Proceedings of DETC'97
1997 ASME Design Engineering Technical Conferences and
Design Theory and Methodology Conference
September 14-17, 1997, Sacramento, California

97-DETC/DTM-02

ADVANCED FAILURE MODES AND EFFECTS ANALYSIS USING BEHAVIOR MODELING

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

ABSTRACT

This paper presents a systematic method applicable at the early stages of design to enhance life-cycle quality of ownership: Advanced Failure Modes and Effect Analysis (AFMEA). The proposed method uses behavior modeling to simulate device operations and helps identify failure and customer dissatisfaction modes beyond component failures. The behavior model reasons about conditions that cause departures from normal operation and provides a framework for analyzing the consequences of failures. The paper shows how Advanced FMEA applies readily to the early stages of design and captures failure modes normally missed by conventional FMEA. The result is a systematic method capable of capturing a wider range of failure modes and effects early in the design cycle. An automatic ice maker from a domestic refrigerator serves as an illustrative example.
KEYWORDS: behavior modeling, FMEA, reliability

1 INTRODUCTION

1.1 Motivation

Rising demand for quality during customer ownership has provided the need for design for reliability and serviceability (Makino, et al., 1989; Berzak, 1991; Eubanks & Ishii, 1993). Probabilistic methods for reliability assessment have been a mainstay of engineering systems development for many years (Levinson, 1964). Reliability methods focus on predicting availability and maintainability of complex systems based on mean time to failure (MTBF) and mean time to repair (MTTR) estimations for functional components or subsystems. Failure rate and distribution data are obtained through component life testing or generic failure rate distribution tables; information generally not available until the detailed design stage. These reliability methods are useful, however, product development teams need to design in quality and reliability at the beginning of conceptual design, before specific components are chosen.

This paper proposes a method to address reliability during the early stages of design, influencing both future design decisions and quality during customer ownership. Advanced FMEA uses behavior modeling to link desired behaviors with the components, operating environment, and supporting systems. Qualitative behavior simulation provides the framework for generating failure modes and their effects. Key elements required to develop this capability include:

- a behavior model suitable for the early stages of design
- a structural model suitable for the early stages of design
- a framework linking these models
- inferencing methods for evaluating effects of failures

The proposed method builds on preliminary work by DiMarco, et al. (1995), showing an FMEA analysis could be extracted from a fairly simple function-to-structure mapping, and extends the development of behavior modeling applied to FMEA by Eubanks (1996) and Eubanks et al. (1996). This paper discusses some advantages of behavior modeling over traditional functional analysis, and shows how Advanced FMEA can capture a wider range of failure modes compared to component-based FMEA. The paper will discuss the motivation for this new approach, an overview of behavior modeling, and the implementation of AFMEA on an automatic ice maker as an illustrative example.

1.2 Failure Modes and Effects Analysis

Design FMEA is a powerful preventative design method generally based on MIL-STD-1629A (Department of Defense, 1980). FMEA helps increase reliability and safety by identifying sources of failures and prioritizing design solutions or appropriate testing. A common approach to FMEA is to analyze failed or degraded components and identify causes and effects. Unfortunately, detailed information on the constituent components is available only after completion of layout design. At this late stage, causes of failures identified by FMEA can be very expensive or impossible to correct.

1.2.1 Shortcomings of Traditional FMEA

While traditional component-based FMEA is effective for identifying failure modes related to components, our industry collaborators have reported difficulty in identifying system-wide failure modes, such as problems with dependent systems, the operating environment, and customer usage. McKinney (1991) lists some major deficiencies with FMEA: narrowness of scope, lack of pertinence to operation and support of the system, timeliness, and the "box-checking" nature of the application. According to Kara-Zaitri et al. (1991) a major problem with FMEA is treating it as a perfunctory "checklist" only to satisfy contractual agreements with customers. Stamatis (1996) emphasizes the need to apply FMEA at an early, system level in order to effectively impact the design and reliability of the device. Traditional FMEA could benefit from a systematic approach capable of capturing a wider range of failure modes, applicable early in the design cycle.

1.2.2 Structured Approaches to FMEA

Several automated FMEA systems have been used to analyze electrical systems, since electrical faults and failures lend themselves to simple characterization 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. Montgomery et al. (1996) proposed a computer simulation of failure modes and their effects for electrical circuits, including qualitative simulation at the early stages.

