FMEA – FRIEND OR FOE

HERBERT HECHT
Vice Chairman of the Board
SoHaR Incorporated
Beverly Hills, California, 90211

MYRON HECHT
President
SoHaR Incorporated
Beverly Hills, California, 90211

KEYWORDS

FMEA, Failure Modes, I&C Systems, Computer Based Systems, Object Oriented Development

ABSTRACT

The value of FMEA for I&C systems is being questioned because its conventional format does not appear suited to digital components where failures in software and in large scale integrated circuits predominate. It is shown that new design techniques and tools help overcome these difficulties and at the same time reduce the cost. An often overlooked benefit of FMEA is that it can show deficiencies in failure detection that, when not corrected, can impose high costs during the O&M phase.

INTRODUCTION

Failure Modes and Effects Analysis (FMEA) is an established technique for evaluating the reliability (and in some cases the safety) of electronic, electromechanical and mechanical equipment. It has been in use so long that it is questioned whether a practice that evolved in the era of discrete component analog systems is suited to the world of instrumentation and control (I&C) systems consisting of highly integrated digital circuits driven by multiple layers of software.

Most of us have heard stories of or been witness to a costly FMEA effort that resulted in useless volumes taking up space on a reliability manager's book case. Stories about an FMEA that led to improvements in a weak link or to better maintenance practices are not encountered as often, partly because they imply that the original condition was in some way deficient. In this paper we will explore what makes for a useful FMEA for the current generation of I&C systems, and then we will address ways of reducing the cost of the effort.

FRIEND OR FOE?

A formal FMEA methodology was developed in the late 1960's[1] but informal procedures for establishing the relation between part failures and system effects date back much further[2]. Yet FMEA is not ready for the rocking chair. In 2000 the Society of Automotive Engineers (SAE) published specialized FMEA procedures for the automotive industry[3]. FMEA is widely used in the process industry in support of safety and reliability[4]. It can help:

- Component designers to identify locations where added strength (or derating), redundancy or self-test may be particularly effective or desirable
- System engineers and project managers in allocation of resources to areas of highest vulnerabilities
- Procuring and regulatory organizations to determine whether reliability and safety goals are being met
- Those responsible for the operations and maintenance (O&M) phase to plan for the fielding of the system.

A formal FMEA is conducted primarily to satisfy the third bullet, an imposed requirement that does not originate in the development team and thus is a non-friend if not an outright foe. Informal studies along the lines of the first two bullets are frequently undertaken in support of the development, but they are seldom published as legacy documents. The fourth bullet is perhaps the most neglected use of the FMEA. But with growing recognition that O&M costs usually overshadow those associated with the acquisition of I&C systems that issue deserves emphasis and it will be discussed later in this paper.

In addition to these organizational issues, there are perceived technical limitations when the conventional FMEA is applied to digital I&C systems. These are discussed below and are seen to be much less relevant when contemporary system development tools are used. The outlook is very much in favor of having the FMEA recognized as a friend of both developer and user.

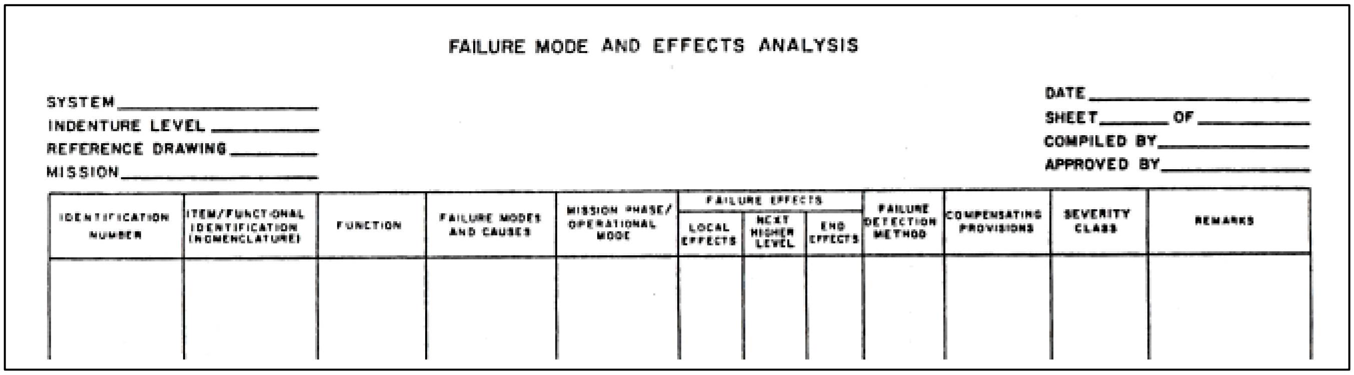
CONVENTIONAL FMEA

The essential concepts of the FMEA process are:

1. Parts can fail in several modes, each of which typically produces a different effect.
2. The effects of the failure depend on the level at which it is detected. Usually we distinguish between three levels that are described below.
3. The effects of a failure can be masked or mitigated by compensating measures (redundancy, alarms)
4. The probability and severity of in-service failures can be reduced by monitoring provisions (built-in-test, supervisory systems)

FMEA worksheets present evidence of the current state on each of these in a standardized tabular format, and this enables reviewers to identify and ultimately correct deficiencies. A summary of the worksheets is frequently prepared to call attention to failure modes that result in effects of the highest severity ("single point failure modes") or in other characteristics that are of particular concern to reviewers such as critical component lists.

The FMEA worksheet from MIL-STD-1629A[5] is shown in Figure 1. Although this standard is no longer active, its provisions serve as a generic format for FMEA documentation.



FAILURE MODE AND EFFECTS ANALYSIS

SYSTEM ____________
INDENTURE LEVEL ________
REFERENCE DRAWING ________
MISSION ____________

DATE ____________
SHEET ______ OF ______
COMPILED BY ________
APPROVED BY ________

IDENTIFICATION NUMBER	ITEM/FUNCTIONAL IDENTIFICATION (NOMENCLATURE)	FUNCTION	FAILURE MODES AND CAUSES	MISSION PHASE/ OPERATIONAL MODE	FAILURE EFFECTS			FAILURE DETECTION METHOD	COMPENSATING PROVISIONS	SEVERITY CLASS	REMARKS
					LOCAL EFFECTS	NEXT HIGHER LEVEL	END EFFECTS				

The left part of the header identifies the system and component for which this sheet has been prepared, and the right header section shows when and by whom. The failure mode identification number shown in the first column is usually a hierarchical designator of the form *ss.mm.cc.ff* where *ss* is an integer representing the subsystem, *mm* an integer representing the major component, *cc* an integer representing the lower level component, and *ff* an integer representing the failure mode. The identification number (ID) is a convenient way of referencing failure modes and is utilized in a later section of this paper.

The next two columns contain textual descriptions either for an "item" that is a single part (switch, integrated circuit, or a gear) or a function (bias supply, gear reduction). The item descriptions may apply to more than one failure mode and ID. The entries in the failure modes and causes column are unique to the ID. Where the item is a part, the failure mode description can be very simple, like "open" or "short" and the cause may be described as random failure, overload, or environmental degradation. Where the item consists of more than a single part, the description may need to be more detailed, and the cause is usually traced to parts, such as: "R1o, R4o, C2s" denoting that an open failure in resistors R1 and R4 or a shorting failure in capacitor C2 all produce this failure mode for the circuit.

Where the phase of operation causes significantly different effects for a given failure mode, the phase descriptions are entered into the next column, and separate row entries are made for each mission phase. An example is a failure mode that produces only minor effects when the plant is in maintenance mode but major effects when it is in operation. This column can be omitted where the effects do not vary significantly between operating modes

Failure effects are usually described at three levels:

Local – the function in which the failed part is located, such as an oscillator, a gear reduction, or a counter

Next higher level (NHL) – usually a partition of the system that furnishes outputs recognized by the system user, such as total flow calculation, input valve controller, or display panel.

System – the highest level of a given development project, such as annealing furnace, plant communication system, or timekeeping system.

The description of the effects serves two purposes: it alerts reviewers to the possible consequences of each failure mode, and it serves as a starting point for selecting failure detection and mitigation provisions. Failures should be detected at the lowest possible level because that permits a very specific response and reduces the probability that other failures may interfere with the detection. Thus, a local failure effect description "stops flow meter oscillator" should lead to providing a means for detecting the presence of oscillator output. This is usually simpler (in mechanization) and involves less delay than detecting failures at the NHL in the total flow calculation.

The compensation for this failure effect may be switching to a redundant oscillator or it may involve using an alternate frequency source (possibly less accurate) derived from another component. In some cases it may be preferable to defer the compensation to a higher level, e. g., by utilizing an alternative total flow calculation; or it may be decided to permit the system to be shut down if the cost of this is small and if the probability of oscillator failure is considered low.

