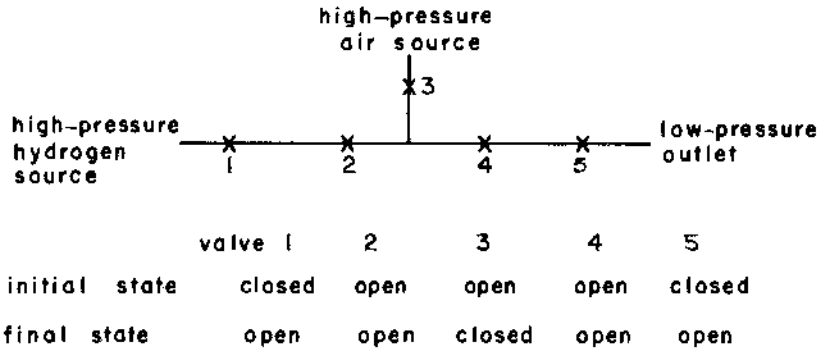

Fig. 4. An example operating-policy synthesis problem. Initially the system is filled with air. The operating objective is to initiate hydrogen flow to the low-pressure outlet. The safety criterion is that hydrogen and air must not be in contact.

Now the goal "initiate hydrogen flow" points directly to the operation [open 1]. In summary, a safe operation sequence is [close 3, open 5, open 1].

We now examine the details of this intuitive synthesis of operating procedures. The original goal, the operating objective "initiate hydrogen flow to low-pressure outlet" drew attention to places in the system where hydrogen was present, namely, above valve 1. However, any changes in valve 1 brought air and hydrogen in contact. Thus, the only hint of where to begin, the word *hydrogen* in the original goal, brought us to a dead end (*see* Table IV).

However, attempts to use the original goal brought up the word *air*, and this suggested a level I goal "evacuate air." But opening valve 5 to evacuate air merely opens a path through the system for air to flow. The inability to reach the level I goal suggested level II goals of "stop air flow" and "open system to low-pressure outlet," which identified valid valve operations.

Table IV shows the penetration into the operating-procedure synthesis problem by a hierarchy of goals until the point is reached where unique valving operations are identified. It appears then that the cutting edge in the tools of synthesis is a hierarchy of goals designed to slice into the problem to the depth of individual valve operations.

TABLE IV

THE HIERARCHY OF GOALS TO IDENTIFY A SAFE OPERATING PROCEDURE FOR THE SYSTEM IN FIGURE 4

Level 0 goal	Level I goal	Level II goal	Valve operation
Initiate hydrogen flow to low pressure outlet	Evacuate air	Stop air flow	Close 3
		Open system to low pressure outlet	Open 5
	Initiate hydrogen flow		Open 1

Final sequence: [close 3, open 5, open 1].

THE LOGICAL STRUCTURE OF THE ANALYSIS

SEQUENTIAL LOGIC CAN be used to develop computer programs to handle the general operating-policy analysis problem. We need only take suitable liberties with the physical problem. If interest is limited to only the presence and absence of species in the system and if the valves are considered to be open if they are not completely closed, the analysis can be cast in a Boolean or two-valued algebraic structure. In reality, certain dangerous situations may not arise unless materials are present in sufficient amounts, and these amounts can occur by the partial opening of valves. These liberties we take with reality lead to conservative actions, and for this reason the policies developed may actually be more resistant to disaster than predicted.

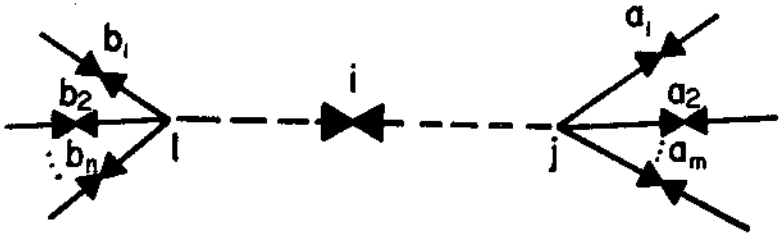

Fig. 5. A general connector i defined between nodes l and j where other connectors join.