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 using belief 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 by measuring the ease of isolating the cause of a malfunction. Palumbo (1994) uses mode variables and behavioral logic to automate FMEA of an actuator control system.

1.3 Our Approach

This paper describes a new approach for performing advanced FMEA using behavior modeling of mechanical and electro-mechanical devices. Guided by the function-structure relationship, one can build a behavior model describing the state changes of design variables expected during normal device operation. The model qualitatively simulates normal operation and analyzes the effects of failures in terms of the resulting state of the system.

Section 2 establishes the theory behind behavior modeling by defining behaviors, describing the device simulation, and classifying failures. Section 3 describes the application of behavior modeling to AFMEA. Section 4 uses an automatic ice maker to illustrate how the behavior model captures failure modes not identified using traditional approaches. The paper concludes with specific future activities to develop, formalize, and validate AFMEA.

2 BEHAVIOR MODELING AND FAILURE SIMULATION

2.1 Basic Concepts

Behavior modeling has received considerable attention in the Artificial Intelligence (AI) and mechanical engineering communities. Behavior knowledge forms the link between the functions, structure and state of a device through a behavior-structure modeling construct (Umeda, et al., 1990). Many AI researchers use the concept of causal chains or networks that are derivable either from the functional description of a device (Chandrasekaran et al., 1993), from its structure (Kuipers, 1984), or from qualitative physics (deKleer and Brown, 1984). Our work builds primarily on Chandrasekaran's functional representation, Keuneke's device modeling (1991), and Iwasaki and Simon's behavior modeling and simulation (1994).

Conventional design texts use functional decomposition as an early conceptual design tool. The design team begins by describing the overall function of the device and progressively decomposing the required functions in order to manage and understand the design (Suh, 1990: Ullman, 1992). A early function block diagram for an ice maker is shown in Figure 1.

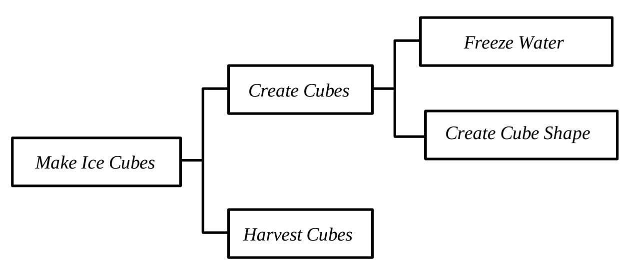

Figure 1: Function Block Diagram for Ice Maker

We use the behavior model in a manner similar to a functional model; as a tool to hierarchically decompose and define the design. However, the behavior model provides more information since each behavior is mapped to a specified state transition. Suppose one recognizes a need for ice cubes in a ice bucket of a household freezer, and defines a behavior called “deposit ice cubes in bucket.” In addition to the behavior, one can define the initial and final states for this behavior. The initial state is a description of a condition one wishes to change, for example:

Initial state: no ice cubes in ice bucket

The condition that results from the behavior of the device is the final state:

Desired final state: ice cubes in ice bucket

Keuneke (1989) uses the (<variable>, <value>) construct for representing state variables. We adapted this construct by dividing the definition of a variable into an object with one or more attributes, forming the (<object>, <attribute>, <value>) triple 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

This construct allows the designer to define easily various intensive and extensive properties of materials, components and systems. We note that a complete description of a device is represented by the set of all state variables. However, dealing with complete sets of state variables can be cumbersome. Chandrasekaran et al. (1993) define a partial state as the values of a relevant subset of the state variables. We express the partial states S_1 and S_2 as:

S_1 = {(ice bucket, ice cube level, empty)} (1)

S_2 = {(ice bucket, ice cube level, not empty)} (2)

The initial and final states are quasi-static, or quasi-steady states. The states can be identified as existing at some instance and may serve to trigger a behavior, for example:

(switch, position, closed)
(coil, status, energized)
(cam, position, 15°)

The designer can describe a behavior several ways, depending on the level of abstraction:

a) verbally: cause a greater amount of water to flow
b) qualitatively: the flow rate increases
c) numerically: the flow rate increases to 0.033 m^3/s
d) mathematically: $Q = vA$ m^3/s

