Generating Behavior Equations from Explicit Representation of Mechanisms

Yumi Iwasaki
June 1987
CMU-CS-87-131

ABSTRACT

The methods of causal ordering and comparative statics provide an operational means to determine the casual relations among the variables and mechanisms that describe a device, and to assess the qualitative effects of a given disturbance to the system. However, for correct application of the method of causal ordering, the equations comprising the model of the device must be such that each of them stands for a conceptually distinct mechanism. In this paper, we discuss the issue of building a model that meets this requirement and present our solution. The approach we have taken for building device models in our domain of a power plant is to represent explicitly one's understanding of mechanisms underlying an equation model as flows of matter and energy. A system was implemented to generate structural equations automatically from this representation. We discuss the results and some of the problems encountered along the way.

This research is supported by the Defense Advanced Research Projects Agency, Department of Defense, ARPA Order 3597, monitored by the Air Force Avionics Laboratory under contract F33615-81-K-1539. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Table of Contents

List of Figures

List of Tables

1. Introduction

The methods of causal ordering [10] and comparative statics [9] provide an operational means to determine the causal relations among the variables and mechanisms that describe a device, and to assess the qualitative effects of a given disturbance to the system. [5, 6, 3]. These procedures, which have been widely used in several fields of sciences, are generally consistent with and somewhat more general than the methods for determining causal relations and for propagating disturbances employed by many researchers of qualitative reasoning. [1, 2, 7, 11]. We have been developing a system to perform qualitative causal analysis of a device behavior based on these methods. For the method of causal ordering to produce the correct causal relations, equations comprising a model must come from understanding of mechanisms. This paper focuses on the issue of building a model that meets this requirement. The approach we have taken to ensure that each equation stands for a distinct mechanism is to represent explicitly one's understanding of mechanisms underlying an equation model. This representation forms, below the level of equation model, another level of model, which represents such understanding and from which equation models are generated automatically. We have developed the representation for mechanisms and a program to generate equation models from the representation.

The next section briefly describes the method of causal ordering. In Section 3, we illustrate the problem of building an equation model. Section 4 presents our representation of mechanisms, and Sections 5 and 6 describe the procedure for generating equation models from the representation. Some results, including some problems encountered, the current status of the system, and future directions are discussed in Section 7. Finally, Section 8 compares our approach to some of the others in the field.

2. Causal Ordering

Causal ordering [10] is an asymmetric relation among the variables and equations of a set of simultaneous equations. The idea of causal ordering in a system of equations can be described roughly as follows. A system of **n** equations is called self-contained if it has exactly **n** unknowns. Given a self-contained system, **S**, if there is a proper subset, **s**, of **S** that is also self-contained and that does not contain a proper self-contained subset, **s** is called a minimal complete subset. Let S_0 be the union of all such minimal complete subsets of **S**; then S_0 is called the set of minimal complete subsets of zero order. Since S_0 is self-contained, the values of all the variables in S_0 can, in general, be obtained by solving the equations in S_0. By substituting these values for all the occurrences of these variables in the equations of the set (S _ S_0), one obtains a new self-contained structure, which is called the derived structure of first order. Let S_1 be the set of minimal complete subsets of this derived structure. It is called the set of complete subsets of 1st order. Repeat the above procedure until the derived structure of the highest order contains no proper subset that is self-contained. If one denotes by V_l the set of variables in the complete subsets of lth order, where l #≥# 0, then the variables in V_l, ($l > 0$), are said to be *directly causally dependent* on the elements in V_{l-1}.

Equations comprising a model come from an understanding of mechanisms. The term *mechanism* is used here in a general sense to refer to distinct conceptual parts in terms of whose functions the working of the whole system is to be explained. Mechanisms are such things as laws describing physical processes or local components that can be described as operating according to such laws. An equation representing such a mechanism is called a *structural equation*, and every equation in the model should be a structural equation standing for a mechanism through which variables influence other variables.