Figure 5 shows a general connector i defined between nodes l and j where other connectors join. A species k can appear at this node in one of three possible conditions, described by the following declarative statements:

$F(i, j, k)$: Species k is flowing through side j of connector i. (1)

$B(i, j, k)$: Species k has reached side j of connector i from an inlet but there is no flow. (2)

$T(i, j, k)$: Species k is trapped in side j of connector i. (3)

Flowing F, blocked B, and trapped T are the only significantly different states in which a species can be found. Flowing indicates the continuous movement of material from an inlet source through the system to an outlet sink. Blocked means that there is a direct connection via an open path to an inlet, but that flow is blocked by a closed valve somewhere downstream. Trapped means that the species is held in place by closed valves somewhere upstream and somewhere downstream. The fundamental problem in analysis is to determine the truth value of statements (1), (2), and (3) as a function of the structure of the system, its initial condition, and the positions of the valves as defined by the variables

$v(i)$: valve on connector i is open. (4)

Since, in our problem formulation, these statements are either true (1) or false (0), the principles of propositional logic are the basis of system analysis.

Analysis of Flow

Six additional propositions must be introduced to determine the value of $F(i, j, k)$. The following interrelations among these variables form the sequential logical structure of the analysis:

$R(i, j, k)$: Side j of connector i is reachable by species k from an inlet. (5)

$P(i, j)$: Side j of connector i is included in a nonlooped path that allows flow to an outlet. (6)

$L(i, j)$: Side j of connector i is included in a looped path that allows flow to an outlet. (7)

If statement (5) and either statement (6) or (7) are true, the flow proposition is true; that is,

$$F(k, j, k) \leftarrow R(i, j, k) \cdot [P(i, j, k) + L(i, j, k)]. \quad \text{(a)}$$

Statement (5) is true if any of the half connectors joined to node j have been reached by an inlet, or if side l has been reached at the same time the valve on i is open:

$$R(i, j, k) \leftarrow [\textstyle\sum_{am} R(am, i, k) + R(i, l, k) \cdot v(i)]. \quad \text{(b)}$$

Statements (6) and (7) may seem redundant until it is realized that there are open paths that will not allow flow; for example, an isolated loop attached to a straight run by a single connector: flow from the inlet along the straight run, through the single connector, around the loop, back through the single connector, and down the straight run to the outlet is not possible—it requires that material flow both ways in the single connector. We must be concerned with such anomalies.

The local direction of flow is determined by a pressure variable:

$HP(i, j)$: Side j of connector i is the high-pressure side. (8)

Now, statement (6) is true if valve i is open and if (a) any of the half connectors joined to node j is included in an open path with side j or the low-pressure side, or (b) if side j is the high-pressure side and side l is in an open path:

$$P(i, j) \leftarrow v(i) \cdot [\textstyle\sum_{am} P(am, j) \cdot \overbrace{HP(i, j)} + HP(i, j) \cdot P(i, l)]. \quad \text{(c)}$$

Inlets are at high pressure; thus, the connectedness to an inlet determines the high-pressure side of an internal connector:

$RI(i, j)$: Side j of connector i is reached by an inlet. (9)

Thus, side j of connector i is the high-pressure side if any of the half connectors joined to node j is reached by an inlet, and if at the same time side l is not also a high-pressure side:

$$HP(i, j) \leftarrow [\textstyle\sum_{am} RI(i, j)] \cdot \overline{HP}(i, l). \quad \text{(d)}$$

Finally, RI is true only if (a) any of the half connectors joined to node j is reached by an inlet, or if (b) side l is reached by an inlet with the valve open:

$$RI(i, j) \leftarrow [\textstyle\sum_{am} RI(am, j)] + RI(i, l) \cdot v(i). \quad \text{(e)}$$

