Proceedings of
The 1996 ASME Design Engineering Technical Conferences and
Computers in Engineering Conference
August 18-22, 1996, Irvine, California

96-DETC/CIE-1340

SYSTEM BEHAVIOR MODELING AS A BASIS FOR ADVANCED FAILURE MODES AND EFFECTS ANALYSIS

Charles F. Eubanks
Steven Kmenta
Kosuke Ishii
Design Division
Mechanical Engineering Dept.
Stanford University
Stanford, CA

ABSTRACT

This paper presents a method for developing a device behavior model to enhance reliability at the early stages of conceptual design. The model facilitates a semi-automated advanced failure modes and effects analysis (FMEA). The model performs analyses and simulations of device behavior, reasons about conditions that depart from desired behaviors, and analyzes the results of those departures. The proposed method rigorously specifies pre- and post-conditions, yet is flexible in the syntax of device operation. The paper shows how the method can capture failures normally missed by existing FMEA methods. An automatic ice maker serves as an example application.

1. INTRODUCTION

The past decade has seen a growing body of research on several concurrent engineering issues. Probably the most mature of these is design for assembly (DFA) (Boothroyd & Dewhurst, 1991; Homem de Mello & Sanderson, 1991; Sturges & Kilani, 1992). Design for producibility (Priest, 1988; Arimoto, et al., 1993) and design for manufacturability (Poli, et al., 1992; Fathailall & Dixon, 1994) have also received attention. These issues are very important, but fail to address costs incurred after the product leaves the factory.

Rising warranty costs have focused attention on the issue of design for serviceability (Makino, et al., 1989; Berzak, 1991; Eubanks & Ishii, 1993), particularly among automotive and major appliance manufacturers. Component manufacturing and assembly play a major role in overall quality and reliability. Similarly, system configuration and assembly methods affect the ease of service. More recently, design for environment (DFE: Navin-Chandra, 1991; Glantsching, 1993; Marks, et al., 1993) has become a factor. DFE focuses on how the materials and manufacturing processes impact the earth’s natural resources. All of these considerations point towards the need for an integrated approach to product design and manufacturing, and for a decision support systems to aid engineers early in the design process.

Product designers begin by defining customer requirements, performing a functional analysis to generate the design concepts, and then map the generated concepts to components and sub-systems that implement them (Suh, 1990; Ullman, 1992). This paper proposes a method of device behavior modeling and behavior to structure mapping that leads to a semi-automated advanced Failure Modes and Effects Analysis (FMEA) system. A method capable of performing FMEA in the early stages of design provides insight into issues surrounding reliability, diagnosis, and serviceability. Key elements required to develop this capability include:

- a behavior model suitable for use in the early stages of design
- a structural model suitable for use in the early stages of design
- a framework linking these two models
- inferencing methods for evaluating effects of both behaviors and misbehaviors

〈 a user intuitive interface consistent with the conceptual design process

The proposed method builds on preliminary work by Di Marco, et al. (1995), who showed that such an analysis could be extracted from a fairly simple function-to-structure mapping. This paper discusses some advantages of the method over standard functional analysis, and shows how the method captures a wider range of failure modes. An automatic ice maker serves as an illustrative example.

2. FAILURE MODES AND EFFECTS ANALYSIS

FMEA has been a standard design practice for many years, and is generally separated into process FMEA and design FMEA. Most of the common FMEA practices are based on MIL-STD-1629A (Department of Defense, 1980), but there exist many domain and industry specific methods. FMEA develops a list of failure modes ranked according to their effect on the user. This ranking provides a measure for deciding which components or subsystems need further testing and redesign. Major factors include component or sub-system failure rate, type of failure (fail, degrade, etc.), severity of failure, and likelihood of detection.

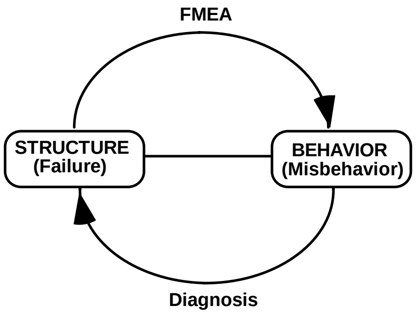

Figure 1. FMEA / Diagnosis Relationship

FMEA begins with a failed or degraded component (Figure 1), and attempts to identify the end-effect, usually expressed as a malfunction or misbehavior. Diagnosis entails the same notion, but occurs in the opposite direction, starting with an observed misbehavior, and attempting to identify the failed component.