One thing to note about the method of causal ordering is that it does not require knowledge about the precise functional forms of equations. The only information that the method makes use of is what variables appear with a non-zero coefficient in what equations, which in terms of mechanisms translates to what variables are causally linked by each mechanism. In fact, for the discussion of this paper, it makes no difference whether or not a model is qualitative. Therefore, in the remainder of this paper, whenever a functional form appears, it will not affect the discussion if the functional form is replaced by a more qualitative expression of the functional relation.

3. Modeling

Describing a system in terms of the mechanisms that determine the values of the variables is fundamental to causal analysis. In order to apply the method of causal ordering to determine the causal structure in a model, each equation in the model must be a structural equation. Unfortunately, there is no simple formal answer to the question of how to know that an equation is structural.

Let us illustrate with a simple example of a condenser model how a choice of equations affect the causal ordering produced. A condenser has inputs of steam (STM.in) and cooling water (CWT.in) and outputs of condensed steam (STM.out) and warm cooling water (CWT.out) Only considering mass flow for now, one can write the following equations for the condenser. M_x stands for the mass of x, where x is one of the inputs or outputs of the condenser. c_1 and c_2 are some constants.

1. The overall conservation of mass law.
   $M_{CWT.in} + M_{STM.in} = M_{CWT.out} + M_{STM.out}$
2. Steam flow
   $M_{STM.in} = M_{STM.out}$
3. Cooling water flow.
   $M_{CWT.in} = M_{CWT.out}$
4. $M_{CWT.in} = c_1$
5. $M_{STM.in} = c_2$

The last two equations represent the assumption that the variables, $M_{STM.in}$ and $M_{CWT.in}$, are exogenous. It can be seen easily that the above set of equations is redundant. The equation (1) is a linear combination of (2) and (3). Each of the following subsets of the above five equations is a self-contained structure, but each will give rise to a different causal ordering. The equations themselves do not tell us which one of these three sets should be selected as the model for the situation.

1. The equations (1), (2), (4) and (5),
2. The equations (1), (3), (4) and (5),
3. The equations (2), (3), (4) and (5).

The problem here is that the equations above are not all structural equations. Concerning ourselves only with mass flows for now, we know there are two distinct mechanisms in the above situation, namely the flow of steam and the flow of cooling water. The equation (1) is clearly not a structural equation, because it mixes up the two mechanisms.

The above example is a very simple case, where the number of equations exceeds the number of variables only by one, and it is not difficult to see which equation is the one that does not correspond to a distinct mechanism. However, in general, given a device, the number of equations one can write down about it far exceeds the number of variables. We must choose from this large set of equations only those equations that reflect our understanding of mechanisms operating in the situation to produce the correct causal structure. The question naturally arises how one can make sure that the equations included in a model are structural equations. If structural equations come from understanding mechanisms, can one build a model that represents one's understanding of mechanisms explicitly, and, furthermore, can one systematically generate structural equations from the representation? The next section presents such an explicit representation of mechanisms that we chose for our domain of a power plant.

Before going on to describe our representation, a few more words need to be said about the concept of mechanism. Causality arises when a mechanism links phenomena. Since the types of phenomena one is concerned with vary from domain to domain, the types of mechanisms that are of interest vary also. If the mechanisms is a lever, the phenomena are the movements of the bodies resting on its arms. If the phenomena are the weather during a growing season and the size of wheat crop, then the mechanism is the function that determines the production of wheat. In our domain of a coal power plant, the phenomena of interest are mass and energy inputs and outputs, the mechanisms are the processes that produce outputs from the inputs. Furthermore, in this paper we are concerning ourselves with only equilibrium models. The relation between static and dynamic models, and that between causal ordering and dynamic models are important issues to consider and are currently being investigated. But these issues are beyond the scope of this paper.

4. Network Representation of Processes

We use a network representation of processes for the purpose of explicitly representing one's understanding of mechanisms underlying an equation model. A process network is a semantic net representation of active processes taking place in a device. Intuitively, processes are things that mechanisms make happen in a device and give rise to the overall behavior of the device. Since we are working in the domain concerned with flows of matter and energy, we represent processes in a device as flows of matter and/or energy which together are responsible for the overall input-output behavior of the device.

