

DETC99/DFM-8939

## ADVANCED FAILURE MODES AND EFFECTS ANALYSIS OF COMPLEX PROCESSES

Steven Kmenta  
Peder Fitch  
Kosuke Ishii  
Design Division  
Mechanical Engineering Dept.  
Stanford University  
Stanford, CA 94305-4022

### ABSTRACT

This paper presents the use of Advanced Failure Modes and Effects Analysis (AFMEA) as a methodology to analyze manufacturing process reliability. The proposed method applies to early process design and seeks to improve product quality, process efficiency, and time to market. The method uses behavior modeling to relate process functions, performance state variables, and physical entities. The model can be used to define process failures explicitly and provides a framework for assessing causes and effects. An example of a precision turning operation illustrates how AFMEA applies to the analysis of manufacturing processes. A pilot analysis of an ultrasonic inspection process revealed that AFMEA is comprehensive and adaptable to other processes. Ongoing work for AFMEA is developing deployment strategies for minimal time burden and links to embedded error proofing.

KEYWORDS: behavior modeling, process FMEA, reliability

### 1. INTRODUCTION

Process reliability assessment is becoming an integral part of product development and is often based on statistical models (DeVor et al., 1992). However, traditional process reliability methods require information that is usually not available until late in the development process. At the detailed design stage, the majority of cost and reliability have already been “designed-in” to the manufacturing process. Development teams need to address product *and* process reliability up-front. Failure Modes and Effects Analysis (FMEA) is a tool for analyzing both product and process reliability, potentially, at an early-stage (Bowles, 1998).

#### 1.1 Failure Modes and Effects Analysis

FMEA is an engineering technique used to identify, prioritize and alleviate potential problems from the system,

design, or process before the problems are actualized (Omdahl, 1988). What is a failure mode? The literature offers many definitions of a “failure mode.” According to the Automotive Industry Action Group (AIAG), a failure mode is “the way in which a product or process could fail to perform its desired function” (AIAG, 1995). Some sources define “failure mode” as a description of an undesired cause-effect chain of events (MIL-STD-1629A, 1994). Others define “failure mode” as a link in the cause-effect chain (Stamatis, 1995; Humphries, 1994). To avoid confusion, we introduce the term *failure scenario* to describe an undesired sequence of causes and effects.

Once development teams identify and prioritize failure scenarios, they can make design decisions leading to improved reliability, quality, and safety. Table 1 describes the three main phases of FMEA.

Table 1. Three phases of FMEA

Phase	Question	Output
<i>Identify</i>	What can go wrong?	Failures: causes & effects
<i>Analyze</i>	How likely is a failure and what are the consequences?	Risk Priority Evaluation (likelihood × severity)
<i>Act</i>	What can be done to eliminate the cause or alleviate the severity?	Design solutions, test plans, manufacturing changes, error proofing, etc.

Identifying failures is a critical aspect of FMEA. It is impossible to evaluate and alleviate a potential failure that is not anticipated. This paper introduces a novel and systematic

method to identify *what can go wrong* in complex manufacturing processes.

### 1.2 Process FMEA

Process FMEA is similar to other types of FMEA: the goal is to identify what problems may occur within a manufacturing process. Process failures fall into two categories: internal and external failures. Internal failures are detected within the process (e.g., scrap, rework, delays, equipment damage, false inspection rejections, safety concerns). External failures are not detected within process boundaries (e.g., process failures resulting in warranty cost, unscheduled maintenance). External failures are much more costly than internal failures (DeVor et al., 1992). Figure 2 illustrates internal and external failures.

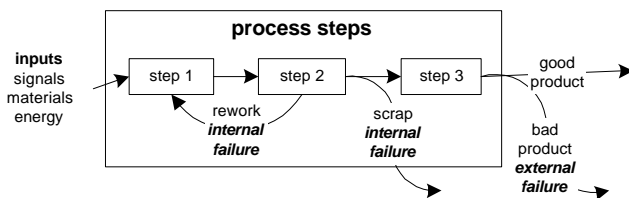


Figure 2. Basic process failures

### 1.3 Shortcomings of FMEA

Three problems with traditional FMEA, in order of importance, are:

- 1) FMEA is performed too late and not used to influence design decisions.
- 2) FMEA does not capture many potential failures.
- 3) The process for performing FMEA is subjective and tedious.

Examples of documented shortcomings of FMEA are:

- FMEA is applied too late and in such detail that it misses key system-wide, in-service failure modes (Bednarz and Marriott, 1988).
- Performing FMEA late does not affect important design and process decisions (McKinney, 1991).
- The analysis is often an afterthought, performed as a “box-checking” exercise (Kara-Zaitri et al., 1991).
- Without a systematic approach, engineers produce a subjective analysis that depends on their experience level (Bell et al., 1992).
- FMEA is tedious and time-consuming (Ormsby et al., 1991).