Like serviceability, the major problem associated with traditional FMEA and diagnosability analysis is that these methods occur too late in the design process, since they rely on the specification of the components making up the device. Generally, the required component information is only available after completion of system design phase. Thus, any shortcomings with the design that might be identified by FMEA can be very expensive and difficult to correct.

Our industry collaborators also indicate that FMEA tends to be carried out on sub-systems without necessarily addressing system wide effects. For example, a critical design criteria for an automatic ice maker is its horizontal alignment. Appropriate alignment assures that the water level is even throughout the ice maker freezing tray, so that ice cubes freeze evenly. Problems develop when the ice maker is correctly aligned with respect to the freezer, but the freezer is not level with respect to the earth. A standard FMEA is likely to miss this failure mode, because it does not account for issues related to the device's interface with the rest of the system.

Several automated FMEA systems exist for use in analyzing electrical systems, since electrical faults and failures can be more easily characterized as numerical quantities. Ormsby, et al., (1991) developed a concept for automated FMEA employing qualitative reasoning in a model-based environment as a means of making the analysis extensible to other domains. Computer-based diagnosis systems have been a popular research subject for the past several years, as evidenced by Hamsher, et al. (1992) Abu-Hanna, et al., (1991) showed that functional design models can be used in model-based diagnosis systems. In the mechanical engineering domain, Umeda, et al., (1992) used functional representations for diagnosis and self-repair of a copy machine. Morjaria, et al., (1992) have developed diagnostic systems based on belief network technology, employing causal networks and probability theory reasoning from symptom to failure in large industrial systems. Clark and Paasch (1994) showed how function to structure mapping can be used in the early stages of design to assess diagnosability; i.e., measuring the ease of isolating the cause of a malfunction.

3. BEHAVIOR MODEL DEVELOPMENT

3.1 Basic concepts and definitions

We define *structure* as the physical topology of a device or system, including the components that make up the system, and the relationships between the components.

A standard definition of *function* is the goal of what must be done without specifying how it is specifically achieved. Functional analysis is probably the most widely accepted practice for defining designs in the conceptual phase. Engineers begin by defining the overall function the device is to perform, and decompose it into sub-functions that delineate the design problems to be solved. (Suh, 1990; Ullman, 1992)

The definition of *behavior* normally follows the notion that it is "how (an) expected result is attained" (Keuneke, 1991), or the "detailed description of internal physical action based on physical principles and phenomena." (Welch & Dixon, 1994) However, using these definitions, the distinction between function and behavior blurs very quickly during the process of functional analysis. (Finger and Rinderle, 1989) The general distinction between models for function and behavior is the latter's use of pre- and post-conditions; i.e., what conditions must be true in order for the behavior to take place, and what conditions exist given that the behavior has taken place.

Behavior modeling has attracted theoretical development in the artificial intelligence (AI) community. Many AI researchers use causal chains or networks that are derivable either from the functional description of a device

(Chandrasekaran, 1993), from its structure, (Kuipers, 1984), or from qualitative physics. (deKleer and Brown, 1984) Researchers have developed more rigorous definitions and methods for describing the behavior of devices from the aspect of causal process descriptions of devices (Iwasaki & Chandrasekaran, 1992) and causal ordering based on process models. (Iwasaki & Simon, 1994)

3.2 Definitions

We submit the following definitions:

Definition 1: *Variable*: a triple (<object>, <attribute>, <value>)
where:
<object> can be any physical or conceptual entity
<attribute> is a distinctive quality or characteristic of the object
<value> is a quantification of the object attribute

Definition 2: *State*: a set of quantified state variables

Definition 3: *Behavior*: a transition from one state to another; i.e., initial state ♦ behavior ♦ final state

3.3 Behavior Definition and State Space Partitioning

Design begins when one recognizes a need and decides to build a device to satisfy that need. At this point, the designer has little idea of the device structure. What he/she does know is some initial existing condition, or state, that the product should alter to create some final desired state. For example, I recognize a need for ice cubes to be present in the ice bucket of my household freezer; i.e.,

Initial state: no ice cubes in ice bucket
Desired state: ice cubes in ice bucket

Conceptualizing a universe S exhibiting all the possible states of any device to be designed, one has essentially partitioned the state space into two regions, each of which is a set of states. A set of states can be thought of as a state for which some of the variables remain unquantified. Hence, one set consists of all possible states which have the common state variable (ice bucket, ice cube level, empty), while the other set consists of all possible states which have the common state variable (ice bucket, ice cube level, not empty). Identifying these sets as S_1 and S_2, respectively, we have:

S1 = {(ice bucket, ice cube level, empty),
*(obj2, *, *), (obj3, *, *),..., (objn, *, *)}* (1)

S_2 = {(ice bucket, ice cube level, not empty),
(obj_2, *,*), (obj_3, *, *),..., (obj_n, *, *)} (2)

where (obj_i, ∗, ∗), i = 2, ..., n, represent as yet unknown objects whose attributes may take on any value.