Loops are dealt with with these two variables:

$LO(i,j)$: Side j of connector i is included in a loop and in an open path. (10)

$LL(i,j)$: Side j of connector i is part of a loop. (11)

Thus,

$$LO(i,j) \leftarrow v(i) \cdot LO(i,l) \cdot [\textstyle\sum_{am} LO(am,j)], \tag{f}$$

$$LL(i,j) \leftarrow LO(i,l)[P(i,j) \cdot LL(i,j) + LL(i,l) + \textstyle\sum_{am} LL(am,i)]. \tag{g}$$

Analysis of Blocking

Statement 2, "species k has reached side j of connector i from an inlet but there is no flow" is true if side j of connector i can be reached by species k from an inlet and if side j of connector i is not included in an open path:

$$B(i,j,k) \leftarrow R(i,j,k) \cdot \overline{[P(i,j) + L(i,j)]}. \tag{h}$$

This is all that is required to determine the truth of the blocking proposition.

Analysis of Trapping

Species k is trapped in side j of connector i only under the following conditions: Species k is present in side j of connector i, and j of connector i is not connected to any open outlet that is not reached by species coming from an inlet, and side j of connector i is not connected to an inlet where species k is present, and side j of connector i is not connected in an open path. The last two statements are functions of the previously defined variables R, P, and L, and the first two statements constitute new information.

Species k is present in side j of connector i (a) if species k is present at any half connector joined to node j, or (b) if species k is present at side l and the valve is open, or (c) if species k was present there at the last operating increment. Thus, if

$PR(i,j,k)$: Species k is present in side j of connector i, (12)

then

$$PR(i,j,k) \leftarrow [\textstyle\sum_{am} PR(i,j,k)] + PR(i,l,k)v(i) + PR^*(i,j,k), \tag{i}$$

where the asterisk denotes the last operating increment.

Finally, we must be concerned with the removal of material via an open connection to an outlet; this is the 'deactivation' of connectors with trapped material:

$ACT(i,j,k)$: Side j of connector i is not connected to an open outlet that is not reachable by species coming from an inlet. (13)

Thence

$$ACT(i,j) \leftarrow [ACT(i,l) + RI(i,l + \overline{v(i)}] \cdot [\textstyle\sum_{am} ACT(am,j) + RI(i,l)]. \tag{j}$$

The truth of the trapped statement is then

$$T(i,j,k) \leftarrow PR(i,j,k) \cdot ACT(i,j) \cdot \bar{R}(i,j,k) \cdot \bar{P}(i,j)\bar{L}(i,j). \tag{k}$$

The sequential solution of equations (a) through (k) constitutes a logical analysis of the movement of material through a general system for any given change in valve positions. As we shall see toward the end of this paper, the analysis is rapid on the UNIVAC 1108 computer, averaging only 0.4 seconds per valve change on the catalyst-regeneration-system test problem. The computer has been programmed to generate the specific logical equations for any given process structure and to execute without error the logic analysis at speeds and a precision well beyond those of the experienced engineer.

THE HIERARCHICAL AND HEURISTIC STRUCTURE OF THE SYNTHESIS

WE NOW ADDRESS the problem of transforming general word statements of an operating objective into the detailed sequence of valve operations required to accomplish the objective. This synthesis problem is very similar to the problem of transforming a strategy of chess play into the detailed moves required to execute it. It pays to examine the similarities.

In chess the rules are sufficiently simple and well defined that a beginner can analyze the play of world champions. However, the synthesis of winning play encounters fantastically large combinatorial problems, there being an estimated 10^{120} unique continuations of play and only 10^{16} microseconds per century to examine them. Skill in chess cannot rely on principles that require the examination of even a minutely small fraction of all possible continuations of play. The analysis problem in process operations has been reduced to a computer program by the developments of the previous section; yet, the synthesis problem remains as a central difficulty. As in chess, we cannot hope to use synthesis principles that require the examination of even a small fraction of possible valving sequences.