FMEA has the most leverage when applied at the early stages of product development since product and process cost and reliability are already fixed after the concept and layout design stages. Figure 1 compares the early and continuous application of FMEA to what often happens: performing the FMEA late or not at all.

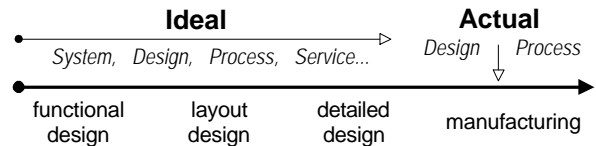


Figure 1. Ideal vs. actual deployment of FMEA

Identifying potential failures is a critical step in assessing reliability, and the literature does not prescribe *systematic* procedures for identifying failures. Our industry collaborators have not had success using FMEA to identify many in-service and in-process failure modes. It is difficult to anticipate many problems associated with:

- interfaces with other processes, systems, and suppliers;
- interaction with controls;
- unexpected operating conditions;
- errors between process operations;
- unanticipated operator or user actions;
- omitted operations that are difficult to detect.

Ideally, Process FMEA should be a systematic process analysis tool that is repeatable (consistent), adaptable to any process, and identifies a more comprehensive set of problems. Section 2 gives an overview of an enhanced version of FMEA called Advanced FMEA (AFMEA). Section 3 proposes a procedure for performing AFMEA on a manufacturing process using a simple example. Section 4 applies AFMEA to a high-precision turning process. Section 5 discusses the relative advantages and drawbacks of structured process analysis compared to brainstorming. Section 6 outlines limitations and potential extensions of this work.

## 2.0 ADVANCED FMEA

Reliability assessment is usually an in-depth engineering analysis performed during detailed design. In-depth analyses are important, yet prone to miss many system-wide failures. Reliability assessment could benefit from simple, system-wide tools performed at an early stage.

Advanced FMEA, first proposed by Eubanks et al. (1996), addresses some of the deficiencies associated with traditional FMEA. AFMEA uses behavior modeling to simulate device operations and to help reason about causes and effects. The goal of AFMEA is to provide a systematic method of capturing a larger set of failure modes early in the design.

### 2.1 Behavior Modeling for Advanced FMEA

Two accepted approaches to FMEA are based on 1) components, and 2) functions (AIAG, 1995: ARP926A, 1979). The two approaches are complementary but not mutually exclusive. Hawkins and Woollens (1998) suggest that the functional approach is suitable for early stages of design. Since functions can be abstracted prior to selection of specific components, development teams can analyze products in this

manner before the majority of cost and reliability are locked into the design (Sturges et al, 1996). In order to leverage reliability analysis at an early-stage, AFMEA uses a behavior model that defines relationships between:

- functions;
- states: pre-conditions (“what is required”) and post-conditions (“what is expected”) of each function;
- elements: physical entities that enable functions to achieve the desired post-conditions.

Such a behavior model represents a system as a causal sequence of functions and states (Keuneke, 1991). Once the model is built, we can insert failures in the form of undesired state variables, propagate the effects, and assess causes. Figure 3 shows a flowchart for Advanced FMEA.

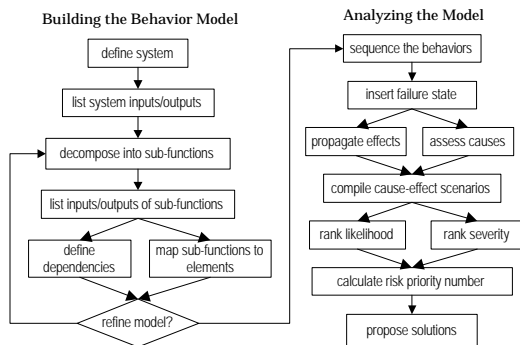


Figure 3. Flowchart for Advanced FMEA

## 2.2 Applying Advanced FMEA to Complex Processes

Our group has applied AFMEA to both detailed and conceptual designs, and the concurrent design of controls and hardware (Table 2).

Table 2. Previous research applications of AFMEA

Applications of AFMEA	Case Studies	Primary Lesson
	<p>ICE MAKER</p> <p>Retroactive case study on an existing design</p>	<p>AFMEA is more comprehensive than traditional FMEA</p> <p>(Eubanks et al., 1996 Eubanks et al., 1997)</p>
	<p>POWER PLANT HARDWARE</p> <p>Conceptual layout design</p>	<p>AFMEA is useful for early system configuration decisions</p> <p>(Eubanks, 1997)</p>
	<p>PLANT CONTROLS &amp; HARDWARE</p> <p>Concurrent design of controls and hardware</p>	<p>AFMEA is effective for concurrent design of controls and hardware systems</p> <p>(Kmenta and Ishii, 1998)</p>

We have show AFMEA to be useful in a wide variety of applications. Behavior modeling uses a simple framework for representing systems at a high level and it is decomposable to model details of a system. The flexibility of the behavior model lends itself to manufacturing processes as well. Process behavior models are analogous to other models (Figure 4).