PARTS AND FUNCTIONAL APPROACH

Hardware FMEA can use the parts approach or the functional approach. The parts approach makes use of the parts list on the component drawing but may aggregate the parts into groups that have identical failure modes as indicated previously. The functional approach can proceed without a parts list and considers only failure modes at the function interface. Attributes of each of these approaches are compared in Table 1.

Table 1. Comparison of Parts and Functional FMEA Approaches

Attribute	Parts Approach	Functional Approach
Prerequisite	Detailed drawings	Functional diagram
Completeness criterion	Based on parts list	Difficult to establish
Knowledge of failure modes	Past experience	Examination of function
Knowledge of failure rates	Public sources	No public sources
Usual progression	Bottom up	Top down
Relative cost	High	Low

The functional approach is suitable for early development phases but the parts approach is usually preferred later on because it works with parts lists and evaluates more detailed and practically meaningful failure modes. As an example, in the functional FMEA for a furnace loader the failure effect of the drive assembly may be described as "loader stops" and that will also be the means of detection. In a parts FMEA the loader will be decomposed into a motor, a small gear (pinion) on the motor shaft, other gears and the loader belt. The effect of a pinion failure can still be described as "loader stops" but the detection will be "high motor rpm and no belt movement." Thus operating and maintenance personnel can be trained to differentiate between a motor failure and a pinion failure.

FME

Because so fail in a omitted from MEA, and re failure can cause system effects that e therefore looking for means of inclu redundancy management of I&C ate for inclusion in the FMEA.

There have been many attempts to at software does not break down into modes for software are not as well lly two approaches for partitioning so considering one variable at a time[8]. approach and shares the advantages program are subjective and differen technique cannot be continued if a p ting a software FMEA based on failures software failure affects multiple var usually impractical because of the f satisfactory FMEA methodology for software.

