Diagnostic Mechanism Modeling

Arthur M. Farley

CIS-TR-87-08
May 13, 1987

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

Diagnostic Mechanism Modeling

Arthur M. Farley*
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403

Abstract

Diagnostic models are designed specifically to support diagnostic reasoning. We first place diagnostic reasoning within the framework of troubleshooting, the process whereby an incorrectly functioning system is restored to normal function. Diagnostic reasoning determines a set of component faults that can account for observed abnormalities in system function. As such, diagnostic models must incorporate elements of function as well as behavior. We propose that function be introduced in the composition of component behaviors. We discuss two general principles of diagnostic modeling: model variables assume values relative to normal (the "normality principle") and these values are propogated to account for the production of single outputs (the "single output principle"). These two principles yield significant simplifications in the value spaces of model variables and in model structure. We illustrate our approach to diagnostic modeling and reasoning with examples from xerography.

Subject Categories: diagnostic reasoning, qualitative modeling

---

* Research performed while author on sabbatical leave at Xerox PARC, Palo Alto, CA; research supported by ARI Contract #MDA 903-83-C-0189.

Introduction

We have come to depend increasingly upon complex mechanical and electronic systems in our daily lives. Both the home and workplace are filled with these tools, ranging from automatic appliances in the kitchen and entertainment systems in the livingroom to xerographic copiers and computer networks in our business environments. With this expansion, maintenance of such systems has become a critical issue. Due to the rapid turnover in products and field personnel, there is a perennial shortage of available people who are knowledgeable about the troubleshooting of these complex systems. This situation leads to long repair delays, with expensive maintenance contracts and service calls.

In the research reported here, we investigate possible roles for computer-based modeling and simulation in the maintenance of complex mechanical and electronic systems. How can the general approaches and tools of modeling and simulation be tailored to the task of efficient, effective troubleshooting? We want models that can assist us either directly, by solving particular maintenance problems when they arise, or indirectly, as assistants or training tools for repair personnel and system users. What are important elements of these diagnostic models of complex systems?

In the next section, we place diagnosis and diagnostic reasoning within the context of the more general activity of troubleshooting. We then introduce an example domain for the consideration of diagnostic reasoning, that of troubleshooting the xerographic subsystem of a copy machine. This is followed by presentation of our approach to diagnostic modeling, based upon an informal consideration of several, expert troubleshooting protocols and general modeling goals. We illustrate our scheme through a model of a component of the xerographic system. Finally we demonstrate diagnostic reasoning based upon our diagnostic model and discuss shortcomings of and other issues raised by our modeling approach.

Troubleshooting and Diagnosis

Troubleshooting refers to the process whereby an incorrectly functioning complex electronic or mechanical system is restored to correct function. An incorrectly functioning system is one that is exhibiting behavior not in accordance with its functional specifications. Troubleshooting is an example of behavior known *problem solving*, defined to be the satisfaction of a goal state through elimination of differences observed between a goal state and a current, undesirable state. Furthermore, troubleshooting is an instance of indirect problem solving, as the difference to be eliminated, that of incorrect system function, can not be altered directly. Rather, we must attribute the source of error to one or more failing components of the system and replace or adjust the believed culprit(s).

Problem solving can characterized as taking place within the framework of a problem space. A *problem space* consists of a set of states, representing situations in the task environment, representations of an initial state and goal space, being elements of the state space, and a set of operators representing actions that can be taken within the environment, where each operator is a function mapping one state to another (Newell and Simon, 1973). A problem space provides a representation for a given problem, within which problem solving, characterized as search, can proceed. A sequence of operators that changes an initial into a goal state constitutes a problem *solution*.

In an earlier report, we describeded troubleshooting as taking place within or across three, interrelated problem spaces, as depicted in Figure 1 (Farley, 1985) . *Observation Space* represents problems associated with acquiring relevant observations of system behavior and interpreting these as function-related symptoms. *Diagnosis Space* is concerned with the reasoning that maps symptoms into beliefs regarding possible faults that could be responsible for observed symptoms; in Diagnosis Space, we also determine further symptoms of interest by noting those that have not yet been observed and are associated with one or more of the possible faults. Tests that evaluate the existence of these symptoms become solutions to problems solved in Observation Space through observation acquisition and interpretation. Finally, *Repair Space* is where adjustment and replacement plans are determined that can eliminate probable faults.

Each problem space has its own set of operators, ranging from those elements of test and repair procedures in Observation and Repair Space, which directly measure or manipulate physical components of the system, to the more cognitive style operators of Diagnosis Space, which update degrees of belief in possible faults and generate new, relevant symptoms for evaluation. The different problem spaces do share certain aspects of their state representations. These shared elements form the bridges between the problem spaces, as shown in Figure 1, allowing the troubleshooting process to make transitions between the different reasoning realms when appropriate. Effective problem solving in each of the problem spaces requires its own specialized knowledge. This segmentation of applicable knowledge is one motivation for our characterization of troubleshooting in terms of multiple problem spaces. A knowledge-based system for troubleshooting can focus on one or more of these problem spaces, thereby providing a framework for the limited, orderly development of these complex software systems.

The representation of knowledge for Diagnosis Space seems particularly well-suited to our modeling and simulation perspective. Within Diagnosis Space we are concerned with the transformation of observed symptoms into beliefs regarding the operational state of system components and from those beliefs to expected symptoms. It is through this reasoning that we come closest to simulating system activity, imagining the propagation of local effects of component faults through neighboring components to observable features in system behavior. Let us consider an example to illustrate our point, by attempting to account for the occurrence of light copies as produced by a defective xerographic copier.