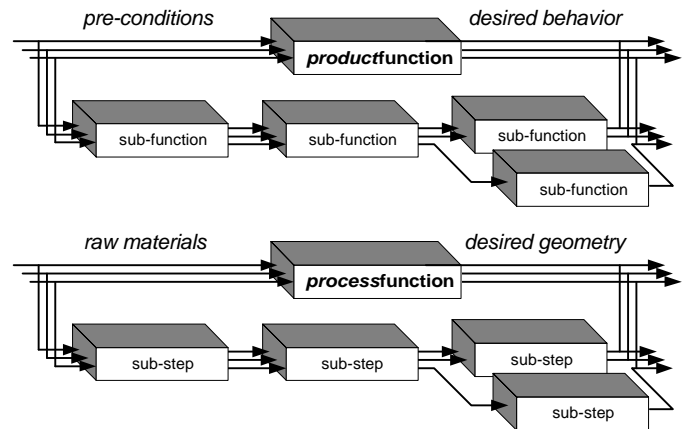


Figure 4. Comparison of product and process models

Behavior modeling can be an effective tool for simulating complex manufacturing processes. We have used behavior modeling to partially analyze several complex processes, such as: ultrasonic inspection of titanium disks, rabbet joint turning, and turbine disk assembly. Using behavior modeling for manufacturing processes offers several advantages:

- a more systematic method for identifying process failures;
- applicability to preliminary process designs, as well as for mature processes;
- the ability to incorporate system level variables in a process model, as well as interfaces to other processes;
- the ability to link potential process failures to process and customer requirements.

The next section proposes a procedure for identifying process failures using behavior modeling.

## 3.0 ADVANCED FMEA APPLIED TO PROCESSES

This section outlines the general procedure for building a process model and then performing AFMEA. We will use a simple example of attaching a cap to a tube of toothpaste using automated equipment.

### 3.1 Building the Model

The procedure for building the behavior model for a manufacturing process parallels the procedure for system models outlined by Eubanks (1996).

### Step 1 Define the boundary

When building a behavior model, one must scope the analysis by defining a boundary to the process. In the case of a manufacturing process, the system boundary might be a plant, a manufacturing line, a manufacturing cell, etc. Once the boundary is defined, we can begin to think about the inputs to the system, what physical entities are contained within the boundaries, and what are the desired outputs.

*Example:* Automated cap attachment station, the equipment associated with this area.

### Step 2 State the process function

State the overall purpose of the process in a verb + noun format.

*Example:* "seal tube"

### Step 3 List the desired output

List the resulting desired state, or post-conditions, of the process in <variable><attribute><value> (Eubanks, 1996) format. These can include process requirements. We indicate true/false values using [1, 0] and list nominal conditions as "nom."

*Example:* <tube> <sealed> <1>

### Step 4 List the inputs as system pre-conditions

List the required external inputs, or pre-conditions to the system including signals, materials, and energy that are entering the process boundary. Inputs can usually be categorized in the following areas:

- Energy:** power, force, friction
- Information:** data, bar codes, paperwork
- Material:** fluid flow, components

*Example:* <tube> <present> <1>,  
 <cap> <present> <1>,  
 <supply air> <pressure> <60psia>,  
 <vacuum line> <pressure> <nom>

### Step 5 Decompose the process

Decompose the process into sub-functions (corresponding to sub-steps) and list the desired outputs and required inputs.

*Example:* Figure 5 shows the overall function "seal tube" decomposed into sub-functions, e.g., "import tube," "get cap," and "attach cap." We can decompose sub-functions further into sub-sub-functions; for example, "get-cap" contains sub-sub-functions "sense cap," and "pick cap." This is the manner in which functions are decomposed into a functional hierarchy. For the rest of this paper, the term "function" will also refer to sub-functions, sub-sub-functions, etc.

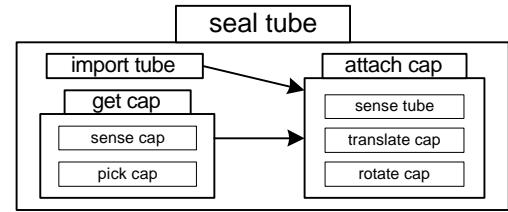


Figure 5. The overall process

We establish a hierarchy of behaviors by adding pre- and post-conditions to all of the functions (Figure 6). More detailed information could be incorporated, such as specification of the cap threads, the rotation torque, etc.