The definition of the desired behavior, b_1, for the device will be "deposit ice cubes in bucket", which causes the state transition to take place. (Figure 2)

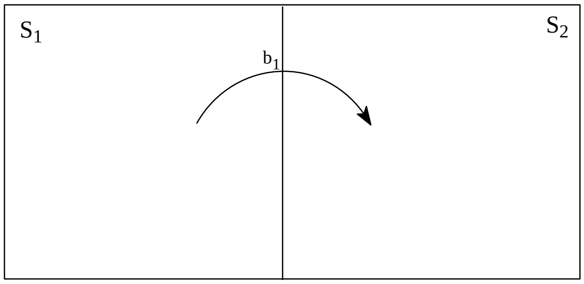

Figure 2. Partitioned State Space with Transition

The flow diagram below can also represent the state transition.

$$S_1 \xrightarrow{b_1} S_2 \qquad (3)$$

The designer may also know some general operating conditions that he/she either expect will exist, or that must exist due to the physical requirements of the process. In this case, the device must exist in an environment cold enough to freeze water, and maintain it in a frozen state. Once again, the state space can partitioned into sets of states which have state variables:

(environment, temperature, ≤ 32 F)
(environment, temperature, > 32 F)

Adding these variables results in the four sets of states:

S_1 = {(ice bucket, ice cube level, empty),
(environment, temperature, £ 32 F), (obj_3, *, *),
(obj_4, *, *),..., (obj_n, *, *)} (4)

S_2 = {(ice bucket, ice cube level, not empty),
(environment, temperature, £ 32 F), (obj_3, *, *),
(obj_4, *, *),..., (obj_n, *, *)} (5)

S_3 = {(ice bucket, ice cube level, empty),
(environment, temperature, > 32 F), (obj_3, *, *),
(obj_4, *, *),..., (obj_n, *, *)} (6)

S_4 = {(ice bucket, ice cube level, not empty),
(environment, temperature, > 32 F), (obj_3, *, *),
(obj_4, *, *),..., (obj_n, *, *)} (7)

Among the 4 definable sets of states, 2 are desirable, both of which have as a state variable the necessary operating condition (environment, temperature, ≤ 32 F). (Figure 3) S_3 is possible but undesirable, since there is no possible transition from this state to the desired condition of having ice cubes in the bucket. A steady state represented by S_4 is not possible under normal circumstances. Laws of nature further partition the state space, separating device states that are possible from those that are impossible; i.e., states that violate the 2nd Law of Thermodynamics, Newton’s Laws, etc. Thus a transition from an initial state in the set S_1 to any state in the set S_4 cannot occur, simply because the laws of nature prevent ice from existing for extended periods in an environment with temperature greater than 32 F.

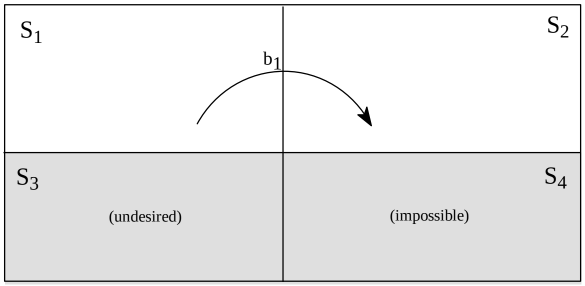

Figure 3. Partitioned State Space with Undesired and Impossible States Defined

In addition to desired, undesired, and impossible sets of states, sets may have partitions that are either “unknown,” or “not applicable.” Unknown sets are possible combinations of state variables which cannot be designated as either desired or undesired, and may represent behavior side-effects. Sets of states are “not applicable” when design decisions exclude them from the realm of possibility; e.g., deciding to use an electric motor for energy input as opposed to a hand crank.

3.4 Reasoning About Failures

Let’s assume that some device exists whose desired behavior is to deposit ice cubes into the ice bucket, and implicitly assume the existence of an operating environment with a temperature less than or equal to 32 F. Consider two possible failures: 1) the failure of the device to deposit ice cubes in the bucket, and 2) the failure of the environment to maintain a temperature less than or equal to 32 F. In failure 1, no state transition takes place. In failure 2, the device makes the transition into the undesired state, S_3. Hence, failures are generally either 1) the failure to transition to the desired end state, or 2) the transition to an undesired state.