Synthesis ought to be viewed in a data-processing context. The primary goals "synthesize a winning game of chess" and "reactivate the catalyst in the reactor" are concepts too complex to communicate to a mechanism handling the details of play. These primary concepts must be replaced by a set of less complex concepts, and these latter by even less complex concepts until communication can be established with the details-handling mechanism. This process then forms a hierarchy of goals that penetrates the combinatorial problem to isolate feasible operating sequences. Success depends critically on the structure of the hierarchy.

Man-Machine Interface

Table IV shows the goal hierarchy for the simple synthesis problem of Fig. 4. Figure 6 illustrates the general structure built into the disaster-resistant operations synthesizer. The operating objective is the single level 0 goal, a concept that cannot be communicated to the computer. This is replaced by an ordered set of level I goals that are verbal instructions still too complex for communication to the computer, but capable of reduction to level II goals. The n level II goals, which replace each of the z level I goals, are logical propositions that form the interface between man and machine. We examine this interface. Level II goals are de-

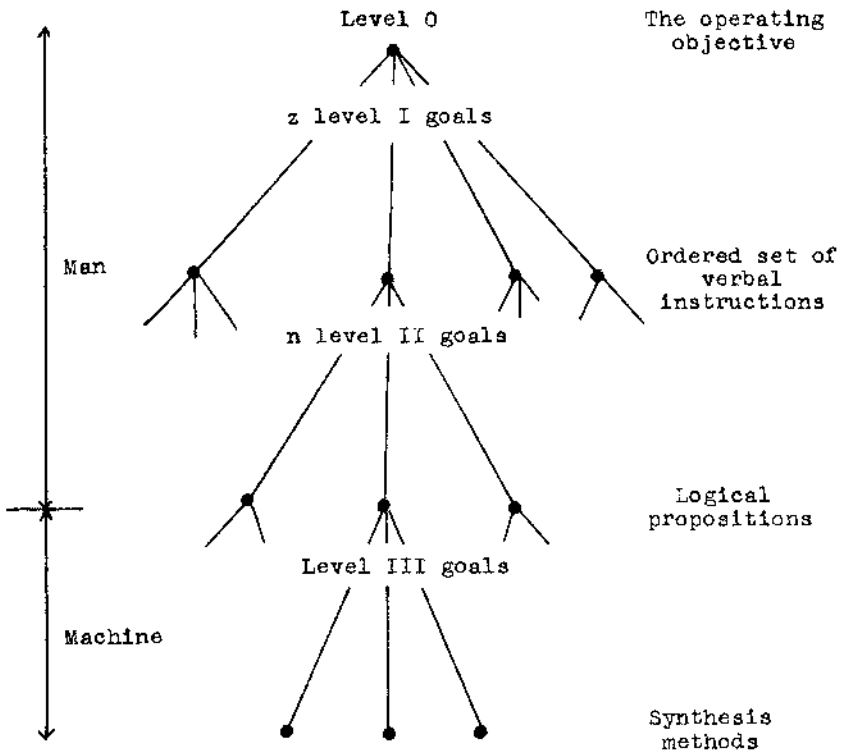

Fig. 6. The goal hierarchy for man-machine communication.

fined by combinations of three propositions:

IIF (n, z): Species $k(n, z)$ is to flow at side $j(n, z)$ of connector $i(n, z)$.

IIB (n, z): Species $k(n, z)$ is to be blocked at side $j(n, z)$ of connector $i(n, z)$.

IIT (n, z): Species $k(n, z)$ is to be trapped at side $j(n, z)$ of connector $i(n, z)$.

where n and z refer to a goal n at level II of goal z at level I.