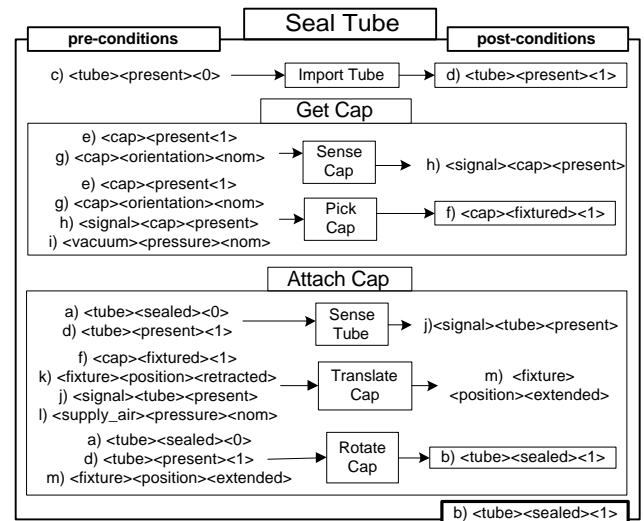


Figure 6: Pre-conditions and post-conditions included in model

### Step 6 Assign influences

Use arrows to assign dependencies between states and functions

*Example:* (Figure 9).

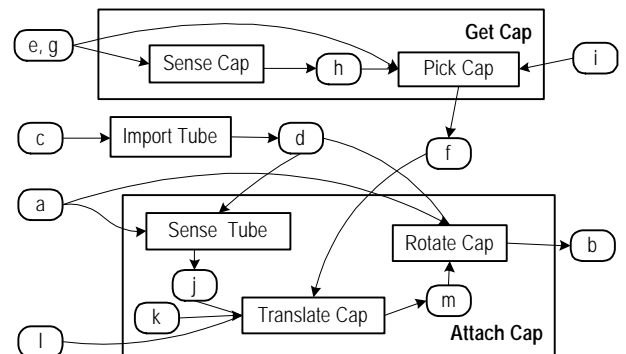


Figure 7. Influence diagram based on a behavior model

### Step 7 Map elements to functions

Elements are the physical entities and agents responsible for performing functions and achieving post-conditions. Map each function to one or more element(s).

*Example:* We map the function “pick cap” to these elements:

{ *fixture, vacuum line, controller* }

### 3.2 Performing the Analysis

This section lists the steps for applying Advanced FMEA to a process model. The steps continue from the previous section and propose a method for analyzing a model for failure modes and effects, and reasoning about their causes. We will continue to use the attachment of the cap to the toothpaste tube as an example.

### Step 8 Sequence the behaviors for analysis

Many behavior models, particularly models of manufacturing processes, have functions and states that only influence downstream functions and states, with very little feedback. It is desirable to simulate failure conditions straight from the beginning of the model to the end. However, not all models are linear and free from feedback loops.

For complicated models we recommend optimizing the order used to analyze the model in such a way that precedence is maintained as much as possible. When analyzing a behavior, we would like to have already analyzed failures of preceding behaviors. Maintaining precedence helps us to identify causes before effects. *Dulmage Mendelsohn (DM) Decomposition* is a graph theoretic ordering technique that computes the block triangular form of sparse asymmetric matrices. DM Decomposition is a means of maintaining precedence and minimizing feedback loops in a directed graph. This approach is a systematic answer the question “*where do I begin the analysis?*” given a behavior model.

*Example:* From the influence diagram, shown in Figure 7, we establish the dependency matrix for state variables *a* to *m*, shown in Table 3. We initially choose the sequence: *a, b, c, d, e, f, g, h, i, j, k, l, m*.

**Table 3.** Dependency matrix for the analysis sequence

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>
<i>a</i>	•												
<i>b</i>	•	•		•	•								•
<i>c</i>			•										
<i>d</i>			•	•									
<i>e</i>					•								
<i>f</i>					•	•	•	•	•				
<i>g</i>							•						
<i>h</i>							•	•					
<i>i</i>								•					
<i>j</i>	•			•						•			
<i>k</i>											•		
<i>l</i>												•	•
<i>m</i>													•

Marks above the diagonal (such as states *b* and *f*) represent violations of precedence. Table 4 shows the ordering using DM Decomposition.

**Table 4.** The re-ordered sequence using DM decomposition

	<i>a</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>f</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>b</i>
<i>a</i>	•												
<i>c</i>		•											
<i>d</i>			•										
<i>e</i>				•									
<i>g</i>					•								
<i>h</i>						•	•						
<i>i</i>							•						
<i>f</i>								•	•	•			
<i>j</i>	•								•				
<i>k</i>										•			
<i>l</i>											•		
<i>m</i>												•	•
<i>b</i>	•												•

The re-ordered sequence does not violate precedence conditions and improves the efficiency of analysis. The analyst might be able to generate this sequence by inspection. However, DM Decomposition helps select an appropriate analysis sequence for models with many nodes and feedback loops.