A behavior (b_1) is uniquely defined by both it’s initial state, or pre-conditions (S_1), and its final state, or post-conditions (S_2) (Iwasaki & Chandrasekaran, 1992). One can define any transition into an undesired state uniquely, and make the claim that the transition from some desired state (S_1) into an undesired state (S_3) results in a unique and undesired behavior (b_1'). It is also possible to define another undesired behavior, $b_1 2$, that transitions from S_2 to S_3, also a result of failure 2. One can assign both b_1' and $b_1 2$ the label “freezer failure”. (Figure 4) On the other hand, failure 1 can be thought of as a non-behavior, $\downarrow$b, representing a case where there is simply no transition to another state. Let us assign $\downarrow$b the label “ice maker failure,” since the goal is to identify failures.

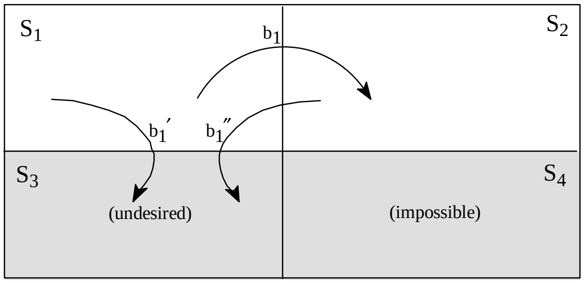

Figure 4. Representation of Undesired State Transitions

From the previous discussion, one can state the FMEA problem as:

> “Given an undesired behavior or a non-behavior, what is the resulting device state?”

Using the ice maker example, one could pose the FMEA question:

> “Given S_1 and “freezer failure”, how would that effect the system?”

From the transition graph, we see that the answer is S_3.

In the same way, one can state the diagnosis problem as:

> “Given an undesired final state, what desired behaviors did not occur, or what undesired behaviors did occur?”

Using the example, the diagnosis question would be:

> “Given that the device is in S_3, what happened?”

From our previous labeling, we know that the answer is :freezer failure.”

3.5 Behavior Decomposition

The device design proceeds with decisions about how the device is to perform its desired behavior. Each decision further partitions the state space, and delineates the sub-behaviors required to accomplish the overall device behavior. Note that the decomposition of a behavior into sub-behaviors is not necessarily unique. The details of the decomposition hierarchy will be a direct reflection of the design team’s implementation decisions. Therefore, any representation model must support multiple sets of decomposed child behaviors for each parent behavior.

Continuing with our example, behavior b_1 can be decomposed into 2 sub-behaviors:

b_{11}: create ice cubes
b_{12}: deposit ice cubes in bucket

Any pre-conditions required for b_1 must also be required for b_{11}, and that any post-conditions resulting from b_1 must also result from b_{12}, if this decomposition is valid. In state space terms, any evaluated variables common to partition S_1

must also be common to partition S_{11}, and likewise for S_2 and S_{22}

In this example, the post-conditions of b_{11} are precisely the pre-conditions of b_{12}, so we can say that b_{11} and b_{12} are causally linked to form the state transition graph:

$$S_{11} \xrightarrow{b_{11}} S_{12} \xrightarrow{b_{12}} S_{22} \qquad (8)$$

Figure 5 shows the expanded state space graphically.

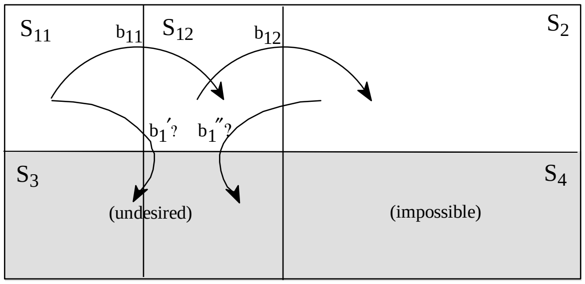

Figure 5. Representation of a Decomposed State Space

4. CONCEPT DEVELOPMENT ON DEVICE STRUCTURE REPRESENTATION

The previous section showed how to define failures and their effects in terms of undesired states and transitions to those states. However, the FMEA makes more sense to engineers when placed in the context of components and/or subassemblies. Designers need a comparable model for device structure, capable of capturing as much knowledge about the physical aspects of the device as possible, early in the design phase.

Ullman (1993) points out that in many cases some structural decisions are made in these early stages, indicating that the structural representation needs to be developed in parallel with the behavioral model. As the decomposition of device behaviors continues, behavior descriptions will approach a detailed enough level to 1) warrant the use of engineering equations to describe state transitions, or 2) map directly to a know artifact performing the desired behavior (Ullman, 1992; Suh, 1990).