Table V shows how combinations of these three variables communicate eight

TABLE V

Meaning of the Combination of Values of the Variables IIF, IIB, and IIT

IIF(n,z)	IIB(n,z)	IIT(n,z)	Meaning
1	0	0	Species k is to be flowing at side j of connector i
0	1	0	Blocked
0	0	1	Trapped
1	1	0	Flowing or coming from an inlet
0	1	1	Coming from an inlet or trapped
1	0	1	Flowing or trapped
1	1	1	Present in any state
0	0	0	Not present at all

kinds of information, and Table VI shows the communication of this information to the computer for the catalyst-regeneration system. In Table VI, the original complex goal "regenerate the catalyst" has been replaced by sixteen ordered verbal instructions, each of which has been reduced to the logical-proposition structure. It is interesting to observe that the verbal instructions are at the complexity level at which an engineer would communicate to another engineer with equal experience with the expectation of error-free communication. In fact, man-man communication might occur at the logical-proposition level to avoid misunderstanding. Thus, this man-machine communication structure does not transmit the solution to the problem, merely a general problem statement.

In Table VI certain of the level II goals are marked by an asterisk; they are part of the advice taking discussed later. Table VI hopefully forms an unambiguous description of the operating objective for the computer, and we next examine the synthesis of valve-operation sequences.

Synthesis Heuristics

The level I goals are examined in the order given, and synthesis begins with a scan of all valve positions searching for the operations that are positively related to the level I goal as described by the n logical propositions. Positive relation to the goals is defined by the following heuristic conditions.

A valve operation is positively related to the operating objective if, at levels II or higher, there exists an unmet goal that:

(A) Requires flow of a species at a site, and the valve operation causes the flow or causes the species to be newly blocked at the site.

(B) Requires blockage of a species at a site, and the valve operation causes the blockage or causes the species to appear newly at the site.

(C) Requires trapping a species at a site, and the valve operation causes the trapping or causes the species to flow newly at the site.

Or (D) requires the evacuation of a site, and the valve operation causes the evacuation or traps material currently blocked there.

Or if (E) the connector is currently empty.

Notice that these heuristics identify actions that will satisfy a goal or cause precursor events that set the stage for the eventual satisfaction of the goal. The services of the operating-analysis logic capability are called upon at each stage in this preliminary scanning of the system for sites at which operating moves are likely to be useful.

During the scanning one of three possibilities will occur: (a) sites are identified where positively related actions can be executed, (b) the only positively related sites lead to hazards, or (c) no positively related sites exist. The first possibility can generate feasible valving sequences, whereas the other two lead to the concepts of level III goals and advice taking. Figure 7 illustrates the sequence of events as they occur during synthesis.

In passing we note that the detection of potentially hazardous conditions is an interesting problem not discussed here. Rivas, Rudd, and Kelly[4] discuss this in detail.

TABLE VI

SETS OF GOALS FOR THE CATALYST-REACTIVATION SYSTEM

Level I goals			Level II goals					
Verbal description	z	n	$k(n,z)$	$j(n,z)$	$i(n,z)$	IIF(n,z)	IIB(n,z)	IIT(n,z)
Replace reactor in service and stop hydrocarbon flow in the reactor to be reactivated	1	1	1	2	11	1	0	0
		2	1	1	10	0	0	0
Purge hydrocarbon from reactor with hydrogen	2	1	2	1	10	1	0	0
Pressurize reactor with hydrogen	3	1	2	1	10	0	0	1
Depressure and evacuate hydrogen from reactor	4	1	2	1	10	0	0	0
Pressurize reactor with inert gas	5	1	4	1	10	0	1	0
Circulate inert gas	6	1	4	1	10	1	0	0
	*6	2	4	1	22	1	0	0
Divert total flow of inert gas to reactor	7	1	4	1	23	0	1	0
	7	2	4	1	12	0	1	0
Chlorinate and rejuvenate	8	1	5	1	10	1	0	0
	8	2	6	1	10	1	0	0
Pressurize reactor with inert gas, chlorine, and oxygen	9	1	4	1	23	1	0	0
	9	2	4	1	10	0	0	1
	9	3	5	1	10	0	0	1
	9	4	6	1	10	0	0	1
	*9	5	4	1	19	0	0	1
	*9	6	5	1	19	0	0	1
	*9	7	6	1	19	0	0	1
Depressurize and evacuate reactor	10	1	4	1	10	0	0	0
	10	2	5	1	10	0	0	0
	10	3	6	1	10	0	0	0
	*10	4	4	1	19	0	0	0
	*10	5	2	2	25	0	0	0
Pressurize with natural gas	11	1	3	1	10	0	1	0
Purge with natural gas	12	1	3	1	10	1	0	0
Stop flow of natural gas	13	1	3	1	10	0	0	0
Purge with hydrogen	14	1	2	1	10	1	0	0
Pressurize with hydrogen	15	1	2	1	10	0	1	0
Depressurize and evacuate hydrogen, start hydrocarbon flow in reactivated reactor, stop hydrocarbon flow in the other reactor and leave hydrogen in the upper header	16	1	2	1	10	0	0	0
	16	2	1	1	10	1	0	0
	16	3	1	2	11	0	0	0
	16	4	2	1	13	0	1	0