An Example: Xerography

In our consideration of copy production, we focus on the xerographic subsystem of a copy machine, as summarized in Figure 2. There are seven stages that occur in the xerographic process, each realized in a copier by one or more components; representative components for the various stages are indicated below the schematic. A central element of the process, acted upon by components from six of the stages, is the photoreceptor. A photoreceptor has the unique property that it will hold a local electrical charge until it is exposed to light; importantly, only those areas exposed to light are then grounded and lose their charge. This allows the photoreceptor to hold a pattern of charge representative of the light pattern to which it has been exposed. The discussion below will remain at a qualitative level, ignoring certain details of the physical processes involved in xerography.

During the xerographic process, a photoreceptor is first given a uniform charge by being passed through a uniform negative field emitted from a dicorotron. It is then exposed to a light image, reflected from and thus patterned according to the copy to be made. During the development stage, the charged image on the photoreceptor is brought into close contact with small, oppositely charged particles of plastic, which adhere to the image. The pattern of charged plastic on the photoreceptor is then brought near a paper sheet in the presence of another, stronger field, at which time most of the particles transfer to the sheet. After the transfer stage, the sheet is passed through hot rollers, fusing the plastic to the paper and producing the final copy. The remaining charge and plastic are removed from the photoreceptor during the cleaning stage. The photoreceptor is then ready to begin the process anew with the charging stage.

Now suppose we observe copies being produced which are too light. How can we remedy this situation? As noted, there is no direct way to eliminate the observed difference -- no action in the world which is "make darker copies". During troubleshooting, we must alter, adjust, or replace one or more components to correct the performance of the system as a whole. Determining which component to affect is the task carried out primarily in Diagnosis Space, by accounting for observed symptoms in terms of faulty components. There are several possible accounts for light copies; we discuss three of these below.

One account would have the charge field emitted by the dicorotron at the transfer stage being too low. This would result in too many of the plastic particles remaining on the photoreceptor during transfer. Thus, too few particles would be transfered and subsequently fused to the paper; the final copy would be light. Another account would have the charge field emitted by the dicorotron at charging being low. This would result in an insufficient number of plastic particles adhering to the photoreceptor during development as the charge on the image would be weak. This would propagate through latter stages of the xerographic process as too few particles being transfered and eventually fused to the paper. Finally, suppose the photoreceptor has become fatigued, no longer holding a charge as it should. An account similar to the charge dicorotron story would be the result. As all photoreceptors fatigue with use, more advanced copiers attempt to compensate by automatically increasing the field emitted at charging as fatigue is detected, thereby prolonging adequate copy contrast. All of the above stories accounting for light copy reflect a qualitative simulation of the xerographic subsystem and its process.

The three accounts above illustrate our notion of diagnostic reasoning based upon a model of a complex system. A fault in a component is seen to give rise to a local effect, which is then propagated by a form of simulation through other, connected components to produce eventual predictions as to observable symptoms. Such a model should likewise support a form of backward reasoning or reverse simulation from observations to possible faults, allowing us to determine a set of faults whose local effects could propagate to produce observed symptoms. Our goal here is to present a scheme for the definition and use of qualitative system models sufficient to support these simulation-based forms of diagnostic reasoning.

In the next section, we discuss general properties of such diagnostic models, both necessary aspects and possible simplifications. We follow this by a formal specification of our modeling scheme, based upon the properties discussed. We then define a process of diagnostic reasoning that makes use of our diagnostic models. We illustrate our notions with a diagnostic model of aspects of the xerographic subsystem of a copier as discussed above. We conclude with a discussion of the possible application of our modeling and reasoning schemes to actual troubleshooting situations and field personnel education.

Diagnostic Modeling

By *diagnostic model* we will mean a model of a mechanism that can serve as basis for diagnostic reasoning about that mechanism. First, a diagnostic model must achieve a significantly high degree of architectural fidelity and functional adequacy. *Architectural fidelity* is a measure of the extent to which a model incorporates structural and process elements from the design and operation of a complex mechanism. A diagnostic model must directly reflect a system's structure by representing relevant components and their interconnections. This is necessary as diagnostic reasoning must isolate faults to components within this architecture. As for process elements, we wish to generate causal accounts of how symptoms could arise from possible faults and the propagation of their local effects through neighboring system components. In other words, we want to represent the performance of a complex system in terms of the composition of locally-determined, component behaviors. One aspect of satisfying this goal is the *locality principle* (deKleer and Brown, 1984), which requires that a component's behavior be modeled solely in terms of values associated with its own state and those of its connection interfaces. The composition of component behaviors will reflect not only the structure of component interconnections but also the time course of the system's process, capturing temporal relationships among component behaviors (e.g., sequential, concurrent, simultaneous).

Recent research on qualitative modeling of physical systems (Bobrow, 1985) has made an important distinction between system function and system behavior. *System function* is represented in terms of intended system purpose, while *system behavior* is neutral with respect to design goals, being just what occurs. Diagnostic reasoning, as part of a troubleshooting process that attempts to restore a system's behavior to its specifications, is heavily function-oriented. A system's behavior can not be judged incorrect unless it is evaluated relative to the system's intended purpose. For example, a copier always generates behavior consistent with the current state of its components; it is judged to be performing improperly when the copies it produces (i.e., its intended purpose) are not adequate. We use the term *performance* to refer to a system's behavior as evaluated relative its function. An aspect of a system's performance can be correct, incorrect, or some degree of goodness, depending on aspect.