The design proceeds with decisions about how the device is to achieve its desired behavior. For example, we can decompose the behavior “deposit ice cubes in bucket” into two sub-behaviors “b_{11}: create ice cubes” and “b_{12}: deposit ice cubes in bucket,” creating state transitions:

$$S_1 \xrightarrow{b_{11}} S_{12} \qquad (3)$$

$$S_{12} \xrightarrow{b_{12}} S_2 \qquad (4)$$

Behavior modeling provides a more robust basis for performing early design analysis for several reasons. First, behaviors do not rely entirely on the physical structure of the device. Although the physical elements or components may change as the design develops, general behaviors can be defined quite early. Therefore the AFMEA is applicable regardless of the component mix and maturity of the design.

Second, behavior modeling can reflect the customer’s desired requirements and attributes and provide a basis for assessing “customer dissatisfaction modes.” Finally, it provides a systematic framework for generating failure modes with an increased scope of analysis. The behavior model analyzes the design in the context of its supporting systems to capture a more comprehensive set of failure modes.

The specification of pre- and post-conditions builds behavior paths through the model allowing us to determine the behavior interactions and the failure propagation, i.e., pathways for the inferencing strategies necessary to perform AFMEA. A mapping between the functions and the structure forms a link between the descriptions of the device operation and the physical entities implementing those actions (Figure 2).

Table 1: Behavior Specification in Spreadsheet Form

Behavior				Pre-condition Specification			Post-condition Specification		
index	behavior	type	mapped to	object	attribute	value	object	attribute	value
1	deposit ice cubes in bucket	desired	ice maker	ice bucket	cube level	not full	ice bucket	cube level	full
			freezer	freezer	temperature	>8 & <15°F	freezer	temperature	>8 & <15°F

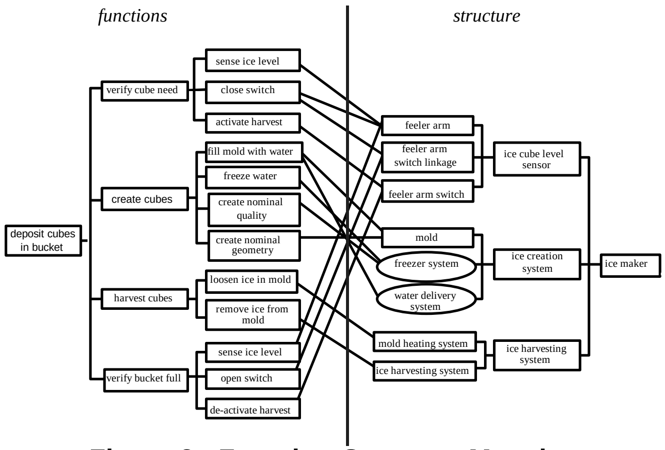

Figure 2: Function-Structure Mapping

We can visualize a portion of the behavior model at any hierarchical level by constructing a behavior fragment. The behavior fragment lists: the relevant initial state variables (pre-conditions), an arrow representing the behavior, and the relevant desired final state variables (post-conditions) and structural elements responsible for executing the behavior are listed below the behavior. Figure 3 shows an example of a behavior fragment of the high-level behavior "deposit ice cubes in bucket." Note how the environment, in this case the freezer temperature, is accounted for in the pre- and post-conditions.

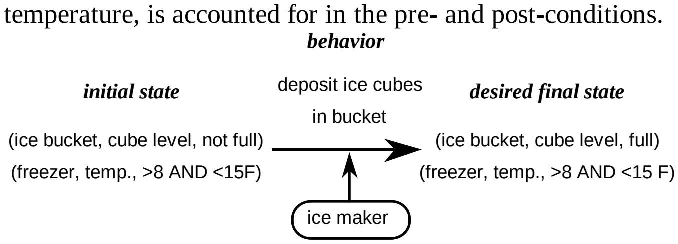

Figure 3: Behavior-Structure Fragment

The modeling information can be placed in spreadsheet form as shown in Table 1.

2.2 Behavior Simulation

Behavior simulation methods are established in the areas of diagnosis (Hamscher et al., 1992), design verification (Iwasaki & Chandrasekaran, 1992), and redesign (Goel & Chandrasekaran, 1989). The following algorithm draws on the method described by Iwasaki and Low (1991).