Advice Taking and Level III Goals

The direction of search for sequences of valve operations is guided by the maintenance of a positive relation between the level II goals and the candidate valve operations. As we have seen, the heuristics that define a positive relation include not only a direct link with the goals but encourage actions that are precursors to goal satisfaction. Even with this flexibility, it is entirely possible that a feasible

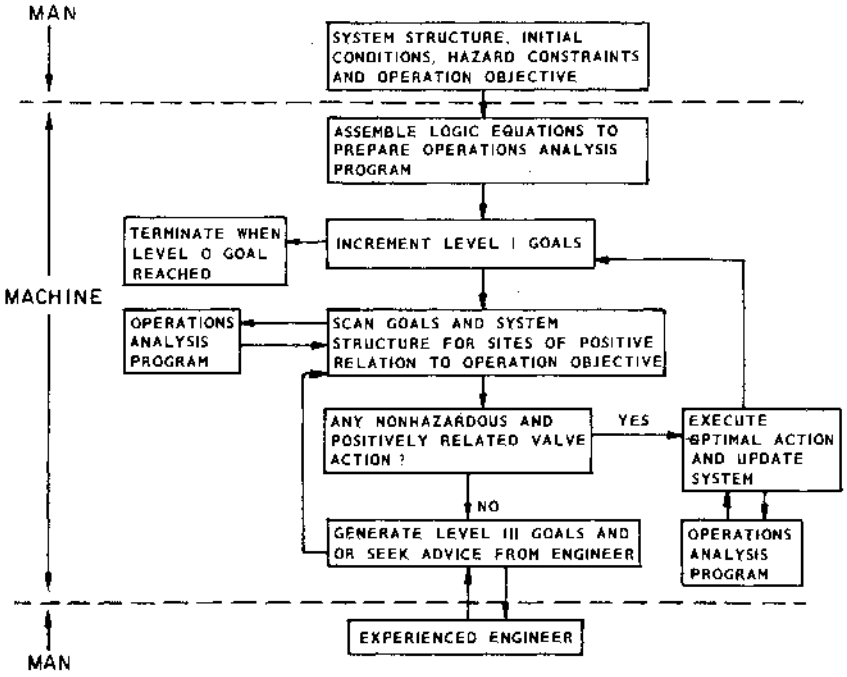

Fig. 7. The program structure of the synthesis.

sequence will not be generated. Either a cycle will be generated with no net operating effect or all valve actions will be deemed hazardous or not positively related. At this point the search must receive guidance.

This happens frequently when the system is filled with hazardous material and the goals do not explicitly mention the removal of the hazardous material. The safety restrictions then block every move the synthesizer proposes. With only the minimum set of goals, the synthesizer has no hint of the importance of a prior purge-operation sequence. Other more complicated examples of situations not involving hazards that stymie the synthesizer can be cited, and they occur in the test problems we examined.