Through the composition of component behaviors, a diagnostic model must transform a representation of component behaviors into a representation of system function. This transformation is an explicit aspect of our diagnostic modeling scheme. It is a prerequisite for providing the second necessary property of diagnostic models, a high degree of functional adequacy. *Functional adequacy* is a measure of the extent to which a diagnostic model represents the interactions whereby a system's function becomes realized through the composition of its component behaviors.

As part of our group's research effort toward understanding the nature of diagnostic knowledge and reasoning, several protocols were collected of expert field personnel as they troubleshot faulted copiers. Two general features of these protocols have served as bases for two further principles that underlie our modeling approach. The first feature is the *normality principle* - that values attributed to component states and interfaces and that are propagated though elements of a diagnostic model are represented relative to normal levels. In our protocols, precise values tended to be discussed only when what would be normal must be evaluated in terms of observable system values. Such discussions reflect knowledge and problem solving in Observation Space, after reasoning in Diagnosis Space has determined the relevance of some possibly abnormal or normal value. The normality principle significantly influences our design of diagnostic models, as symptoms propagated within such models will be represented by descriptive variables that may assume values only at, above, or below normal levels.

The second principle underlying our approach to diagnostic modeling is the *single product principle*. The principle was exhibited in our protocols, as we found that most discussions revolved around how one incorrect output, such as a too light copy, could have been produced. The single product principle allows us to adopt a production line perspective toward system process. A *production line* is a process structure that processes a set of inputs through an acyclic succession of possibly concurrent stages to produce an eventual output. Not only does this perspective eliminate feedback interactions that can be very difficult to model and understand, it also eliminates continuous time as a direct concern of the model. A trace of the behavior of a production line from a particular input to an output can be represented as an alternating sequence of distinct product states and system operations. This trace is considerably easier to represent and interpret than a time-based account would be. It closely resembles the traditional, operator-based representation of plans used in problem space models of problem solving (Newell and Simon, 1973).

A Diagnostic Modeling Scheme

Now that we have discussed several general principles of diagnostic modeling, it is time to present our diagnostic modeling scheme. The five basic types of elements in our scheme are the PART, COMPONENT, STAGE, LINE, and PRODUCT. The PART and COMPONENT types are used to represent behaviors of the basic architectural elements of a system, including how their behaviors are affected by possible fault states. The STAGE, LINE, and PRODUCT types are used to interconnect these elements into a production line, thereby capturing basic functional aspects of the system. The PRODUCT type is used to represent entities operated on by a system; the STAGE and LINE types represent how PARTs and COMPONENTs are brought into contact with PRODUCTs to realize the system's function. It is important to note that, according to our scheme, the interconnections between COMPONENTs of a diagnostic model are distinctly functional in nature, while still reflecting the physical interconnections of a system's architectural structure.

A PART is used to represent a primitive element of a system's architecture, one not broken down into further sub-components. A PART's behavior is represented by a set of Function Variables, with associated Fault States and Causes, indicating causal relations between Fault States and abnormal Function Variable values. The Function Variables represent a PART's contribution to the behavior of its environment, maintaining values that can be propagated to other elements of a model when computing system behavior. The Fault States and Causes represent diagnostic knowledge about a PART. The Fault States indicate the known ways that a PART may fail. The Causes represent the effects such failures have upon values of a PART's Function Variables, causing one or more Function Variables to assume values that differ from normal.

The normality principle discussed above allows us to simplify the value space associated with a Function Variable. The value space consists of only three values -- UP, NORMAL, and DOWN. The value space of the Fault States associated with a PART or COMPONENT consists of a set of symbolic names representing known possible, incorrect operational states. Causes consists of a set of associations between elements of Faults States and abnormal values of Function Variables. As such, our models of PARTs and COMPONENTs satisfy the locality principle discussed previously. The effects of any Fault State are represented solely in terms of deviations from normal in the values of local Function Variables. Any effects upon eventual system performance must be realized by propagation (qualitative simulation) through other elements of a diagnostic model.

As an example, two PART type elements - a coronode and a bias shield - model a negative dicorotron component of a xerographic system. A negative dicorotron is used in several stages of the xerographic process, when a uniform field is needed to charge the photoreceptor or transfer the toner image to paper (Figure 1). A negative dicorotron consists of a coronode, being a wire in a glass tube that creates a surrounding field of ionized air, and a bias shield. In a negative dicorotron the bias shield is negatively charged, so that the dicorotron emits a uniform, negatively charged field. Figure 4 presents our diagnostic model of the coronode PART of a negative dicorotron. The coronode can fail to be correctly operating in one of several ways, as indicated by the elements of Fault States. Causes indicates, among other causal relations, that if a coronode assumes the Fault State of DIRTY, then the level of the charge emitted will be DOWN.

A COMPONENT corresponds to a physical element of a system that we choose to represent as a composite of subcomponents; a subcomponent could be either a PART or a COMPONENT. Figure 5 presents our model of a negative dicorotron, represented as a COMPONENT having a coronode and a negative bias shield (two PARTs) as its subcomponents. A COMPONENT's behavior is represented in terms of a set of Function

Variables, whose values are seen to be related to those associated with Function Variables of its subcomponents. These relations are of two types: P (propagate) and I (invert). These relations correspond directly to the M+ and M- functions discussed by Kuipers (Kuipers, 1985). An M+ relation between two function variables indicates that their values are directly related mathematically. For diagnostic purposes, P is the M+ relation, which represents the direct propagation of an abnormal or normal value. As an example, our model indicates that if *Ch* of a coronode is DOWN, then *Nf* of the dicorotron will also be DOWN. The I relation corresponds to the M-, indirect functional relationship; I inverts UP to DOWN and vice versa, given we assume other relevant variables are NORMAL, thus propagating the inverse abnormality. Faults at the COMPONENT level represent possible incorrect interactions among subcomponents. For example, if the bias shield and coronode are misaligned in a dicorotron, the evenness of the charge field emitted by the dicorotron will be DOWN.