### Step 9 Insert failures

For a given behavior (pre-conditions + function + post-conditions) we define failure as any deviation from the desired post-condition. We can generically represent a failure as the negation of the a desired post-conditions:

$$\text{not}(\langle \text{variable} \rangle \langle \text{value} \rangle \langle \text{nom} \rangle)$$

An undesired state is any deviation from the intended state. For example, for the function “provide flow” with the post-condition  $\langle \text{water} \rangle \langle \text{flowrate} \rangle \langle 5-6 \text{ lpm} \rangle$ , anything other than the nominal range is considered a failure. Our classification of variable values is as follows:

nom	<i>nominal condition</i>
1	<i>True</i>
0	<i>False</i>
+	<i>too-high</i>
-	<i>too-low</i>
∅	<i>none or absence</i>

Using this convention,  $\text{not}(\langle \text{value} \rangle)$  could include any of the set of { $\langle + \rangle$ ,  $\langle - \rangle$ ,  $\langle \emptyset \rangle$ ,  $\langle \text{other-value} \rangle$ }. The use of the general failure states *too-high* and *too-low* are useful in many practical examples, giving a good compromise between simplicity and completeness of analysis (Chittaro, et al., 1998).

*Example:* For the failure “cap is not fixtured”

$$\text{not}(\langle \text{cap} \rangle \langle \text{fixtured} \rangle \langle 1 \rangle) \text{ (equivalent to } \langle \text{cap} \rangle \langle \text{fixtured} \rangle \langle 0 \rangle)$$

### Step 10 Propagate effects

Given an undesired post-condition, what other behaviors will be affected? Post-condition failures can become pre-condition violations for subsequent behaviors.

*Example:* For the failure:  $\text{not}(\langle \text{cap} \rangle \langle \text{fixtured} \rangle \langle 1 \rangle)$

The function-state behaviors would be affected as follows:

- not(<cap> <fixtured> <1>)
- not(translate cap)
- not(<fixture><position><extended>)
- not(rotate cap)
- not(<tube><sealed><1>) – the tube is not sealed.

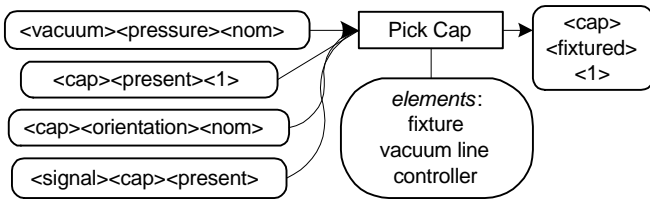
**Step 11 Assess causes**

The causes of the failures can be reduced to two categories:

- 1) Element failures: failure of elements mapped to a function.
- 2) Pre-condition failures: failures of other behaviors within the system (internal) or of inputs to the system (external). Pre-condition failures are generally linked to element failures elsewhere in the model.

The compiled list of element and pre-condition failures comprises a set of potential causes for a failure scenario.

*Example:* For the failure <cap><fixtured><0> we can reason about potential causes. Figure 8 shows the behavior-structure fragment for the function “pick cap.”



**Figure 8.** Behavior fragment for analyzing causes

Generic element failures would include the set:

- { not(fixture), not(controller), not(vacuum line) }

Pre-condition failures include the set:

- { not(<cap><present><1>), not(<signal><cap><present>), not(<vacuum><pressure><nom>), not(<cap><orientation><nom>) }

Pre-condition failures can be linked to element failure:

- { not(hopper), not(sensor), not(vacuum pump), not(controller) }

There may be many specific causes for a generic element failure “not(element)” and the exact nature of the failure may vary. Specific causes can be included explicitly into the analysis. When detailed causes of failure are not known, we recommend including generic element failures for several reasons:

- they act as placeholders for more specific information;
- they account for the relationship between physical elements and behaviors;
- generic failure modes are amenable to the early application of FMEA;
- we can reason about impact on the system performance even without specific information about causes of failure;

- *specific* physical causes for an improper behavior can be incorporated later as pre-conditions, for example <fixture><alignment><nom>.

**Step 12 Compile failure scenarios**

Failure scenarios are the links of an undesired cause-effect chain, and include a list of element failures, state failures, and function failures. Several failure scenarios can be associated with each function.

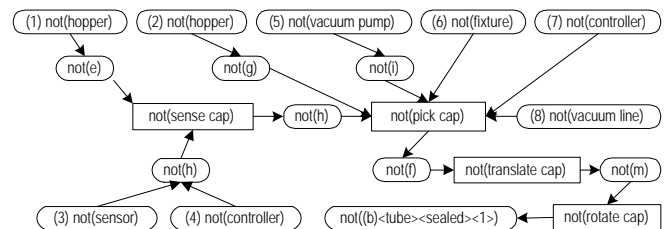
*Example:* For the failed function not(pick cap), all failures can be traced to elements and result in an unsealed tube. Eight failure scenarios associated with not(pick cap):

- (1) not(hopper) → not(<cap><present><1>) [state “e”]
- (2) not(hopper) → not(<cap><orientation><nom>) [state “g”]
- (3) not(sensor) → not(sense cap) → not(<signal> <cap> <present>) [state “h”]
- (4) not(controller) → not(sense cap) → not(<signal> <cap> <present>) [state “h”]
- (5) not(vacuum pump) → not(<vacuum><pressure><nom>) [state “i”]
- (6) not(fixture)
- (7) not(controller)
- (8) not(vacuum line)

All scenarios result in these effects:

- not(pick cap)
- not(<cap><fixtured><1>) [state “f”]
- not(translate cap)
- not(<fixture><position><extended>) [state “m”]
- not(rotate cap)
- not(<tube><sealed><1>) [state “b”]

Failure scenarios can be represented in a diagram (Figure 9).