Step 1: Determine the initial values for all state variables based on the model definition and the user inputs, and list them as the current state.

Step 2: Compile a list of candidate behaviors having current state variable values as initial conditions, allowing decomposition of behaviors to subsume parent behaviors.

Step 3: Execute the candidate behaviors by determining the values of the state variables after the behavior's occurrence.

Step 4: Change the values in the list of current state variables list to reflect the results of Step 3. This list represents the current partial state of the system.

Step 5: Repeat Steps 2-4 until the list of candidate behaviors from Step 2 is empty.

The behavior model considers failure modes as the inability to transition from an initial or intermediate state to the desired final state, or, as the transition to an undesired state. Consider two possible failures: 1) after successful creation of the cubes, the cubes are not deposited in the bucket; and 2) during operation, the freezer temperature rises above 32°F and remains at this level.

Failure 1 is a "non-behavior" failure since the device remains in an expected intermediate state. Failure 2 is a "failure behavior" which takes the system into an undesired state to remain there until the failure is corrected. In this case, the conditions for maintaining freezer temperature have been violated. We can map the failure cause to associated variables and systems, for example: supply power, cooling system, thermostat setting, freezer seal condition, external temperature, etc. In this way we can generate causes for the failure mode by examining the requisite variables for the relevant behavior.

In addition, unwanted side effects, or misbehaviors can be defined as deviations from the voice of the customer, what we call the "negative VOC" ($\neg$VOC). For example, the customer prefers uniform sized cubes, translating into a desired post-condition: (cube, size, equal). A side effect resulting in a "customer dissatisfaction mode" would be any behavior resulting in a final state other than (cube, size, equal). The same approach can identify violations in other performance requirements such as noise, external heat gain, high power consumption, etc. These represent side-effects, largely unanticipated during design, that adversely impact the user's perception of the device even when the device still performs its basic functions.

Thus, we define failure as a condition where the *achieved* final state of a behavior does not match the *desired* final state. Similarly, we define misbehaviors as side-effects that result in an annoyance or shortcoming in the eyes of the user, even though the device successfully achieves the desired final state. Section 3 elaborates on types of failures identified by the behavior model and Section 4 gives examples of each type of failure.

3 APPLYING BEHAVIOR MODELING TO AFMEA

This section outlines the procedure for building the behavior model, simulating device operation, and identifying failure modes. The general procedure for Advanced FMEA is shown in Figure 4.

Table 2: Ice Maker: Ice Creation Behavior Model

Behavior				Pre-condition Specification			Post-condition Specification		
index	**behavior**	**type**	**mapped to**	**object**	**attribute**	**value**	**object**	**attribute**	**value**
1	deposit cubes in bucket	desired	ice maker	ice bucket	cube level	not full	ice bucket	cube level	full
			freezer	freezer	temperature	>8 & <15°F	freezer	temperature	>8 & <15°F
1.1	verify cube need	desired	cube level sensor	ice maker	harvesting status	inactive	ice maker	harvesting	active
				ice bucket	cube level	not full			
1.2	create cubes	desired	mold	mold	ice cubes present	no	mold	ice present	yes
1.3	harvest cubes	desired	mold	mold	ice cubes present	yes	mold	ice present	no
			ice bucket	ice bucket	cube level	not full	ice bucket	cube level	full
			ice maker	ice maker	harvesting status	active			
1.4	de-activate harvest	desired	ice maker	ice maker	harvesting status	active	ice maker	harvesting	inactive
			ice bucket	ice bucket	cube level	full			

Table 3: Decomposition of Behavior: Create Cubes

Behavior				Pre-condition Specification			Post-condition Specification		
index	**behavior**	**type**	**mapped to**	**object**	**attribute**	**value**	**object**	**attribute**	**value**
1.2	create cubes	desired	ice creation system	mold	ice cubes present	no	mold	ice cubes present	yes
1.2.1	fill mold with water	desired	water delivery system	mold	water level	none	mold	water level	full
			mold	mold	ice cubes present	no			
1.2.2	freeze water	desired	freezer system	water	state	liquid	water	state	solid
			mold	freezer	temperature	<32°F	mold	ice cubes present	yes
				mold	water level	full			