Our model of the behavior of a negative dicorotron is already represented in functionally relevant terms according to our choice of Function Variables. However, it is independent of the dicorotron's actual use within the copying process and satisfies the locality principle. The Function Variables anticipate uses of a PART, but do not directly indicate its function in a given system. Nothing in our dicorotron model mentions a copy or other elements of the copying process. Due to our adherence to the locality principle, the dicorotron model still could be inserted anywhere in our model of the xerographic system.

The model of a PART or COMPONENT is incorporated into a system model by becoming a member of the Components aspect of a STAGE type element. A STAGE represents the bringing of a set of input PRODUCTs into contact with one or more PARTs or COMPONENTs to produce a set of output PRODUCTs. The STAGE and PRODUCT are the basic elements in our representation of system function in a diagnostic model.

PRODUCTs represent the things being acted upon by system components. These correspond to inputs, sub-assemblies, and eventual outputs of a system. A PRODUCT is represented in a manner analogous to a PART, having Function Variables, Fault States, and Cause aspects. This reflects the notion that a PRODUCT acts like a COMPONENT at a STAGE to the extent that a Fault State associated with an input PRODUCT may adversely affect the output PRODUCT of the STAGE. In addition to its COMPONENT-like properties, a PRODUCT has an associated set of Effect Variables, representing those aspects of the PRODUCT that are altered by operations of the system at one or more STAGEs. Figures 6 and 7 present our representation of the photoreceptor PRODUCT (PR) and the Charging STAGE of the xerographic system.

A STAGE is represented in terms of Inputs and Outputs, both being sets of PRODUCTs, and a set of PARTs and COMPONENTs that generate the STAGE's function. Using the P and I relations, a STAGE represents values of the Effect Variables of its Outputs as functions of values of the Effect and Function Variables of its Inputs and the Function Variables associated with its PARTs and COMPONENTs. For example, according to our model of the Charging STAGE, the level of charge held by a photoreceptor (PR) after charging is a direct function of the capacitance of the input PR and the level of the charge field emitted by the dicorotron. A fatigued photoreceptor results in a capacitance that is DOWN, illustrating our notion that a PRODUCT can act like a COMPONENT during the interaction occurring at a STAGE; a faulty input can adversely affect outputs even when system components are functioning correctly.

To complete a diagnostic model, we must compose the individual functions represented by the various STAGEs into the overall function of the system. This is accomplished through definition of a LINE, capturing the process of a complex system from the production line perspective. A production line can be modeled by a directed acyclic graph, leading from an INPUT, an external source of PRODUCTs, through various STAGEs, to an OUTPUT, a sink of final PRODUCTs. A LINE description associates the Inputs of one STAGE with the Outputs of other, prior STAGEs, while introducing the implicit, external INPUT and OUTPUT stages to complete the model. Figure 8 presents our LINE representation of the xerographic system of a copier. Each element of a LINE specification indicates the type and source of a STAGE's Inputs.

Diagnostic Reasoning

Given a diagnostic model, how can we use it to realize the goals of diagnostic problem solving? We remember these goals are to determine a set of possible faults consistent with observed abnormalities in system performance, to predict further symptoms implied by one or more of the possible faults, and to generate causal accounts which indicate how the local effects of possible faults could have propagated through the stages of a system's process to yield observed and predicted symptoms of the system's performance. To account for an observed abnormality in system output (i.e., an UP or DOWN value of an Effect Variable associated with an OUTPUT), we propagate values "backward" through STAGEs of the LINE according to the diagnostic reasoning rules of Figure 9. The propagation process terminates at a possible fault or when there is no further relation available. As the LINE is acyclic and each component is represented as a tree of subcomponents, the process will halt with a finite set of causal stories. Figure 10 presents a subset of the accounts for the observation of light copy (i.e., the amount of toner on the image at output is DOWN), as computed by our implementation of a diagnostic model for the xerographic subsystem. Figure 11 indicates ways that a charge dicorotron could be involved in producing this symptom according to our previously presented model of the negative dicorotron.

Several comments about this reasoning process and its results are in order. First, much information is lost due to the simplified value spaces and relations. This can lead to directly conflicting predictions during propagation. If the value of function variable v is equal to the sum of values of x and y when represented quantitatively, then we have P(v, x) and P(v, y) when this relation is represented diagnostically. If propagation has produced the possibilities that x is UP and y is DOWN, then we could predict the two contradictory possibilities that v is UP and v is DOWN. Fortunately, these predictions would be found in independent causal stories. We argue that this is not completely inappropriate, as it represents the one-track focusing of diagnostic reasoning often observed in expert troubleshooting protocols. Another consequence of information loss in a diagnostic model is the potentially large number of faults that can be generated to account for a given symptom. From a tutorial perspective this may not be a disadvantage, as a wide range of possible faults are described and possible interactions within the system are made explicit; but from a field application perspective, something must be done to reduce or order the possible faults.