Figure 4-1 shows the process network for a turbo-generator. Part of the energy of the super-heater steam that goes into the turbo-generator is used to produce electricity, and the steam comes out with a lower temperature. A non-negligible fraction of the energy is also lost into the atmosphere as heat radiation. There are four distinct processes happening in the turbo-generator, namely the steam flow (denoted MF_{stm} in the figure), the flow of energy of the steam accompanying the steam flow (EF_{stm}), generation of electricity from the input steam energy (EF_{ele}), and heat loss into the atmosphere (EF_{htl}).

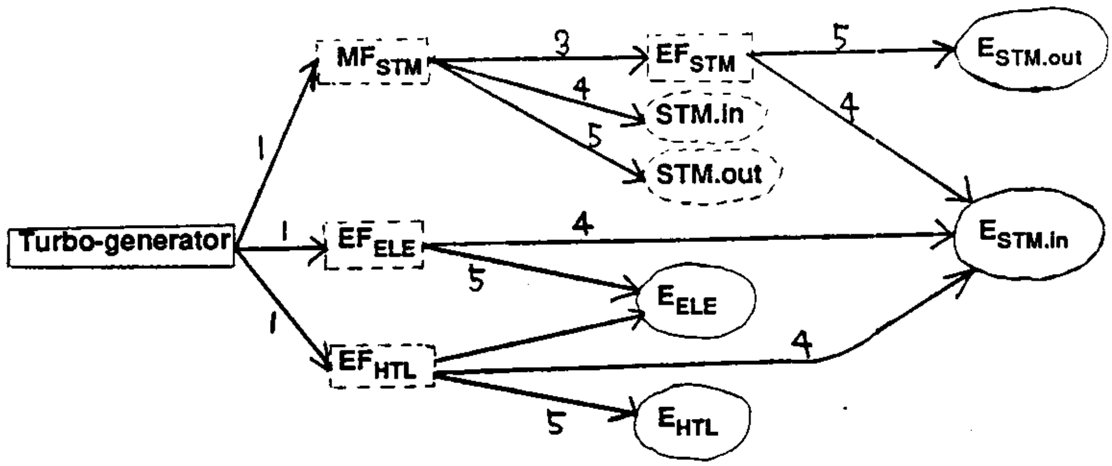
4.1. Nodes

A process network contains the following four types of nodes, namely *device, material, energy*, and *process.*

Device — A device node represents the device whose behavior is described by the network.

Material — A material node represents a conceptually distinct body of matter such as some type of input and output matter of a process.

Energy — An energy node represents a conceptually distinct body of energy, which is an input or output of a process. The internal energy of matter is represented as an energy node separate from the matter itself. Since material always has internal energy, a



- Nodes

dotted rectangle	process node	
	MF	material flow
	EF	energy flow
solid rectangle	component node	
dotted ellipse	material node	
solid ellipse	energy node	

- Subscripts

STM	steam
ELE	electricity
HTL	heat lost by radiation

Figure 4-1: The process network for a turbo-generator

material node must always have an associated energy node.

Process A process is modeled as a flow. There are material flows and energy flows.

A process may be complex or simple: A complex process involves one or more other processes. A simple process involves no other processes. A process is said to *involve other processes* when the latter processes are integral parts of the former and can only exist as such. Since a body of material must have internal energy, a flow of material implies a flow of energy also. Therefore, a material flow is never a simple process as it always involves an energy flow. A complex process forms a tree with the processes that it involves, the processes that they in turn involve, and so on. We will refer to such a tree as a *process hierarchy*, to the complex process at the root as a *top-level process*, and to other processes in the tree as *subordinate processes*.

4.2. Links

Nodes in a process network are connected by the following six types of links; *has-process, involves-material-flow, involves-energy-flow, source, destination,* and *internal-energy.* Numbers correspond to the link numbers in the figure 4-1.