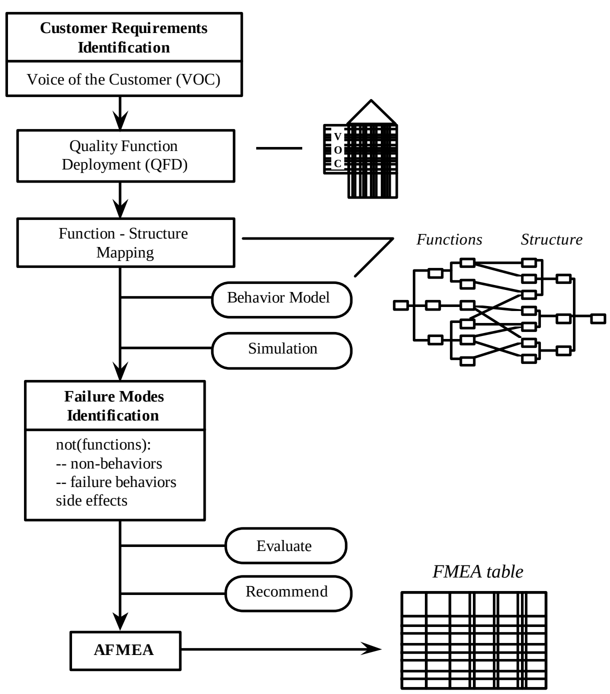

Figure 4: Flow Chart of the AFMEA Process

3.1 Procedure for Using the Behavior Model

We begin by looking at the key quality characteristics which the device must deliver. Quality Function Deployment (QFD) helps identify these attributes (Hauser and Clausing, 1985.) Performing QFD first is especially useful, since it both identifies customer requirements, and relates the VOC to the engineering metrics and functional requirements responsible for satisfying the customer.

Constructing the behavior model for a system is similar to traditional functional decomposition. We start by looking at the overall behavior of the device, then assign variable attributes that identify how the system affects input and output parameters. The design team examines the overall behavior and decomposes it into sub-behaviors and assigns the appropriate pre- and post-condition state variables. For example, we can use input parameters and output requirements for the ice maker in spreadsheet form as shown in Table 2. Table 3 shows an example of decomposition of the behavior "create cubes."

3.2 Types of Failure Modes

Through qualitative simulation, the behavior model generates three types of failure modes: non-behaviors, undesired behaviors, and misbehaviors.

3.2.1 Non-behavior Failures

While following the modeling steps given in Section 2, we select a behavior for investigation, consider it not to occur, and simulate how the system responds. Once the simulation process is complete, we compare the list of *resulting* final state variable values with the list of *desired* values to indicate which system or component failed.

3.2.2 Undesired Behavior Failures

In addition to failures due to non-behaviors, one may define failure behaviors and associated state variables corresponding to failure modes. It is important that we identify failure behaviors, as they will affect the flow of the simulation algorithm. When a failure behavior appears as a candidate for execution, we execute the failure behavior only, and recompile the list of candidate behaviors. Failure behaviors result in state variable values that generally eliminate one or more desired candidates.

3.2.3 Misbehavior Failures

We define misbehaviors and incorporate them into the behavior model. Misbehaviors do not affect the flow of the simulation algorithm since they do not affect the behavior path. When a misbehavior appears as a candidate, we execute it first, then check the list of candidate behaviors to see if it remains the same, minus the misbehavior. If the list of desired candidate behaviors has changed, then the misbehavior has been improperly defined.

4 VALIDATION EXAMPLE: ICE MAKER

This section illustrates specific examples of failure modes as they relate to the operation of an automatic ice maker. The ice maker used in this example is shown in Figure 5.