We propose an object-oriented approach to a structural representation syntax that aids in the definition of the basic physical objects and relationships between these objects. (Figure 6)

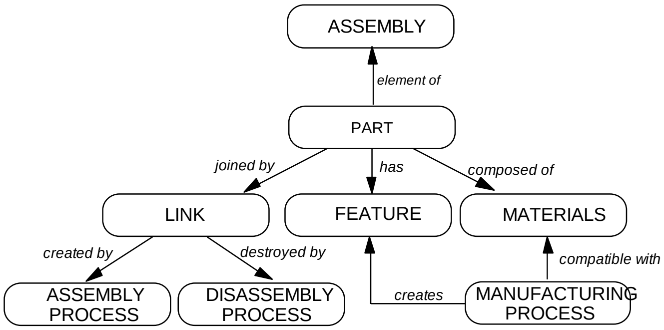

Figure 6. Representation of Existing Structural Representation

5. ICE MAKER EXAMPLE

5.1 Model Definition

We have developed a qualitative behavior model for an automatic ice maker typically found in household refrigerators. Table 1 lists the qualitative state variables that we have defined in order to describe the ice maker’s behaviors.

Table 1: Ice Maker State Variables

index	object	attribute	values
v1	ice bucket	ice level	empty, not full, full
v2	ice bucket	water level	none, some
v3	tray	water level	empty, full
v4	tray	water state	liquid, solid
v5	environment	temperature	<=15, >=15, >=32
v6	water valve	status	open, closed
v11	water switch	status	open, closed
v31	feeler arm switch	status	open, closed
v32	tray	temperature	<=15, >=15, >=32
v33	thermostat	status	open, closed
v34	heater	status	on, off
v35	motor	status	on, off
v36	cam	rotation	on, off
v37	ejector	rotation	on, off
v38	ice	interface state	solid, liquid

Figure 7 shows a three tiered model for the desired behaviors of the ice maker. The top level represents the overall desired behavior. The second tier (b1, b2, b3) shows three sub-behaviors that can be combined to give the overall behavior. Below this tier are the respective decompositions of the three second tier behaviors; i.e., b1 decomposes into b1.1 through b1.5, b2 decomposes into b2.1 and b2.2, and b3 decomposes into b3.1 through b3.10.

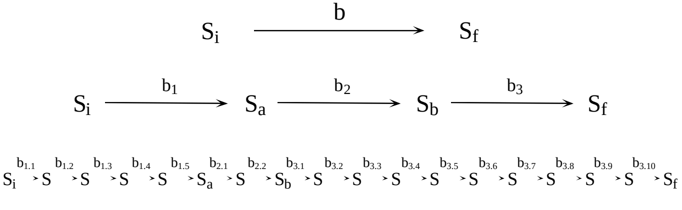

Figure 7. Behavior Representation and Decomposition

Table 2 identifies the state variables and behaviors associated with each tier. The required value defining the state variable appearing following the colon. An asterisk indicates the value has changed as a result of the behavior, but is no longer applicable for defining the state.

Table 2: Ice Maker Behavior Model

index	state 1	behavior	state 2
b	v1:empty, v2:none	icemaker fills ice bucket with ice	v1:full, v2:none
b1	v3:empty	tray fills with water	v3:full
b2	v3:full, v4:liquid	water freezes	v4:solid
b3	v4:solid	icemaker harvests ice	v3:empty, v4:*, v1:full
b1.1	v35:on, v33:open	water switch closes	v11:closed
b1.2	v11:closed	water valve opens	v6:open
b1.3	v6:open	tray fills with water	v3:full
b1.4	v35:on	water switch opens	v11:open
b1.5	v11:open	water valve closes	v6:closed
b2.1	v5:<=32	water freezes	v4:solid
b2.2	v5:<=15	tray gets cold	v32:<=15
b3.1	v32:<=15	thermostat closes	v33:closed
b3.2	v31:closed, v33:closed	heater circuit closed	v34:on
b3.3	v31:closed, v33:closed	motor circuit closed	v35:on
b3.4	v35:on	motor turns cam	v36:on
b3.5	v36:on	cam turns ejector	v37:on
b3.6	v34:on	heater warms tray	v32:>= 15
b3.7	v32:>= 15	thermostat opens	v33:open
b3.8	v34:on	heater warms tray	v32:>=32
b3.9	v3:full, v4:solid, v32:>=32	heater loosens ice	v38:liquid
b3.10	v4:solid, v37:on, v38:liquid	ejector pushes ice	v3:empty, v4:*

As stated in the section 3, one can identify failure effects in two ways: 1) by the results of non-behaviors, or 2) by examining known failure paths.