One assumption that is often made during troubleshooting is that at most one fault is present in the system at any time. This *single fault assumption* is useful for reasonably reliable systems and allows us to better prune the set of possible faults when several symptoms have been observed. We merge causal stories accounting for a set of observed symptoms that all begin with a common fault and do not contain direct conflicts regarding Function Variable values. The resultant causal story is a tree starting at the fault and ending in leaves that are observed abnormalities. This contrasts with the set of stories for an individual symptom, which is a tree having leaves that are possible faults and the symptom as root. They represent the bi-directional nature of diagnostic reasoning made possible by our diagnostic modeling scheme.

Discussion

Diagnostic reasoning has long been a topic of research in artificial intelligence. Many expert systems have had as their goal the effective diagnosis of either diseases in biological systems or faults in electronic or mechanical systems. Until recently, most of these expert systems adopted an *experience-based approach* to the representation of diagnostic knowledge. In such systems, diagnostic associations between symptoms and possible faults are represented directly, reflecting "compilation" of an expert's experience as to the co-occurrence of a fault or disease and the observation of particular symptoms. Research based upon this approach has resulted in several effective strategies for diagnosis (Clancey,1984; Reggia, et.al., 1978; Gomez and Chandrasekaran 1981).

Our scheme is representative of a *model-based approach* to diagnostic reasoning, whereby the associations between symptoms and faults are represented only indirectly and must be derived by propagation of the local effects of faults through behaviors of system components to observable symptoms. Kuipers and Kassirer (1984) characterize the differences between the experience- and model-based approaches in medicine. Genesereth (1985) reports on a model-based approach to the diagnosis of faults in complex digital computer circuits. His models are referred to as design descriptions, indicating their high degree of architectural fidelity. Genesereth presents techniques for manipulating the clauses of a design description to determine possible faults or to generate symptom-related tests of interest. Genesereth's models propagate actual values rather than values taken relative to normal; however, the domain of digital circuits he considers limits the value space of descriptive variables to be {0, 1}.

Another feature discussed by Genesereth, also emphasized by Davis (1985), is the hierarchic nature of complex systems. Their hierarchical models differ from the hierarchy included in our COMPONENT models; theirs is a hierarchy over STAGEs. This suggests how our modeling scheme could be extended to realize a more general form of hierarchical mechanism model. We could introduce the notion of a STAGE-LINE, which has its functional aspect represented as a LINE rather than as a set of instantaneously acting COMPONENTs. The Inputs and Outputs of a STAGE-LINE correspond to those of the LINE implementing its function.

An interesting point arises in Davis' discussion of the determination of bridge faults in computer circuits, where a "potential connection" exists between two circuits due to their spatial proximity. A bridge fault can then create an actual, functional connection between the two, effectively altering the architecture of the device (i.e., a different LINE is formed). As currently discussed, the modeling scheme we propose could not determine such errors, since our model of a system's architecture is fixed. The diagnosis of non-functional faults is a difficult issue to address. We would argue that not being able to easily diagnose faults outside the functional architecture of a mechanism is not a serious shortcoming of our modeling scheme. Rather, our scheme is defined so as to take advantage of the simplifications observed in normal diagnostic reasoning. These simplifications are heuristic in nature, but allow most problems to be solved with reasonable, appropriately focused effort. Only when a functionally-oriented diagnostic model fails, would one want to resort to more naive physical or behavioral models. Typically, non-functional faults go undiagnosed or become part of an experience-based addendum to a model-based, diagnostic model. This is the case for bridge faults in circuits and for most commonly occurring sets of multiple faults.

Conclusion

In this paper we have presented a general scheme for the diagnostic modeling of complex mechanisms. Basic elements of the associated reasoning methods for single and multiple symptoms have been implemented as part of a prototype system. The causal narratives presented in Figures 10 and 11 represent an example of this system's output. Features of the modeling approach and reasoning algorithm are being incorporated into a system that will assist with the training of troubleshooting personnel. A diagnostic model of the xerographic system will be used to explain why certain tests are being performed, showing how expectations of symptom observations follow from propagation of local effects of likely faults through system components to observable values. Similarly, the model can indicate why a particular fault is suspected on the basis of previously observed symptoms. The instructional system is enhanced by graphical displays of the various stages of the xerographic process. Educational applications of qualitative mechanism models hold much promise for improving instruction about operational and diagnostic procedures associated with complex systems.

In closing, we reemphasize the maxim that models are created for a particular purpose and are limited in their capabilities according to that purpose. Diagnostic models reflect their purpose in the simplification of value spaces associated with model variables to values relative to normal and in the representation of model architectures as acyclic process lines.

References

Bobrow, D.G. ed. (1985), *Qualitative Reasoning about Physical Systems*, MIT Press: Cambridge, MA.

Clancy, W.J. (1984), "Classification problem solving", in *Proceedings of the National Conference of the American Association for Artificial Intelligence*; Austin, Texas; August, 1984; p49-55.

Davis, R. (1985), "Diagnostic reasoning based on structure and behavior", in Bobrow (1985), *Qualitative Reasoning about Physical Systems*, MIT Press: Cambridge, MA, p347-410.

DeKleer, J. and Brown, J.S., "A qualitative physics based on confluences", in Bobrow (1985), *Qualitative Reasoning about Physical Systems*, MIT Press: Cambridge, MA, p7-84.

Farley, A.M. (1985), "A general model of troubleshooting and its application to computer support", in *Proceedings of the Fifth International Workshop on Expert Systems and their Application*; Avignon, Fr., April, 1985, p489-503.

Genesereth, M. R. (1985), "The use of design descriptions in automated diagnosis", in Bobrow (1985), p411-436.