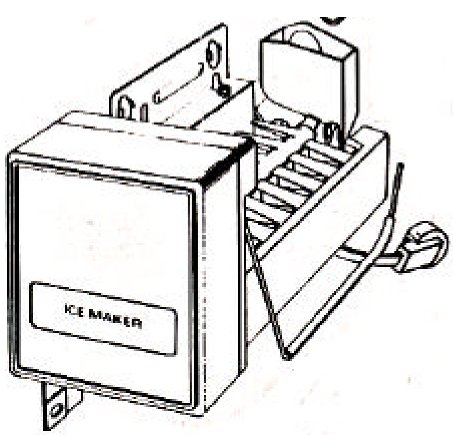

Figure 5: Ice Maker

Although we are applying AFMEA to a mature design rather than a conceptual design, the approach we take is from a high, abstract level. We begin from the general behavior “deposit ice cubes in bucket,” and decompose the behavioral description so we can relate behaviors to general structural elements and other systems. AFMEA applies to any hierarchical level of the behavior model, and the further the decomposition progresses, the closer the behaviors map to specific components and systems. Thus AFMEA can be initiated during the early design stages and updated and extended as the design develops.

4.1 Undesired Behavior Example

Some problems with the ice maker are less-than-obvious failure modes, completely ignored by the component-based FMEA. For example, the designers decided to use water volume for the cube mold is set by adjusting the fill time assuming a standard water line pressure. The total fill volume expressed as a function of time is:

$$V = \sqrt{\frac{2p}{r}}\,A\,t \tag{5}$$

where: V = volume

p = pressure

r = density

A = cross-sectional area

t = time

The fill time is set to 7 seconds, providing water volume of 190 cc ± 10 cc using 6.35 mm tubing and nominal water pressure of 138 kPa ± 14 kPa. Consequently, over-fill and under-fill failures occur when pressure is above or below nominal conditions, respectively. Table 4 describes the filling of the mold with water, decomposed into two sub-behaviors.

By changing the expected pre-conditions for the “fill mold” behavior (e.g. water pressure) we can define failure behaviors from the undesired pre-conditions. We define two failure behaviors representing the under-fill and over-fill conditions. Undesired behaviors create pre-conditions which can enable subsequent undesired behaviors, such as spillage in the case of an over-fill, or small ice cubes in the case of an under-fill.

4.2 Non-behavior Failure Example

Failure modes can be captured by examining non-behaviors of the system, for example, a thermostat that fails to close when prescribed. The analysis appears as follows:

Failure mode: **not** ($b_{1.3.1.1.1}$: thermostat closes)
  suspect components: thermostat
Resulted in deviation:
  not(ice bucket, cube level, full) resulting from:
    not ($b_{1.3.2.4}$: ejector pushes ice)
  not (mold, ice cubes present, no) resulting from:
    not ($b_{1.3.2.4}$: ejector pushes ice)
  not (mold, water level, empty) resulting from:
    not ($b_{1.3.2.4}$: ejector pushes ice)
  not(mold, temperature, >32°) resulting from:
    not($b_{1.3.1.1.4}$: heater heats mold)

Note: the subscripts indicate the level of behavioral decompostion consistent with table 2; a period is added for each indenture of decomposition. The list of non-behaviors is generated by searching for discrepancies at the lowest levels of abstraction. In terms of FMEA, we have generated the “local” effects. Next, we generate the “end” effects by investigating the effects of the non-behavior on top level behaviors. For this example, the end effect is the behavior “harvest cubes” behavior does not occur:

Failure mode: **not**($b_{1.3.1.1.1}$: thermostat closes)
  suspect components: thermostat
Resulted in deviation:
  not(ice bucket, cube level, full) resulting from:
    not($b_{1.3}$: harvest cubes)
  not(mold, ice cubes present, no) resulting from:
    not($b_{1.3}$: harvest cubes)

4.3 Misbehavior Failure Example

Slight misalignment of the ice maker causes variations in the sizes of the ice but does not otherwise affect the operation of the ice maker. We term this type of discrepancy a misbehavior or side effect. To ensure an even fill of the mold, the ice maker must tilt two degrees forward with respect to the refrigerator cabinet, and manufacturing takes special precautions to meet this requirement. However, suppose the refrigerator is not leveled properly with respect to the floor during installation.

Slight misalignments only cause the side effect of non-uniform cube size. A severe backward tilt results in shallow water level in the mold end nearest the thermostat, which freezes earlier than the deeper water at the opposite end of the mold. The shallow frozen cubes trigger the thermostat and initiates the harvest cycle, resulting in a mixture of small cubes and partially frozen cubes in the collection bucket.