**Figure 9:** Failure scenarios associated with “pick cap”

#### 4.0 AFMEA EXAMPLE: TURNING APPLICATION

This section gives a demonstration of Advanced FMEA applied to a turning application with tight tolerance requirements on large-diameter parts.

#### 4.1 Behavior Model of Process

Following steps 1-7 from Section 3, we developed a model for the turning process shown partially in Figure 10.

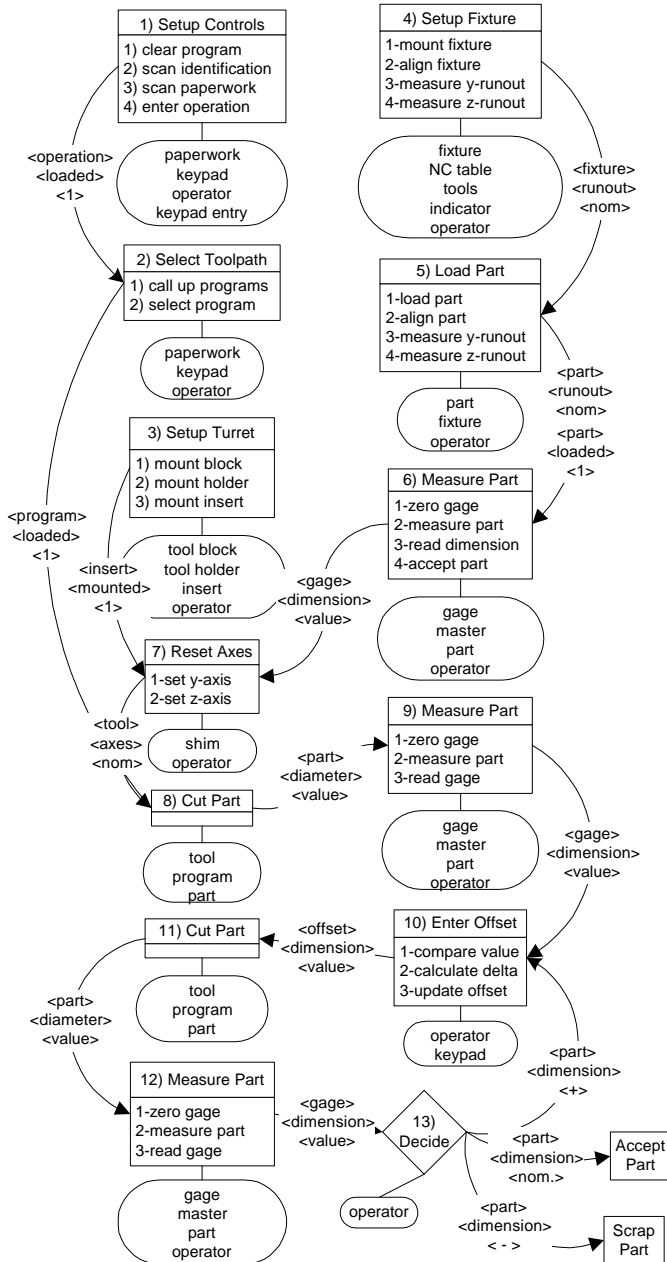


Figure 10. Partial behavior model of a turning process

Once we added states and dependencies to all the sub-functions we discovered some other dependencies not shown in the initial process map. Figure 11 shows how <part><y-runout> (step 5.3) is affected by <fixture><y-runout> (step 4.3).

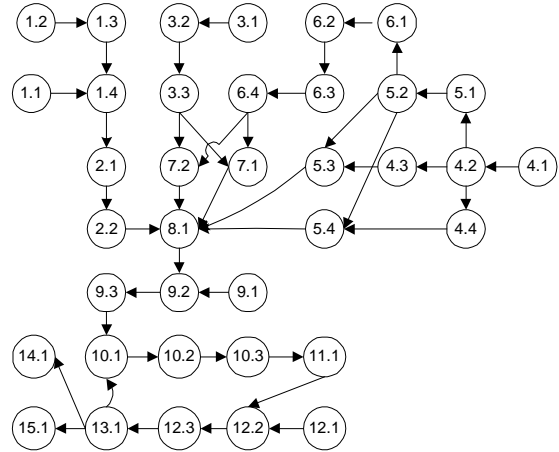


Figure 11. Functions with state dependencies

#### 4.2 Performing the Analysis

We applied DM Decomposition to the numerical sequence of behaviors and the ordering did not change. Next, we inserted a failure state, simulated the effects, and traced back to find a set of potential causes.