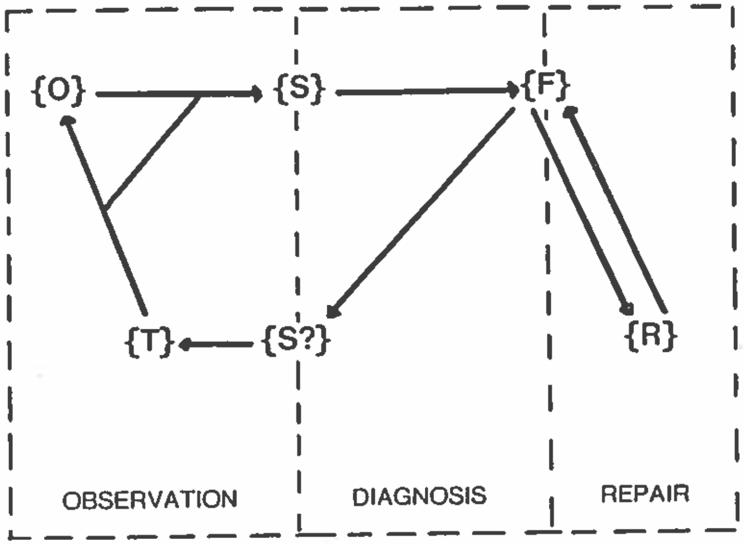
Gomez, F. and Chandrasekeran, B. (1981), "Knowledge organization and distribution for medical diagnosis", *I.E.E.E. Transactions on Systems, Man, and Cybernetics*, SMC-11(1), p34-42.

Kuipers, B. (1985), "The limits of qualitative simulation", *Proceedings of IJCAI-85*, Los Angeles, CA, August, 1985.

Kuipers, B. and Kassirer, J.P., (1984), "Causal reasoning in medicine: analysis of a protocol", *Cognitive Science*, 8, p363-385.

Newell, A. and Simon, H.A. (1973), *Human Problem Solving*, Prentice-Hall: Englewood Cliffs, NJ.

Reggia, J.A., Nau, D.S., and Wang, P.Y. (1978), "Diagnostic expert systems based on a set covering model", *International Journal of Man-Machine Studies*, 19, p437-460.



{O} observations

{F} faults

{S?} expected symptoms

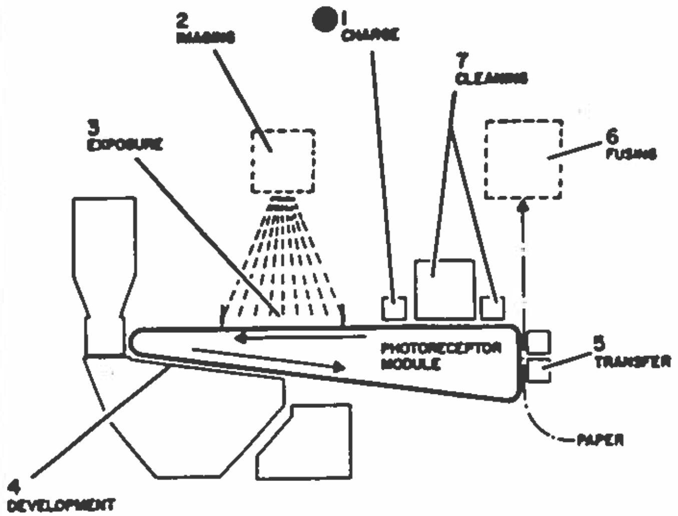
{S} symptoms

{T} test procedures

{R} repair procedures

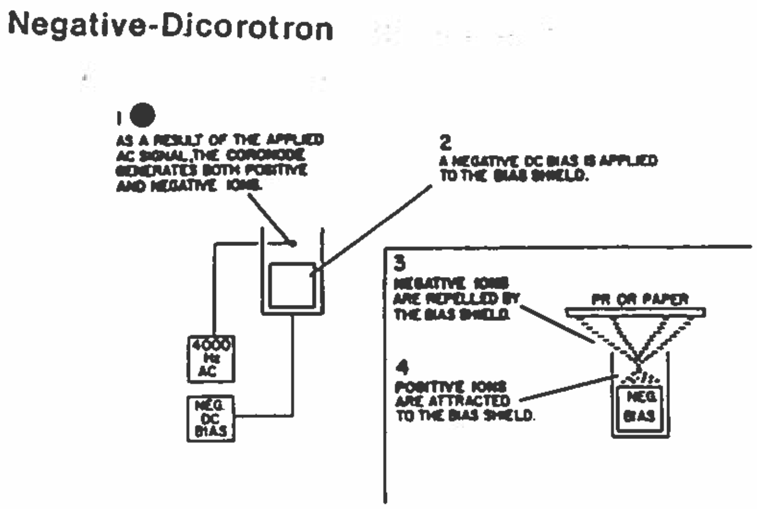
Figure 1. A General Model of Troubleshooting

Xerography



1. Charging: Charge Dicorotron
2. Imaging: Image Light, Lens, Shutter
3. Exposing: Edge Erasers, Interdocument Lamps
4. Developing: Toner Dispenser, Developer Housing
5. Transfer: Pre-transfer Light, Transfer Dicorotron
6. Fusing: Fusing Roller
7. Cleaning: Pre-clean Dicorotron, Cleaner Housing