Has-process (1)
: A device node has has-process links to process nodes in the network. When some of the processes form a process hierarchy, the device has a link only to the top-level process of the hierarchy.

Involves-material-flow (2), involves-energy-flow (3)
: Processes in a process hierarchy are connected by these links. A complex process has this type of links to its subordinate processes.

Source (4), destination (5)
: Processes must have a source and destination. A flow node has a source link and a destination link to material or energy nodes.

Internal-energy (not shown in the figure to avoid visual clutter)
: A body of matter and its internal energy are represented as separate material and energy nodes, where the material node must have an internal-energy link to the energy node.

5. Generating Variables

As stated in the previous section, the purpose of a process network model is to represent explicitly one's understanding of processes giving rise to the behavior of a device so that an equation model can be constructed systematically.

Equations express aspects of overall behavior in terms of variables. Process network models do so in terms of material and energy flows. Thus, before an equation model can be generated from a process network model, variables must be generated first.

Variables represent attributes of such entities as material, device or process. They are some directly or indirectly measurable quantities about these entities, and the behavior of a component is to be expressed in terms of these quantities.

Given a process network, relevant variables for each node in the network are automatically created. The information about what types of variables must be created for material nodes resides in the knowledge base, which contains general knowledge about different types of matter. For example, the knowledge base contains knowledge about steam that each instance of steam has the following attributes that are relevant to its behavior; mass, pressure, temperature, steam quality and internal energy. The system creates a variable for each of them.

5.1. Efficiency of a process

Besides such physical quantities as energy, mass, temperature, etc. mentioned above, there is another quantity needed to describe behavior, the *efficiency* of a process. A top-level process has an efficiency variable, which represents this measure of efficiency of the process. The word, *efficiency* is used here to mean *some* measure of how much of the quantity at the source of the flow (mass or energy, depending on the type of flow) actually flows to the destination. The exact significance of the efficiency variable of a process must be defined appropriately for each type of process. The efficiency variable in many cases is defined as the ratio, D/S, where D and S are the quantities (mass or energy) at the source and destination of the process.

For example, in the turbo-generator example of Figure 4-1, the efficiency variable, R_{HTL}, of the heat-loss process is defined as;

$E_{HTL} / E_{STM.in}$

where E_{HTL} and $E_{STM.in}$ are the energies lost to the atmosphere and the internal energy of the input steam respectively.

Efficiency variables are employed so that when the system later generates behavior equations from a process network, the flow-rate of a flow can be expressed in terms of the quantities (mass or energy) at the source and the efficiency variables of the process. See the discussion on generation of flow equations in Section 6.2.

Only a top-level process can have an efficiency variable and it cannot have more than one. Therefore, the efficiency variable of a complex process must be defined in such a way that it is possible to express efficiencies of all the subordinate processes using the efficiency variable of the top-level process. If a process or a process hierarchy needs two independent efficiency variables, it must actually consist of two distinct processes or process hierarchies. In other words, one process can add only one degree of freedom to the model i.e. the system of equations, and the efficiency variable represents the degree of freedom introduced by the process.

5.2. Active and Passive Processes

Earlier, it was stated that each independent process has one and only one efficiency variable, and that efficiency variables of processes in a device are independent of each other as the top-level processes are independent. This is true in principle when each process is considered in isolation. However, where several independent processes interact as in a device, the situation becomes a little more complicated. The way in which independent processes interact in our representation is by sharing a common source. When there are multiple flows going out from one source, the flows are independent in the sense that any one of them can exist regardless of the existence of others, but their efficiencies are constrained by the fact the they share the same source. In such a case, it is more convenient to make the efficiency of one of the processes a function of the efficiencies of others. As the number of variables in a model represents the degrees of freedom of the model, expressing the efficiency of one of these processes as a function of the others instead of as a separate variable decreases the degrees of freedom by decreasing the total number of variables. That some of the processes share the same source is a constraint that should decrease the degrees of freedom of the model by one.