5.2 Non-behavior Failures

Suppose we have the behavior-to-structure link shown in Figure 8, and wish to simulate a thermostat failure. As a result of the failure, behavior b3.1 does not occur; i.e., the thermostat does not close. On this level, variable v33 retains the value "open," which means that the pre-conditions for behaviors b3.2 and b3.3 cannot be met. In this example, the failure continues to cascade, so that none of b3's sub-behaviors occur. The overall effect propagates up the behavior hierarchy such that b3 does not occur, and therefore b does not occur.

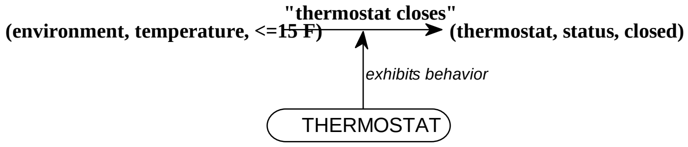

Figure 8. Ice Maker Behavior Model Fragment

For the question "What if the thermostat fails?", the answer can be inferred from the fact that the "thermostat closes" behavior has *not* occurred. Therefore, behavior "ice maker harvests ice" does not occur.

In addition, one can also infer a diagnosis path by determining the behavior that did not take place, based upon the deviation of state variable from their expected values. In the example, one expected the ice bucket level to be full, and instead it is not. The examination of the state of the device finds that the tray is full of frozen water, indicating that behavior b3 did not occur. Any components associated with accomplishing behavior b3 are now suspect. Traditional FMEA usually targets this class of failures (i.e., component failures) exclusively.

5.3 Defined Failures and Misbehaviors

A standard FMEA is likely to miss some failure modes, because it does not account for issues related to the device's interface with the rest of the system. Our industry collaborators indicate that standard FMEA tends to be carried out on sub-systems without necessarily addressing system wide effects. Consequently, variations in refrigerator alignment, water pressure, and freezer temperature, which contribute to nearly half of ice maker service calls, go unchecked.

In order to capture a wider class of failures, our method allows users to define failure modes explicitly. Section 2.4 describes problems resulting from ice maker misalignment. For proper operation, the ice maker should have a forward tilt of about 2 degrees. The refrigerator should be as close to level as possible. One can define state variables for the refrigerator and ice maker alignments as:

index	object	attribute	values
v7	refrigerator	alignment	nominal, >=2, <=-2
v8	ice maker	alignment	nominal, >=0, <=-4

One can define the cause-and-effect misbehaviors of these two variables as:

index	state 1	behavior	state 2
mb1	v7:>=2	refrigerator tilted back	v8:>=0
mb2	v7:<=-2	refrigerator tilted forward	v8:<=-4

Ice maker misalignment adversely affects ice cube quality. For example, a backward tilt causes the thermostat to close prematurely, resulting in hollow ice cubes. Either a backward or forward tilt causes ice cube shape to be non-uniform. With this information, one can define additional state variables and misbehaviors that identify the results of the ice maker misalignment. Engineers identified three major design attributes as:

index	object	attribute	values
v9	ice cube	shape	uniform, non-uniform
v10	ice cube	size	nominal, small, large
v11	ice cube	quality	nominal, hollow, brittle

To make the design attributes explicit, one can describe the overall desired behavior as:

index	state 1	behavior	state 2
b1	v1:empty, v2:none, v7:nominal	icemaker fills bucket with ice	v1:full, v2:none, v9:uniform, v10:nominal, v11:nominal

One can define the cause-and-effect misbehaviors for ice cube quality as:

index	state 1	behavior	state 2
mb3	v8:>=0	causes uneven water depth	v9:non-uniform
mb4	v8:>=0, v32:<=15	thermostat closes prematurely	v11:hollow, v33:closed
mb5	v8:<=-4	causes uneven water depth	v9:non-uniform

We can now run an FMEA for refrigerator misalignment by setting v7:>=2. The pre-condition for mb1 is now satisfied, so the system sets v8:>=0, causing mb3 and mb4 to occur, thus setting v9:non-uniform and v11:hollow. In words, the analysis shows that the refrigerator misalignment failure causes the ice maker to produce hollow ice cubes of non-uniform shape. An FMEA on the ice maker alone (i.e., separate from the refrigerator) would probably have missed this critical failure mode.