**Insert failure:** Fixture y-runout is too high

<fixture><y-runout>< + + >  
 → <part> <y-runout> < + + >

resulting in either

- (1) → <tool> <y-axis> < - - > or,
- (2) → <tool> <y-axis> < + + >

**Propagate effects:** case (1) the tool y-axis offset is too small

→ <tool> <y-axis> < - - >  
 → <part> <diameter> < - - >  
 → <gage> <dimension> < - - >  
 → <part> <dimension> < - - >  
 → "scrap part"

**Assess causes:** potential causes of this failure mode include:

{ not(fixture), not(NC table), not(tools), not(operator), not(indicator) }

**Compile failure scenarios:** (Figure 12)

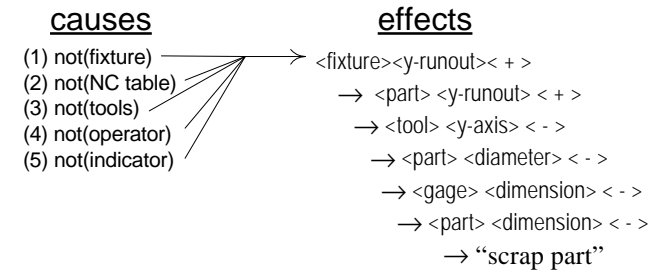


Figure 12. Failure scenarios associated with <y-runout>

## 5.0 DISCUSSION: FREE FORMAT VS. STRUCTURED FMEA

This section discusses some of the benefits and limitations of structured process analysis compared to free-format brainstorming. When initially evaluating “process AFMEA,” we took part in a workshop analyzing an ultrasonic inspection process for titanium disks. The workshop used two different approaches for identifying potential failures:

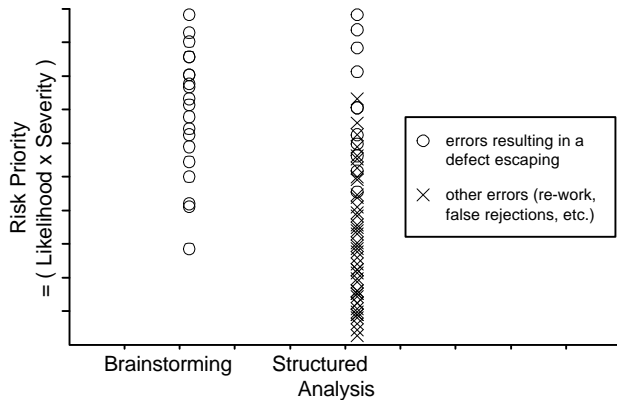
- 1) free format brainstorming session
- 2) a version of structured AFMEA

The objectives were to predict error sources for this specific process and to develop a technique for analyzing other processes. Table 5 compares the results of the two approaches.

**Table 5.** Comparison of brainstorming and a structured AFMEA analysis

	BRAINSTORMING	STRUCTURED AFMEA
Question asked	How could a defect escape?	What problems might occur?
Technique	Asked probing questions related to process steps	analyzed the process by sub-steps, using a cause-effect diagram
Facilitator knowledge of process	high	low
Adherence to process flowchart	moderate	high
No. of problems identified	32	119
No. of external failures (defects escaping)	32	25
No. of internal failures (scrap, rework, delays)	0	96
No. of overlapping failures	19	19

The brainstorming group went into more detail regarding scenarios for escaping defects, but neglected internal process problems (e.g. false rejections). The structured FMEA group covered more ground regarding general process problems, but neglected the specifics of some failures (e.g. mechanical problems). Figure 13 compares the failures identified by the two approaches qualitatively in terms of expected risk.



**Figure 13.** Comparison of relative failure risk

Structured AFMEA is transferable to other processes and does not rely on a process-specific expert. In addition, the structured approach identifies a large number of potential problems, and could be useful for situations where a specific problem or error is unknown. However, the structured analysis may limit the team’s thinking to items closely related to the process model. The structured approach also may seem too dry or tedious compared to brainstorming. Table 6 lists the advantages and disadvantages of AFMEA as perceived by the participants.

**Table 6:** Collected responses to the structured FMEA Approach

Advantages	Disadvantages
Clear structure Logical approach Completeness	Broad scope limits depth Less "off-the-wall- thinking"

The next steps are to reconcile the structured approach of AFMEA with information gathered from less formal workshops and brainstorming efforts. For example, an individual or small team could begin the AFMEA without intimate knowledge of the process. Then, the team could augment the analysis with information from focused meetings with process experts and test results. AFMEA could provide a structured framework to be populated with detailed information from many sources.