We will refer to a flow as *passive* if its efficiency is treated as a function of the efficiencies of other flows sharing the same source, and to others as *active*. A passive flow is the one perceived as the "left over" flow, meaning the flow of whatever has not flowed into other directions. A passive flow is not given an efficiency variable of its own, but its flow rate is expressed in terms of those of all the other active flows sharing the same source as :

The flow rate = the quantity at the source - (sum of the flow rates of all other flows).

For example, in the turbo-generator network of Figure 4-1, the energy flows, EF_{HTL}, EF_{STM}, and EF_{ELE} share the same source, namely the internal energy of the input steam, $E_{STM.in}$. Among the three energy flows, one of them is treated as the passive one, and the the system does not create an efficiency variable for it. The decision as to which one of them to treat as the passive one is left to the model builder, who must make the decision based on his mental model of the situation.

The distinction of active and passive flows is a subjective one. The system currently has no heuristics for deciding which process among several sharing the same source to regard as the passive one. However, it may be possible to make this decision automatically if more detailed structural knowledge of the device is available.

6. Generating Equations

Once variables are generated, the system can construct an equation model by generating behavior equations. The system generates the following three types of equations from the network:

Energy equations — Compute the internal energy of material nodes.

Flow equations — Describe flow of material and energy.

Equations about characteristics — Describe behavioral characteristics of different kinds of material, processes or subcomponents.

Generation of each type of equations is described below.

6.1. Internal energy equations

The knowledge base contains information about how to compute the internal energy of each type of material. For each energy node which is the internal energy of material of type, M, the general formula for computing the internal energy of M is instantiated for the particular instance of M. For example, the formula for computing the internal energy of coal is;

internal-energy = heat-value * mass.

From this formula, an actual equation is generated for a particular instance of coal by substituting in its variables;

6.2. Flow equations

For each material node that has n (>0) incoming flows, flow equation is generated, in the form of:

$M = \sum_{i=1}^{n} flow\text{-}rate_i$

where M is the mass variable of the node, and $flow\text{-}rate_i$ ($1 <= i <= n$) are expressions for the flow rates of the flows into the node.

An expression for the flow rate of a flow, F, is generated as follows:

If F is an active flow (i.e. has an efficiency variable),
"M_S * $EFFICIENCY_F$"
else
"$M_S - \sum_{j=1}^{s} flow\text{-}rate_j$",

where M_S is the mass variable of the source of F, s is the total number of flows going out of the source, $EFFICIENCY_F$ is the expression for the efficiency of the process F, and $flow\text{-}rate_j$'s ($1 <= j <= s$) are the expressions for the flow rates of the flows going out of the source. Note that the above procedure will not terminate unless all but one of the flows sharing the same source are active processes.

6.3. Characteristics of materials, processes, and subcomponents

These equations describe characteristics of different types of material, processes or subcomponents. For example, for gas, a certain relation holds between its pressure and temperature. For a conductive heat transfer process, a functional relationship holds between the heat transfer rate and the temperature difference between the heat source and sink. Equations representing control exercised over variables by a subcomponent fall in this class, also.

These equations are generated in much the same way as the internal energy equations. The knowledge base contains knowledge about functional relations that must hold among variables based on physical properties of matter, processes, or devices. These functional relations are instantiated for each instance to generate actual equations.

7. Results and Discussions

We have built six process network models of various power plant components, with which we tested the program to generate equation models. Table 7-1 summarizes these results. It shows the numbers of independent processes and flows in each network model and the total numbers of parameters and equations in the equation model generated.