Figure 2. The Xerographic Process



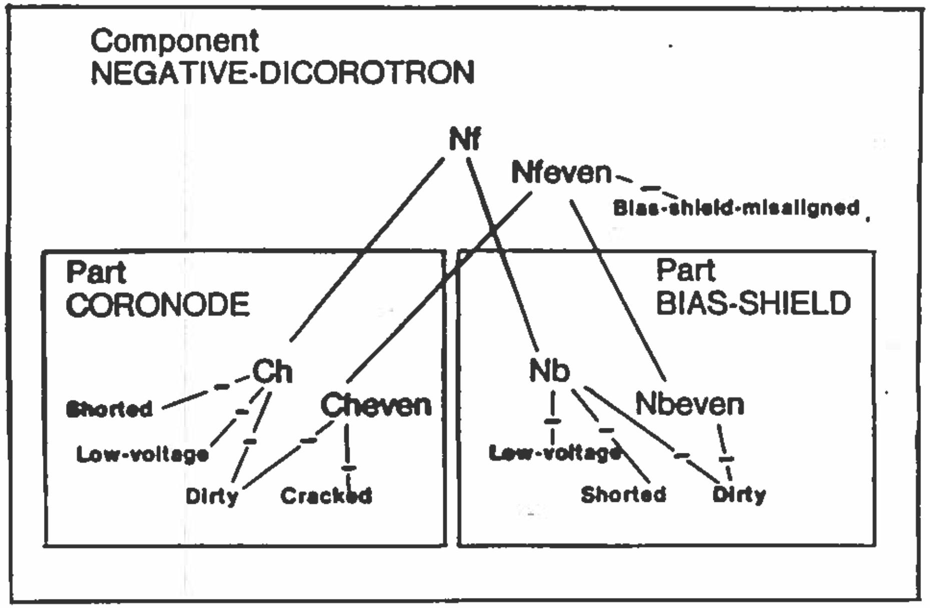
PART: CORONODE

Function Variables
Ch -- Level of charge from coronode
Cheven -- Evenness of charge from coronode

Fault States
Shorted -- Coronode wire shorted
Cracked -- Coronode glass cracked
Dirty -- Coronode glass dirty
Low-Voltage -- Insufficient voltage at coronode

Causes
(Ch DOWN) <== (Fault Low-Voltage) (Fault Shorted) (Fault Dirty)
(Cheven DOWN) <== (Fault Cracked) (Fault Dirty)

Figure 4. The Negative Dicorotron and
a Diagnostic Model of a Coronode



COMPONENT: NEGATIVE DICOROTRON

Subcomponents
(CORONODE NEGATIVE-BIAS-SHIELD)

Function Variables
Nf -- Level of negative field from dicorotron
Nfeven -- Evenness of field from dicorotron

Function
Nf <== (P Ch) (P Nb)
Nfeven <== (P Cheven) (P Nbeven)

Fault States
Misaligned -- Bias Shield not aligned correctly

Causes
(Nfeven DOWN) <== (Fault Misaligned)

Figure 5. A Diagnostic Model of the Negative Dicorotron

```
PRODUCT: PR (Photoreceptor)

Effect Variables
    V -- Level of negative voltage held by photoreceptor
    Veven -- Evenness of voltage held by photoreceptor
    To -- Amount of toner on photoreceptor

Function Variables
    Vc -- Voltage capacitance of photoreceptor
    Vceven -- Evenness of capacitance of photoreceptor

Fault States
    Fatigued -- Photoreceptor fatigued
    Belt-Grounded -- Photoreceptor belt grounded

Causes
    (Vc DOWN) <== (Fault Fatigued)
    (Vceven DOWN) <== (Fault Belt-Grounded)


STAGE: CHARGING

Inputs
    (PR)

Outputs
    (PR)

Components
    (Negative-Dicorotron)

Function
    V <== (P Nf) (P Vc) (P V)
    Veven <== (P Nfeven) (P Vceven) (P Veven)
    To <== (P To)
```