The essential feature is that the synthesizer must be drawn away from the most direct-appearing route to the goals and forced to examine round-about avenues that may involve valving operations not positively related to the minimum set of level II goals. The synthesizer seeks advice in the form of level III goals. The level III

goals point to regions of operation that the synthesizer would normally not examine. Certain of the level III goals are generated automatically by the computer, and others are sought as advice from the engineer.

Should the synthesizer discover that all its proposed operations are hazardous and the hazards involve a species that does not enter into the minimum set of level II goals, the computer will generate its own level III goal of purging this species from the system. These level III goals are added to the existing level II goals and synthesis is attempted. New purge goals may then arise, resulting in a layering of level III goals until synthesis is accomplished or this route to convergence fails. The computer successfully generated level III goals in both the catalyst-regeneration system shown here and in a more complicated problem with a hydrogen-dryer system not described here.

The engineer is a second source of level III goals, as he offers advice to the computer. In Table VI the goals with the asterisk are really level III goals provided by the engineer; they were added in anticipation of problems the computer might have, and others are added when the difficulties occur and the computer seeks help. For example, the additional goal of set 6 indicates that if it is desired to have species 4, inert gas, flowing in connector 10, it would also have to be flowing in connector 22. Of course more information could be provided by stating that inert gas has to be flowing in connectors 13, 17, 19, etc., but the single level III hint was sufficient for the synthesizer.

The additional goal of set 9, which states that the species are to be trapped in the upper header as well as the reactor, makes it possible to evacuate them later. The additional goals of set 10 mean that hydrogen and inert gas have to be evacuated from the upper header to continue with the next set of goals.

The ability to generate and receive level III goals greatly increases the power of the synthesizer, and provides access to the intuitive problem-solving capability of the experienced engineer.

Operating Criteria

A feasible operating sequence is defined as one that accomplishes the operating objective with no hazards. Current programs synthesize feasible sequences.

More elaborate criteria ought to be added to the programs as research progresses, and it is interesting to examine the possibilities. We recognize that hazards are of different magnitudes, and that certain disastrous events ought to be skirted as widely as possible, while others can be cut close. The hazards ought to be ranked according to damage potential and operating sequences synthesized to be the largest number of operating errors away from the highest-ranking hazards.

Another criterion might be that the operating objectives be reached in as few as possible valve operations. The operators then have the fewest possible chances of error.

A more complex criterion is to give an opponent k chances to cause disaster, and at each step of the operation to allow him to use any number of his unused chances in any way. This then becomes a real game of disaster. Investigations along this line ought to give insight into the inherent disaster resistance of industrial systems.

TABLE VII

SUMMARY OF ENGINEER AND COMPUTER-SYNTHESIZED OPERATING PROCEDURES FOR THE CATALYST-REACTIVATION SYSTEM

Goal	Engineer	Computer
Replace reactor in service and stop hydrocarbon flow in the reactor to be reactivated	Open valve 26 Open valve 8 Close valve 14	Open valve 26 Open valve 8 Close valve 14
Purge hydrocarbon from reactor with hydrogen	Open valve 13	Close valve 7 Open valve 13 Open valve 7
Pressurize reactor with hydrogen	Close valve 7 Close valve 25 Close valve 2	Close valve 7 Close valve 13
Depressure and evacuate hydrogen from reactor	Open valve 12 Open valve 17 Open valve 7	Open valve 7
Pressurize reactor with inert gas	Close valve 7 Open valve 20	Close valve 25 Open valve 13 Close valve 7 Open valve 20
Circulate inert gas	Open valve 22	Open valve 12 Open valve 22 Open valve 17
Divert total flow of inert gas to reactor	Close valve 23 Close valve 12	Close valve 12 Close valve 23
Chlorinate and rejuvenate	Open valve 5 Open valve 6	Open valve 5 Open valve 6
Pressurize reactor with inert gas, chlorine and oxygen	Open valve 23 Close valve 22 Close valve 20 Close valve 5 Close valve 6	Open valve 23 Close valve 13 Close valve 17 Close valve 20 Close valve 6
Depressurize and evacuate reactor	Open valve 7	Close valve 2 Open valve 7 Open valve 13 Open valve 25
Pressurize reactor with natural gas	Open valve 25 Close valve 7 Open valve 3	Open valve 3 Close valve 7
Purge reactor with natural gas	Open valve 7	Open valve 7
Stop flow of natural gas	Close valve 3	Close valve 3
Purge with hydrogen	Open valve 2	Open valve 2
Pressurize with hydrogen	Close valve 7	Close valve 7