Device	Independent Processes	Flows	Parameters	Equations Generated
Power Plant	6	13	25	26
Boiler	5	12	26	27
Condenser	3	5	17	17
Feedwater Pump	3	5	20	20
Turbo-generator	3	4	13	14
Pressure Regulator	1	2	9	9

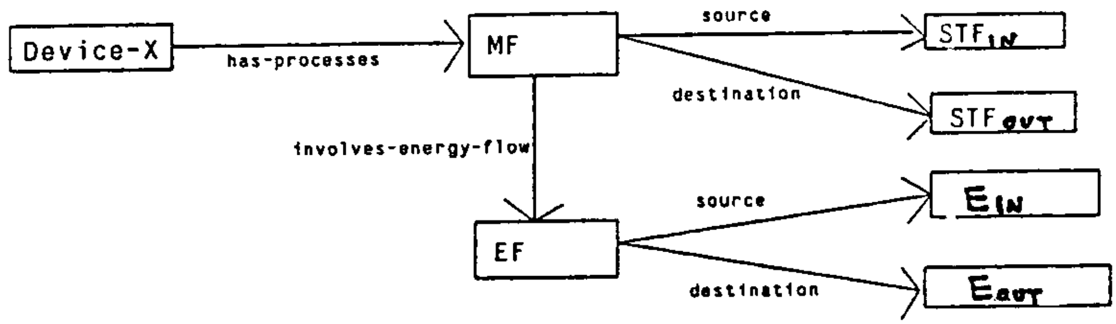
Table 7-1: Equation Models Generated

Contrary to our expectation for the equation models produced to be self-contained, in three out of the six cases, the number of equations exceeded the number of variables by one. Unless the situation modeled is a physically impossible one, the procedure should never produce a truly over-constrained system because it only produces equations that are true in the situation depicted. By examining the three cases, we discovered two problems that caused the number of equations to exceed the number of variables. In two cases, the power plant and the boiler, the equations produced were not linearly independent. In the third, the turbo-generator, we discovered that one of the equations in the model representing an external control should belong to a model at a higher level of aggregation than the level at which the rest of the model was described, and that its inclusion at the current level lead to the model being over-constrained. We discuss these problems in detail below.

7.1. Linearly dependent equations

The first case we consider is a case where the equations are not independent. Consider a device (e.g. a piece of pipe), device-X, through which some substance, STF, flows. Figure 7-1 shows the process network for the situation. The device has one process, MF, which is a material flow with an input, STF_{in}, and an output, STF_{out}. EF is the energy flow accompanying the material flow.

Suppose that the internal energy of the substance is a function of its mass only, say h * mass, where h



MF	Material flow of STF.
STF_{in}	Input STF to Device-X.
STF_{out}	Output STF of device-X.
EF	Energy flow accompanying MF.
E_{in}	Internal energy of STF_{in}.
E_{out}	Internal energy of STF_{out}.

Figure 7-1: Process network for device-X

is a positive constant representing the heat value of the substance. Then, the system will produce the following equations, if we assume the mass of STF_{in} to be exogenous. M_{in} and M_{out} are mass variables of STF_{in} and STF_{out}, and c is some constant.

1. $M_{out} = M_{in}$
2. $E_{out} = E_{in}$
3. $E_{in} = M_{in} * h$
4. $E_{out} = M_{out} * h$
5. $M_{in} = c$.

This set of equations has five equations in four variables. However, observation of the equations reveals that equations 1 through 4 are not linearly independent, as the equation 2 can be derived from the equations 1, 3, and 4.

This problem arises for the following reason. The decision to represent material flows and energy flows separately in a process network was made based on the assumption that the two are independent. This assumption is generally true in our domain. However, cases where the energy flow is solely dependent on the material flow do arise sometimes, and in these cases the system produces redundant equations. A coal flow is an example of a such flow.

This problem of redundant equations does not pose a fundamental difficulty to our approach as it can be remedied by adding a checking step for this dependence when generating an energy flow equation. However, what is to be learned from this mistake is that when this network representation is generalized to include other types of mechanisms (those which may not be conveniently modeled as flows), it will be very important to make explicit the assumptions, such as that energy and mass flows are independent in the present case, which go into making fundamental design decisions. One must keep these assumptions in mind when constructing a procedure for generating equations from the representation.