Figure 6. Diagnostic Models of the PR PRODUCT and the Charging STAGE

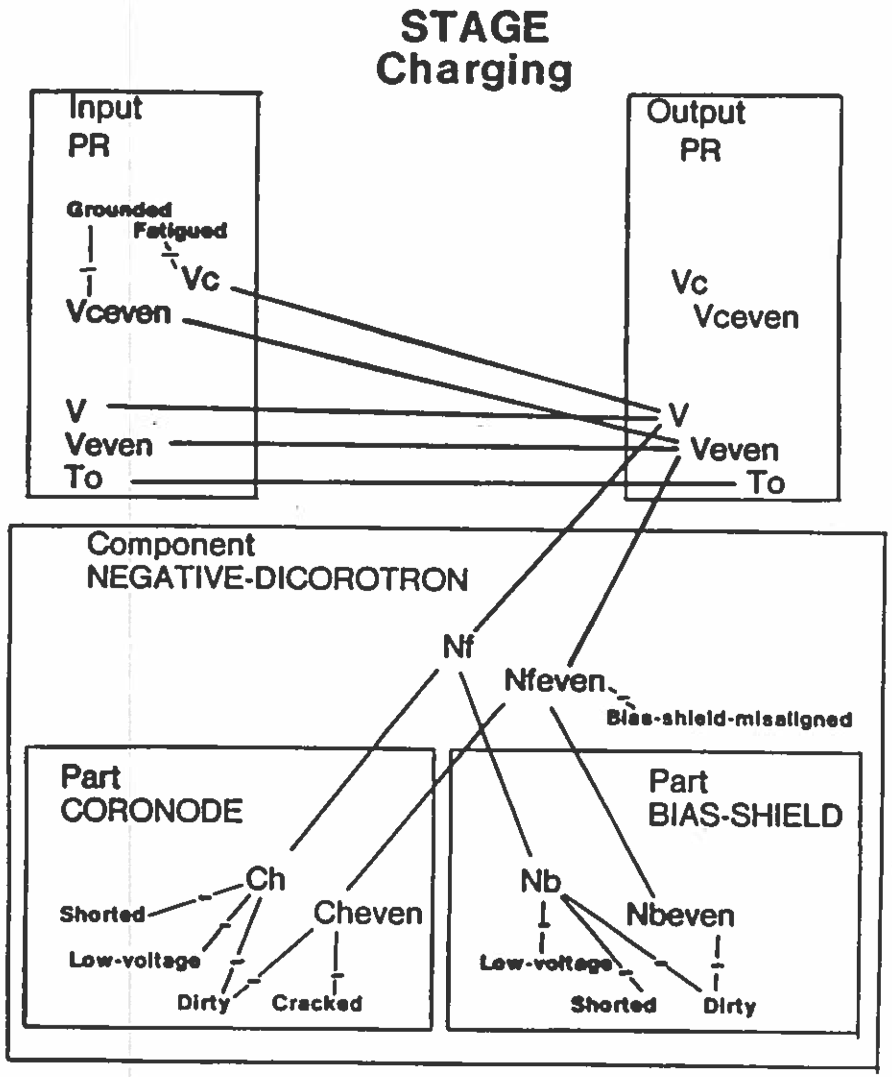


Figure 7. A Graphical Characterization of Charging

Xerography

Connections:

```
((((INPUT PR)) Cleaning)
  (((Cleaning PR)) Charging)
  (((Charging PR)) Exposing)
  (((Exposing Exposed-PR)) Developing)
  (((Developing Developed-PR) (INPUT PAPER)) Transfer)
  (((Transfer Developed-PAPER)) Fusing)
  (((Fusing Fused-PAPER)) OUTPUT))
```

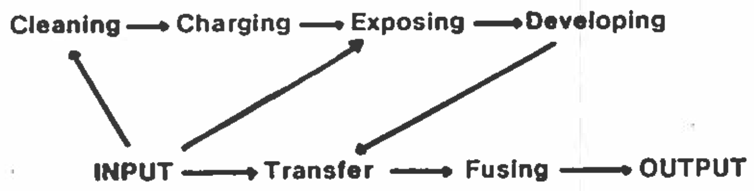


Figure 8. A Diagnostic Model of the Xerographic Process Viewed as a Process Line

R1: If have observed symptom or possible effect (x DOWN)
and the model states that x <== (P y)
then have possible effect (y DOWN).

R2: If have observed symptom or possible effect (x UP)
and the model states that x <== (P y)
then have possible effect (y UP).

R3: If have observed symptom or possible effect (x DOWN)
and the model states that x <== (I y)
then have possible effect (y UP).

R4: If have observed symptom or possible effect (x UP)
and the model states that x <== (I y)
then have possible effect (y DOWN).

R5: If have observed symptom or possible effect (x DOWN)
and the models states that (x DOWN) <== (Fault y)
then have possible fault y.

R6: If have observed symptom or possible effect (x UP)
and the models states that (x UP) <== (Fault y)
then have possible fault y.

Figure 9. Basic Diagnostic Reasoning Rules

```
The following represent causal stories for the effects:
((Toner-for-image-on-paper-at-Output DOWN))

(Negative-Voltage-on-PR-after-Charging DOWN)
  = = > (Negative-Voltage-for-image-on-PR-after-Exposure DOWN)
   = = > (Negative-Voltage-for-image-up-at-Development DOWN)
    = = > (Toner-for-image-on-PR-after-Development DOWN)
     = = > (Toner-for-image-on-paper-after-Transfer DOWN)
      = = > (Toner-for-image-on-paper-after-Fusing DOWN)
       = = > (Toner-for-image-on-paper-at-Output DOWN)


The following represent causal stories for the effects:
((Nogative-Voltage-on-PR-after-Charging DOWN))

(Negative-Voltage-on-PR-before-Charging DOWN)
   = = > (Negative-Voltage-on-PR-after-Charging DOWN)

(FAULT Photo-receptor-fatigued)
   = = > (Negative-Voltage-capacitance-of-PR DOWN)
    = = > (Negative-Voltage-on-PR-after-Charging DOWN)

(FAULT Voltage-low-at-negative-bias-shield-at-Charging)
   = = > (Negative-field-at-shield-at-Charging DOWN)
    = = > (Negative-field-from-dicorotron-at-Charging DOWN)
     = = > (Negative-Voltage-on-PR-after-Charging DOWN)

(FAULT Nogative-bias-shield-shorted-at-Charging)
   = = > (Negative-field-at-shield-at-Charging DOWN)
    = = > (Negative-field-from-dicorotron-at-Charging DOWN)
     = = > (Negative-Voltage-on-PR-after-Charging DOWN)

(FAULT Coronode-wire-shorted-at-Charging)
   = = > (Charge-from-coronode-at-Charging DOWN)
    = = > (Negative-field-from-dicorotron-at-Charging DOWN)
     = = > (Negative-Voltage-on-PR-after-Charging DOWN)

(FAULT Coronode-glass-dirty-at-Charging)
   = = > (Charge-from-coronode-at-Charging DOWN)
    = = > (Negative-field-from-dicorotron-at-Charging DOWN)
     = = > (Negative-Voltage-on-PR-after-Charging DOWN)

(FAULT Voltage-insufficient-to-coronode-at-Charging)
   = = > (Charge-from-coronode-at-Charging DOWN)
    = = > (Negative-field-from-dicorotron-at-Charging DOWN)
     = = > (Negative-Voltage-on-PR-after-Charging DOWN)
```

Figure 10. Causal stories for light copy that involve the negative dicorotron at charging