TABLE VII—*Continued*

Goal	Engineer	Computer
Depressurize and evacuate hydrogen, start hydrocarbon flow in reactivated reactor, stop hydrocarbon flow in the other reactor and leave hydrogen in upper header	Close valve 13 Open valve 7 Open valve 14 Close valve 26 Close valve 8	Close valve 2 Open valve 7 Open valve 2 Close valve 7 Close valve 13 Open valve 14 Open valve 7 Close valve 26
Total number of operations	35	43

Experience on Real Problems

Table VII compares the valving sequence for the catalyst-regeneration system synthesized by the engineers of a major petroleum company to one generated by the computer. Both reach the operating goal, the engineers in 35 moves and the computer in 43. Of primary interest is the speed with which the computer worked, only one minute and thirty-four seconds of UNIVAC 1108 time were required, and just over two minutes were used to synthesize a safe operating sequence for a more elaborate hydrogen-drying system. These methods are feasible for real-time monitoring of industrial disasters.

CONCLUSIONS

Ideally, during a disaster one could call on the computer for a rapid assessment of the situation and for recommendations on strategic courses of action to intercept and quench the disaster. It appears as if this capability is at hand. Given the structure of the system, the hazards to be avoided, the initial conditions, and operating goals to reach, the present programs can synthesize disaster-resistant opera-

TABLE VIII

Summary of Performance on Industrial Problems

	Catalyst-regeneration system	Hydrogen-drying system
Number of connectors	27	47
Number of valves	17	24
Number of species	6	4
Level I goals	16	13
Level II goals	27	16
Level III goals generated	6	6
Number of hazardous conditions	18	17
Number of operations generated by engineer	35	23
Number of operations generated by computer	43	32
Synthesis time	1 min 47 sec	2 min 26 sec

tions. Table VIII shows the magnitudes of the two test problems mentioned here and the speeds of the synthesis program. These are industrially significant problems that were handled at real-time speeds.

ACKNOWLEDGMENTS

THIS RESEARCH WAS sponsored in part by the National Science Foundation, The Shell Foundation, the AMOCO Foundation, and the United States Army. The work was performed at the University of Wisconsin.

REFERENCES

For basic information on the problem formulation:

1. C. L. BROWNING, *Accident Prevention and Loss Control*, American Management Association, New York, 1969.
2. T. A. KLETZ, "Specifying and Designing Protective Systems," Loss Prevention No. 6, *Chem. Engr. Prog. Technical Manual*, 1972.

The logical analysis draws on these references:

3. D. L. DIETMEYER, *Logic Design of Digital Systems*, Allyn and Bacon, New York, 1971.
4. J. R. RIVAS, D. F. RUDD, AND R. KELLY, "Computer-Aided Safety Interlock Systems," *A. I. Ch. E. J.* **20,** 311–319 (1974).

The methods of synthesis are introduced in these references:

5. A. H. NEWELL AND H. R. SIMON, *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, N. J. 1971.
6. J. R. RIVAS AND D. F. RUDD, "Synthesis of Failure-Safe Operations," *A. I. Ch. E. J.* **20,** 319–325 (1974).
7. J. J. SIIROLA AND D. F. RUDD, "Computer-Aided Synthesis of Integrated Process Designs," *Ind. and Engr. Chem. Fund.* **10,** 353–363 (1971).