7.2. Equations of different aggregation levels

The second case where more equations than the number of variables are produced is when equations of different aggregation levels are included in a model. In particular, if control functions that take relatively long time to reach equilibrium are included in a model along with equations that describe functions that reach equilibrium in a much shorter time, one will end up with too many equations. Here, the situation is not really over-determined, but the model given by the model builder is incorrect as it confuses levels of abstraction.

To illustrate this problem, consider the following example of a device, device-Y, through which steam flows. The output steam is wet steam, STM_{out} with variables, mass, temperature, pressure, energy, and steam quality, denoted by M, T, P, E, and X. (Steam quality is the ratio of steam to the condensed water in a mixture of steam and water.) For simplicity, I omit the description of inputs of the device and focus on just the output of device-Y, treating E and M as exogenous variables. Then, the equations for this part of the model are as follows, where f_i (i = 1, 2, or 3) denote some monotonically increasing functions of the arguments, and c_1 and c_2 are constants. This set of equations is self-contained.

1. $E = f_1(X, T, P, M)$
   The internal energy of a body of wet steam is a function of its steam quality, temperature, pressure and the quantity.
2. $T = f_2(P)$
   The temperature of the steam is a function of the pressure.
3. $P = f_3(T, X)$
   The pressure of the steam is a function of the temperature as well as the steam quality.
4. $M = c_1$
   M is exogenous relative to this part of the model.
5. $E = c_2$
   E is exogenous relative to this part of the model.

Now if one adds to the model that the pressure P is maintained constant by some means of control, adding the equation

6. $P = c_3$

to the set, the set of equations becomes over-constrained. What is the problem here? Is this a physically impossible situation though it seems possible to control pressure by some external means?

The cause of the difficulty here is that the control represented by the equation 6 actually cannot be accomplished independently of other variables. The only way to maintain P constant despite disturbances in inputs is to detect such disturbances and make adjustments to keep P constant. For example, one could have a feedback control mechanism that senses a disturbance in in P and adjusts M to compensate for it, in which case the equation 4 must be replaces by

4'. $M = f^{-}(P)$,

where f^{-} is a monotonically decreasing function of P. Even with this replacement, the equations 1, 2, 3, 4', 5, and 6 do not form a self-contained set.

The control represented by the equation 6 is accomplished over time by the feedback mechanism, which brings the system back to equilibrium some time after a disturbance. In other words, the added equation is only true in a more macroscopic view of the behavior, and therefore it belongs to a more aggregate model than the model presented here. At a more abstract level, one may regard P to be constant as in equation 6, ignoring the details about how the control is accomplished. At that level, the equation 3 must be replaced by the equation 6 because the equation 3 belongs to the detail about how the control represented by the equation 6 is accomplished.

Models at various levels of aggregation are certainly necessary to deal with a complex device. When building a behavior model, one must be careful not to mix up equations of different abstraction levels as such mistakes are in fact very easy to commit.

7.3. Current Status

After the problems discussed in the previous section were fixed, our system produced self-contained equation models for the six process networks. We have also written a program to apply the method of causal ordering to the equation model thus generated to produce the causal dependency structure in the model. The causal dependency structure produced by this program from the equation models were in good agreement with our coal power plant expert's view of causal relations among the variables involved.

As a solution to the problem of building a behavior model consisting of structural equations, our approach of explicitly representing processes from which equations are generated automatically seems successful. Though the usefulness of modeling processes as flows is limited to domains mainly concerned with flows of various things, they include many important domains. Also, this approach of automatically generating equilibrium models from explicit network representation of processes viewed as mechanisms acting on inputs to produce outputs seems generalizable to other domains dealing with other types of processes, though it remains to be demonstrated.

We have also implemented a program to assess the qualitative effects of a given disturbance on variables without the capability to determine the stability of a system involving a system. A behavioral model produced from a process network as discussed in this paper is an equilibrium model. The methods of causal ordering and comparative statics provide an operational means to determine the causal relations among the variables in such a model and to assess the qualitative effects of a given disturbance to the system. However, if the situation modeled involves feedback, before one could determine the system's response over time to a given disturbance, one must determine stability of the dynamic behavior of the system before such an assessment can be made. We are studying ways to incorporate into the system the capability to qualitatively analyze the stability of a dynamic model.