The above example demonstrates how using behavior modeling allows the designer to define factors affecting both the function of the system and the quality of the output. By making these factors explicit, one can investigate mis-behaviors and non-behaviors, and relate behaviors to system level parameters. For example, by evaluating "ice cube size," and listing all the factors contributing to this measure of quality, we find that this parameter is affected not only by fill time, but also by water pressure. At this stage the engineer can document the fact that water pressure is critical to ice cube quality, test for this condition, and make design changes early on, if necessary.

5.4 Summary of Standard vs. Advanced FMEA

To reiterate, we believe that our proposed method captures the effects of wider class of failures. Designers accomplish this by focusing not just on component failure within the system of interest, but also on failure that may occur *outside* the system that have direct bearing on system performance. Our industrial collaborators indicate that nearly 50% of reported ice maker failures are the result of failure in supporting systems, and not the ice maker system itself. To bring this into focus, Table 3 presents a partial list of common failures that appear as ice maker malfunctions, and whether the they would be addressed by a standard FMEA, and by our proposed advanced FMEA (AFMEA).

Table 3: FMEA / AFMEA Comparison

failure mode	FMEA	AFMEA
thermostat failure	yes	yes
water switch failure	yes	yes
feeler arm bent	yes	yes
gears iced up	yes	yes
power cord disconnected	yes	yes
refrigerator misalignment	no	yes
water pressure low	no	yes
freezer temperature high	no	yes

6. CONCLUSIONS AND FUTURE WORK

This paper presented a method for developing a device behavior model to enhance reliability at the early stages of conceptual design. The model expands existing AI behavior modeling techniques into the area of mechanical system design, and facilitates a semi-automated advanced failure modes and effects analysis (FMEA). Using an ice maker as an example, we demonstrated how the model can be used to perform analyses and simulations of device behavior, reasons about conditions that depart from desired behaviors, and analyzes the results of those departures. The paper showed how the method can also capture failures normally missed by existing FMEA methods by defining cause-and-effect relationships between system-wide design variables and sub-system quality measures. We believe that this approach allows designers to more rigorously define operating parameters, and aids in revealing parameter interaction. The result is a method capable of capturing a wider range of system failure mode effects.

We continue to refine our proposed method and the models that support it. Future efforts include:

〈 develop algorithms and a user interface that implement the proposed method

〈 validate the proposed method in a design evaluation setting

〈 extend the proposed method to diagnosability analysis

ACKNOWLEDGMENTS

The authors would like to thank General Electric, NASA Lewis Research Center, and the National Science Foundation for sponsoring this research. Special thanks to Dr. Mahesh Morjaria for his valuable assistance in developing the set-based behavior model.

REFERENCES

Abu-Hanna, A., Benjamins, R., Jansweijer, W. (1991), "Device Understanding and Modeling for Diagnosis," IEEE Expert, Vol 6, No. 2, pp. 26-32.

Arimoto, S., T. Ohashi, M. Ikeda and S. Mitakawa (1993), “Development of Machining-Producibility Evaluation Method (MEM),” Annals of the CIRP, vol. 42/1/1993, pp. 119-122.

Berzak, N. (1991), “Serviceability by Design,” Proc. of the 23rd Int’l SAMPE Technical Conference, October 21-24, 1991, pp. 1060-1071.

Boothroyd, G. and P. Dewhurst (1991), Product Design for Assembly. Boothroyd Dewhurst, Inc., Wakerfield, RI.

Clark, G. and R. Paasch (1994), "Diagnostic Modeling and Diagnosability Evaluation of Mechanical Systems," Proc. of the ASME 6th International Conference on Design Theory and Methodology (DTM '94), Minneapolis, MN, pp. 179-189.

Chandrasekaran, B., A. Goel, and Y. Iwasaki (1993), “Functional Representation as a Basis for Design Rationale,” IEEE Computer, 26(1):48-56, January, 1993.

deKleer, J. and J. Brown (1984), “A Qualitative Physics Based on Confluences,” Artificial Intelligence, 24(3), pp. 7-83.

Department of Defense (1980), "MIL-STD-1629A: Procedures for Performing a Failure Mode, Effects and Criticality Analysis," November 24, 1980, Washington, DC.

Di Marco, P., C. Eubanks and K. Ishii (1995), “Service Modes and Effect Analysis: Integration of Failure Analysis and Serviceability Design,” Proc. of the 15th Annual International Computers in Engineering Conference (CIE ‘95), Boston, MA, pp. 833-840.

Eubanks, C. and K. Ishii (1993), “AI Methods for Life-cycle Serviceability Design of Mechanical Systems,” Artificial Intelligence in Engineering Journal, 8(2):127-140.