If the tilt is severe enough, the larger cubes break when deposited, spilling water in the collection bucket and fusing the cubes together. This failure is not associated with the ice maker components or refrigerator. It is caused by an interaction with an external factor: alignment of the refrigerator to the floor. A component-based FMEA would consider this to be outside the scope of the ice maker and its components, however, this failure mode directly affects the quality of the ice cubes. Table 5 describes the effects of misalignment as two misbehaviors and a failure behavior, depending on the orientation and severity of the tilt.

Table 4: Mold Fill Behavior Decomposition

Behavior				Pre-condition Specification			Post-condition Specification		
index	behavior	type	mapped to	object	attribute	value	object	attribute	value
1.2.1.3	fill mold	desired	water tube	mold	water level	empty	mold	water level	full
			cup	mold	ice cubes present	no			
			water	water valve	status	open			
				water	pressure	<p>			
1.2.1.3.1	convey water to mold	desired	fill switch	water	pressure	<p>	water	fill volume	6.06*sqrt(p)*t
			water	fill switch	time setting	<t>			
				mold	water level	empty			
				mold	ice cubes present	no			
				water valve	status	open			
1.2.1.3.2	fill mold to proper level	desired	water	water	fill volume	^{3}180 & ^{2}200 cc	mold	water level	full

Table 5: Ice Maker - Alignment Misbehavior and Failure Behaviors

Behavior				Pre-condition Specification			Post-condition Specification		
index	behavior	type	mapped to	object	attribute	value	object	attribute	value
1.2.1.5m	misalignment and uneven cubes	misbehavior	ice maker refrigerator	ice maker	alignment	>-9° AND < -4°	ice cubes	size	non-uniform
1.2.1.5m	misalignment and uneven cubes	misbehavior	ice maker refrigerator	ice maker	alignment	>0° AND <5°	ice cubes	size	non-uniform
1.2.1.5.1f	misalignment and hollow cubes	failure	ice maker refrigerator	ice maker	alignment	³ 5°	ice cubes	size	non-uniform

4.4 Capturing More Failure Modes

AFMEA can capture a richer set of failure modes than traditional FMEA using the increased scope of the behavior model and analyzing many variables contributing to the system's function. While component failures *are* identified using AFMEA, many other failure modes are addressed which do not necessarily relate to components. These undesired behaviors are found when analyzing the interaction with other mechanical, hydraulic, and electrical interfaces such as collection bucket placement, incoming water pressure, or supply power. In addition, undesired behaviors are identified by accounting for the device's interaction with the environment, for example, freezer temperature or alignment. The ice maker example shows that we can obtain insight into the operation of any subsystem by "violating" nominal operating parameters. Table 6 illustrates failure modes identified using AFMEA, providing a more comprehensive analysis of the system's reliability at an earlier point in time.

Table 6: Comparison of Failure Modes Captured by FMEA vs. AFMEA

Failure Mode	FMEA	AFMEA
thermostat failure	yes	yes
water switch failure	yes	yes
feeler arm damaged	yes	yes
power cord disconnect	yes	yes
high/low water pressure	no	yes
bucket misplacement	no	yes
refrigerator misalignment	no	yes
iced gears	no	yes
high freezer temperature	no	yes

This comparison was conducted between the original design FMEA for the ice maker and the failure modes captured using AFMEA and the behavior model. In order to compare our approach to a component-based FMEA, we needed to analyze a complete and detailed design. The advantage of AFMEA is that it can be initiated much earlier in the design phase by assessing failure modes associated with high-level functions and behaviors. The logical next step is to apply AFMEA to a conceptual design as it develops, and future applications of AFMEA to conceptual designs are discussed in the next section.

5 CONCLUSIONS AND FUTURE WORK

This paper presented a new method for performing FMEA based on a device behavior model, allowing designers to address ownership quality at the early stages of conceptual design more effectively. Using an automatic ice maker as an example, the paper demonstrated how AFMEA can capture failures normally missed by traditional component-based FMEA by defining cause-and-effect relationships between system-wide design variables, environmental interaction, and sub-system quality measures. The result is a systematic method capable of capturing a wider range of system failure modes and effects early in the design cycle.

We are currently applying AFMEA to the development of a new combined cycle power generation system, well before the design of the plant and its central systems are finalized. Applying AFMEA to a complex conceptual design helps to refine the procedure and validate the approach as a design tool at the conceptual stage.