## 6.0 CONCLUSIONS AND FUTURE WORK

This paper demonstrated a new use of behavior modeling for Advanced FMEA of manufacturing processes. The model related process functions and states using causal dependencies. The paper described a method for analyzing processes and demonstrated its application on a high-precision turning example. A pilot study compared some of the benefits of a structured approach, such as a broad scope and portability, to some of its shortcomings, such as its lack of depth.

Advanced FMEA can help facilitate concurrent engineering efforts throughout both product and process development. Moreover, the method lends itself to automation as a product development tool and to act as a single repository for failure identification throughout product development. In the future, application of AFMEA could include broader areas such as supply chain management, the product development process, and other business processes. Extension and validation of AFMEA as a design methodology will include:

- integrating AFMEA with information from other methodologies (QFD, DFA, etc.);
- linking AFMEA with human error proofing;
- automating the procedure using software;
- developing improved methods for using AFMEA with minimal time and resources;



- documenting additional case examples;
- applying the behavior model to supply chain logistics and product development processes.

## ACKNOWLEDGEMENTS

The authors would like to thank General Electric and the Department of Energy for providing support for this research. The authors express sincere appreciation to Gene Wiggs, Jeff Keller, and Richard Menzies for facilitating the application of Advanced FMEA and providing valuable feedback. Thanks to Toshiki Mori, Mark Martin and Catherine Rose for their valuable ideas and assistance.

## REFERENCES

- Aerospace Recommended Practice ARP926A, 1979.
- Automotive Industry Action Group, 1995, *Potential Failure Modes and Effects Analysis (FMEA) Reference Manual*, Detroit, MI.
- Bednarz, S. and D. Marriott, 1988, "Efficient Analysis for FMEA," *Proceedings of the 1988 IEEE Annual Reliability and Maintainability Symposium*, pp. 416-421.
- Bell, D., L. Cox, S. Jackson, and P. Shaefer, 1993, "Using Causal Reasoning for Automated Failure Modes & Effects Analysis (FMEA)," *Proceedings of the 1992 IEEE Annual Reliability and Maintainability Symposium*, pp. 343-353.
- Bowles, J., 1998, "The New SAE FMECA Standard," *Proceedings of the 1998 IEEE Annual Reliability and Maintainability Symposium*, pp. 48-53.
- Chittaro, L., R. Ranon, and A. Soldati, 1998, "Introducing Deviations and Multiple Abstraction Levels in the Functional Diagnosis of Fluid Transfer Systems," *Artificial Intelligence in Engineering*, 12(4), Elsevier Science Limited, Oxford, pp. 355-373.
- DeVor, R.E., T. Chang, and J.W. Sutherland, 1992, *Statistical Quality Design and Control*, Macmillan, New York, pp. 51-57.
- Eubanks, C.F., S. Kmenta, and K. Ishii, 1996, "System Behavior Modeling as a Basis for Advanced Failure Modes and Effects Analysis," *Proceedings of the 1996 ASME Design Engineering Technical Conferences*, Irvine, CA.
- Eubanks, C.F., S. Kmenta, and K. Ishii, 1997, "Advanced Failure Modes and Effects Analysis Using Behavior Modeling," *Proceedings of the 1997 ASME Design Engineering Technical Conferences*, Sacramento, CA.
- Hawkins, P.G., and D. J. Woollons, 1998, "Failure Modes and Effects Analysis of Complex Engineering Systems Using Functional Models," *Artificial Intelligence in Engineering*, 12(4), Elsevier Science Limited, Oxford, pp. 375-397.
- Kara-Zaitri, C., Keller, A., Barody, I., and Fleming, P. (1991), "An Improved FMEA Methodology," *Proceedings of the 1991 IEEE Annual Reliability and Maintainability Symposium*, pp. 248-252.
- Kmenta, S., and K. Ishii, 1998, "Advanced FMEA Using Meta Behavior Modeling for Concurrent Design of Products and Controls," *Proceedings of the 1998 ASME Design Engineering Technical Conference*, Atlanta, GA.
- Keuneke, A., 1991, "Device Representation: The Significance of Functional Knowledge," *IEEE Expert*, 6(2), pp. 22-25.
- McKinney, B., 1991, "FMECA, The Right Way," *Proceedings of the 1991 IEEE Annual Reliability and Maintainability Symposium*, pp. 253-259.
- Omdahl, T.P. (ed.), 1988, *Reliability, Availability, and Maintainability Dictionary*, ASQC Quality Press, Milwaukee, WI.
- Ormsby, A., J. Hunt, M. Lee, 1991, "Towards an Automated FMEA Assistant," *Artificial Intelligence in Engineering, Computational Mechanics Publications*, Southampton, Boston, pp. 739-752.
- Stamatis, D.H., 1995, *Failure Mode and Effect Analysis: FMEA from Theory to Execution*, ASQC Quality Press, Milwaukee, WI.
- Sturges, R.H., K. O'Shaughnessy, and M.I. Kilani, 1996, "Computational Model for Conceptual Design Based on Extended Function Logic," *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, Cambridge University Press, 10(4), pp. 255-274.