Fathailall, A. and J. Dixon (1994), “A Method for Early Manufacturability Evaluation of Proposed Tolerance Plans for Thin-Walled Parts,” Proc. of the ASME 6th Int’l Conf. on Design Theory and Methodology, Minneapolis, MN, pp. 9-17.

Finger, S. and J. Rinderle (1989), “A Transformational Approach to Mechanical Design Using a Bond Graph Grammar,” Proc. of the ASME 1st Int’l Conf. on Design Theory and Methodology, Montreal, pp. 107-116.

Glantschnig, W. (1993), “Green Design: A Review of Issues and Challenges,” Proc. of the 1993 IEEE Int’l Symposium on Electronics and the Environment, Arlington, VA, pp. 74-77.

Goel, A. and B. Chandrasekaran (1989), “Functional Representation of Designs and Redesign Problem Solving,” Proc. of the 11th Int’l Joint Conf. on Artificial Intelligence, Detroit, MI, pp. 1388-1394.

Hamscher, W., Console, L., de Kleer, J. (eds.) (1992), Readings in Model-based Diagnosis, Morgan Kaufmann Publishers, San Mateo, CA.

Homem de Mello, L. and A. Sanderson (1991), “A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences,” IEEE Trans. on Robotics & Automation, vol. 7, no. 2, pp. 228-240, April, 1991.

Iwasaki, Y. and B. Chandrasekaran (1992), “Design Verification Through Function- and Behavior-Oriented Representations,” Artificial Intelligence in Engineering Design ‘92 (J. Gero, ed.), Kluwer Academic Publishers, Boston, pp. 597-616.

Iwasaki, Y. and H. Simon (1994), “Causality and Model Extraction,” Artificial Intelligence, (67)?Kluwer Academic Publishers, Boston, pp. 597-616.

Keuneke, A. (1991), “Device Representation: The Significance of Functional Knowledge,” IEEE Expert, vol. 6, no. 2, April, 1991, pp. 22-25.

Kuipers, B. (1984), “Commonsense Reasoning About Causality: Deriving Behavior from Structure,” Artificial Intelligence, 24(3), pp. 169-203.

Makino, A., P. Barkan, L. Reynolds and E. Pfaff (1989), “A Design for Serviceability Expert System,” Proc. of the ASME Winter Annual Meeting 1989, San Francisco, CA, pp. 213-218.

Marks, M., C. Eubanks, M. Shriver and K. Ishii (1993), “Life-cycle Design for Recyclability,” Proc. of the JSME-ASME Joint Workshop on Design ‘93, Tokyo, Japan, pp. 19-24.

Morjaria, M., M. Simmons, J. Stillman (1992), "Development of Belief Networks Technology for Diagnostic Systems," General Electric Corporate Research and Development Technical Information Series, GE CR&D, Schnectady, NY, April, 1992, 35 pages.

Navin-Chandra, D. (1991), “Design for Environmentability,” Proc. of the ASME 3rd Int’l Conf. on Design Theory and Methodology, Miami, FL, pp. 119-125.

Ormsby, A., J. Hunt, M. Lee (1991), Towards an Automated FMEA Assistant, Applications of Artificial Intelligence in Engineering VI, eds. Rzevski, G. and Adey, R., Computational Mechanics Publications, Southampton, Boston, pp. 739-752.

Poli, C., P. Dastidar and R. Graves (1992), “Design Knowledge Acquisition for DFM Methodologies,” Research in Engineering Design, (1992) 4:131-145.

Priest, J. (1988), Engineering Design for Producibility and Reliability, Marcel Dekker, Inc., New York, 1988

Sturges, R. and M. Kilani (1992), “Towards an Integrated Design for an Assembly Evaluation and Reasoning System,” Computer-Aided Design, vol. 24, no. 2, pp. 67-79, February, 1992.

Suh, N. (1990), The Principles of Design, Oxford University Press, New York, 1990.

Welch, R. and J. Dixon (1994), “Guiding Conceptual Design Through Behavioral Reasoning,” Research in Engineering Design, (1994) 6: 169-188.

Ullman, D. (1992), The Mechanical Design Process, McGraw-Hill, Inc., New York.

Ullman, D. (1993), "A New View on Functional Modeling," Proc. of the 9th International Conference on Engineering Design (ICED'93), The Hague, pp. 177-193.

Umeda, Y., T. Tomiyama, Y. Yoshikawa (1992), "A Design Methodology for a Self-Maintenance Machine Based on Functional Redundancy," Proc. of the 4th Int'l . Conference on Design Theory and Methodology, Scottsdale, AZ, pp. 317-324.