Many challenges lie ahead in the development of the behavior model for advanced FMEA. A major challenge is to address failure modes in transition situations, for instance, during the start-up of a power plant. Many failure modes occur when transitioning from one quasi steady-state mode to another through a sequence of specified behaviors, what we refer as meta-behaviors (Figure 6).

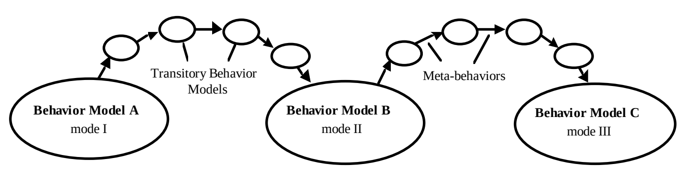

Figure 6: Meta-behaviors Describing Transition of Operational Modes

Using meta-behaviors we can model operations of a system structure that causes its functional constructs to change; that is, the function to structure mapping changes. Extending the behavioral model to include failure modes associated with non-steady state event would address, for example, sequences of switch actuation, valve operations, and sensor errors. Thus, meta-behavior modeling and simulation should provide the foundation for applying AFMEA to transient operations. By focusing on behavioral transition during start-up and shut down sequencing we can directly impact design of controls and monitoring systems. Such an extension would complement the quasi-steady state or “in-mode” AFMEA described in this paper.

Specific future activities are planned to develop further and validate the AFMEA procedure. While continuing to apply AFMEA to the conceptual design of a new power generation system, Stanford's ME217 graduate life-cycle design course will serve as a test-bed for AFMEA. Student teams will use AFMEA for reliability-related industry sponsored projects and both project teams and industrial associates will provide feedback on the approach. Our continued collaboration with Stanford Linear Accelerator (SLAC) presents an additional opportunity to use AFMEA on the design of the next generation linear collider.

ACKNOWLEDGMENTS

This research was funded by General Electric and the National Science Foundation. The authors would like to express sincere appreciation to Dr. Mahesh Morjaria for his continued support in the development of the behavior model, and to Dr. Harindra Rajiyah for his valuable feedback on the implementation of Advanced FMEA.

REFERENCES

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

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*, 8(2), pp. 127-140.

Eubanks, C. F. (1996), "A Framework for Computer-based Failure Modes and Effects Analysis of Mechanical Systems in the Conceptual Design Phase," Ph.D. Thesis, Graduate School, The Ohio State University, December, 1996.

Eubanks, C., S. Kmenta, and K. Ishii (1996), "System Behavior Modeling as a Basis for Advanced Failure Modes and Effects Analysis," ASME Computers In Engineering Conference, Sept. 1996, Irvine, CA.

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

Hauser, J., and D. Clausing (1985)."The House of Quality," *Harvard Business Review*, May-June, pp. 63-73.

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

Iwasaki, Y and C. Low (1991), "Model Generation and Simulation of Device Behavior with Continuous and Discrete Changes," Knowledge Systems Laboratory Report No. KSL 91-69, Dept. of Computer Science, Stanford University, November, 1991.

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

Kara-Zaitri, C., Keller, A., Barody, I., and Fleming, P., "An Improved FMEA Methodology," Proceedings of the 1991 IEEE Annual Reliability and Maintainability Symposium, pp. 248-252.

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.

Levinson, C. (1964), "System Reliability Analysis," Mechanical Design and Systems Handbook, (H. Rothbart, ed.) McGraw-Hill, Inc., New York.

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.

McKinney, B. (1991), "FMECA, The Right Way," Proceedings of the 1991 IEEE Annual Reliability and Maintainability Symposium, pp. 253-259.

Montgomery, T., Pugh, D., Leedham, S. and Twitchett, S. (1996), "FMEA Automation for the Complete Design Process," Proceedings of the 1996 IEEE Annual Reliability and Maintainability Symposium, pp. 30-36.

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, Schenectady, NY, April, 1992, 35 pages.

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.

Palumbo, D. (1994) "Automating Failure Modes and Effects Analysis," Proceedings of the 1994 IEEE Annual Reliability and Maintainability Symposium, pp. 304-309.

Stamatis, D. (1996), *Failure Mode and Effect Analysis*, ASQC Quality Press, Milwaukee, 1996.

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

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

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.