8. Related work

The concept of mechanisms that give rise to causality by linking phenomena is similar to the concept of physical processes in Qualitative Process Theory (QPT) by Forbus [4]. In QPT, a physical process is "something that acts through time to change the parameters of objects in a situation." A process is specified by a set of objects involved, conditions for the process to happen, relations imposed by the process on parameters, and influences of the process on the objects. Given a state description, QPT infers what processes are in progress and based on that it predicts what will happen in the future. We have not tried to develop a general ontology of physical processes because we start from the knowledge of mechanisms (or processes) in place and do not try to figure out what processes are giving rise to the behavior. The goal of QPT is to understand common sense physical reasoning while our aim to investigate a more formal approach to causal qualitative reasoning about the behavior of a system that may not reflect a lay-person's naive view of the world but that nevertheless appeals to the intuition of a trained person with knowledge of the domain.

In the work on Qualitative Physics by de Kleer, Brown, and Williams [1, 2, 11] and also in the work on Qualitative Simulation by Kuipers [7, 8], the behavior model is given by a collection of functional relations (constraints) among variables as it is in our approach. However, the their constraints come from the knowledge of the topological structure of the device, and the notion of causality in qualitative physics is based on the way disturbance propagates through a confluence network. Despite these seeming differences, our approaches using the method of causal ordering and comparative statics is consistent with the method of qualitative physics for determining instantaneous behavior. The equations in our model come from the working of distinct mechanisms, which represent specific connections among components, and which is a notion that includes but is more general than the physical connections, which equations are based on in Qualitative Physics. In construction of a behavior model, de Kleer and Brown postulate a locality principle and a no-function-in-structure principle to avoid using non-local and teleological knowledge as much as possible to make sure that their description of a device is free from any pre-conceived notion of what their behavior ought to be. We believe that the distinction they draw between a "neutral" description free of such pre-conception and one that is not is only a difference between abstraction levels. We believe that non-local and teleological knowledge is not only useful but is necessary in building a model of a complex device. For more discussion on the differences and similarities of the two approaches, see [5, 6, 3].

Acknowledgment

The author would like to thank Professors Herb Simon and John McDermott for their guidance in conducting this research and also for their helpful comments on the paper. Professor Holger Sommer of the Department of Mechanical Engineering, University of Maryland, provided me with knowledge of coal power plants.

References

[1] De Kleer, J., and Brown, J.S.
Qualitative Physics Based on Confluences.
Artificial Intelligence 24, 1984.

[2] De Kleer, J.
How Circuits Work.
Artificial Intelligence 24, 1984.

[3] de Kleer, J., and Brown, J.S.
Theory of Causal Ordering.
Artificial Intelligence 29, 1986.

[4] Forbus, K.
Qualitative Process Theory.
Artificial Intelligence 24, 1984.

[5] Iwasaki, Y., and Simon, H.A.
Causality in Device Behavior.
Artificial Intelligence 29, 1986.

[6] Iwasaki, Y., and Simon, H.A.
Theory of Causal Ordering: Reply to de Kleer and Brown.
Artificial Intelligence 29, 1986.

[7] Kuipers, B.
Commonsense Reasoning about Causality: Deriving Behavior from Structure.
Artificial Intelligence 24, 1984.

[8] Kuipers, B.
Qualitative Simulation.
Artificial Intelligence 29, 1986.

[9] Samuelson, P.A.
Foundations of Economic Analysis.
Harvard University Press, Cambridge, MA, 1947.

[10] Simon, H.A.
Causes and Possible Worlds.
Boston Studies in the Philosophy of Science. Volume 54.*Models of Discovery: and other topics in the methods of science.*
D. Reidel Publishing Company, 1977, Chapter 2.1 - 2.4.

[11] Williams, B.C.
Qualitative Analysis of MOS Circuits.
Artificial Intelligence 24, 1984.