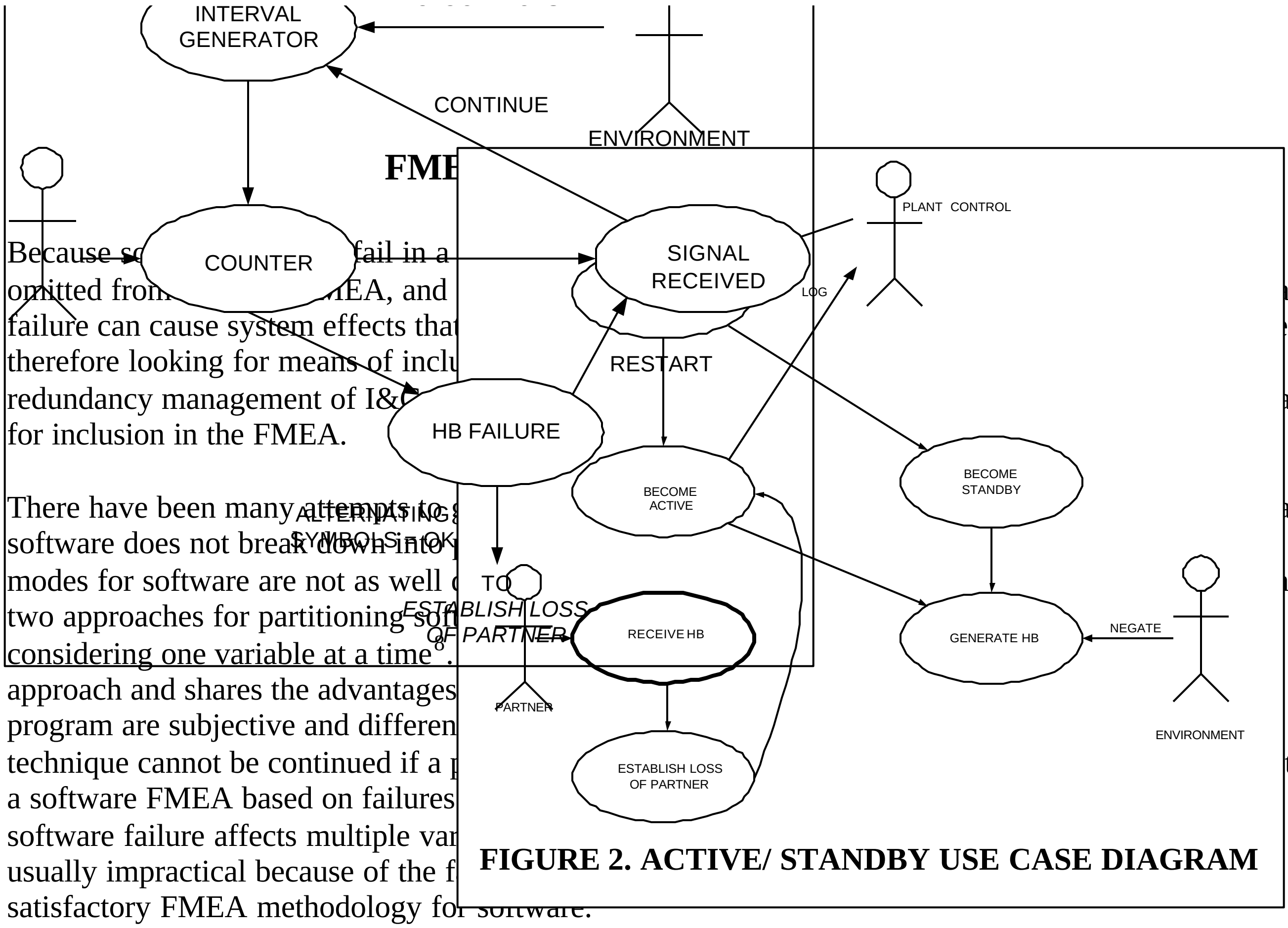


FIGURE 2. ACTIVE/ STANDBY USE CASE DIAGRAM

While most ICs can in theory be decomposed into gates or equivalent primitives that can be subjected to a part level FMEA, the effort can hardly be justified for chips containing tens of thousands of gates. Also, failure mechanisms such as thin oxide or electron-migration may affect multiple gates, thus rendering the single gate analysis useless. For these reasons ICs have in the past been primarily analyzed by the functional approach. As an alternative, the analysis of the effect of pin failures, one pin at a time, has been employed. This approach is obviously limited because most internal failures will affect multiple pins. Thus the handling of ICs in the FMEA is in many ways equivalent to that of software.

A NEW APPROACH

System development tools are being increasingly used for I&C systems, and the disciplined statement of requirements that is enforced by most of these can be exploited to generate FMEAs for software and ICs that are compatible with the part level approach. In the following example we use UML (Unified Modeling Language) to demonstrate how failure modes and effects for software can be extracted from formalized requirements statements. UML is probably the leading contemporary methodology for system development and it is well supported in the literature and by tools but the benefits described below may also be achievable in other ways.

UML based development tools permit us to take a fresh look at software partitioning that permits the part paradigm to be applied. The benefits that can be derived from UML for all forms of dependability analysis have been recognized by European researchers[9]. The key to this fresh look is that objects are uniquely characterized by their *methods* (sometimes called *behaviors*). At first glance *method* might be thought to be just another word for *function* but it is formally a part of the object structure and is not subject to the ambiguities that we have noted for functions. As long as all methods of an object are executing in accordance with their specification the object has not failed. Conversely, when a method does not execute in accordance with its specification the object has failed. The failure effect will depend on the mode in which the method failed. Also, methods can frequently be partitioned in the same way that a mechanical or electronic assembly can be partitioned.

These concepts will now be explored by means of a *use case diagram*[10] for active/standby role assignment in a plant communication system. This role assignment is a software construct that is encountered wherever components are automatically switched from active to standby status. In the implementation shown in Figure 2 the plant control initially assigns the roles but in operation the two entities manage their roles autonomously, primarily with the aid of exchanges of heartbeat (HB) data.

The use case diagram is usually the first artifact created in UML-based development. The stick figures are called *actors* and are not necessarily persons. In this example the plant control may be a human or a control system. The partner is the corresponding computer program running in the partner system, and the environment represents random external events. The ovals are the specified methods and therefore the items for which failure modes are to be established. The directed lines or arcs denote information flow and hence the paths through which failure effects propagate.

The following discussion concentrates on the RECEIVE HB method (the heavy framed oval). The most common failure mode for this method is failure to report receipt of a heartbeat when in fact it was sent by the partner. If this happens when own program is already in the active role the only action is to log the failure. If it happens when own program is in the standby role it will transition to the active role and notify plant control. Again, this is a low severity failure. But the RECEIVE HB method *may* also have a failure mode in which it signals receipt of heartbeats when the partner does not generate them. That failure mode may disable the entire plant communication system under the following scenario: Partner fails while in active mode. Own program does not note absence of heartbeats and does not take over. To determine whether this failure mode is likely to happen we need to examine the use case diagram for the RECEIVE HB method, shown in Figure 3. At that level a detailed FMEA can be generated that is comparable to the part approach for hardware FMEA

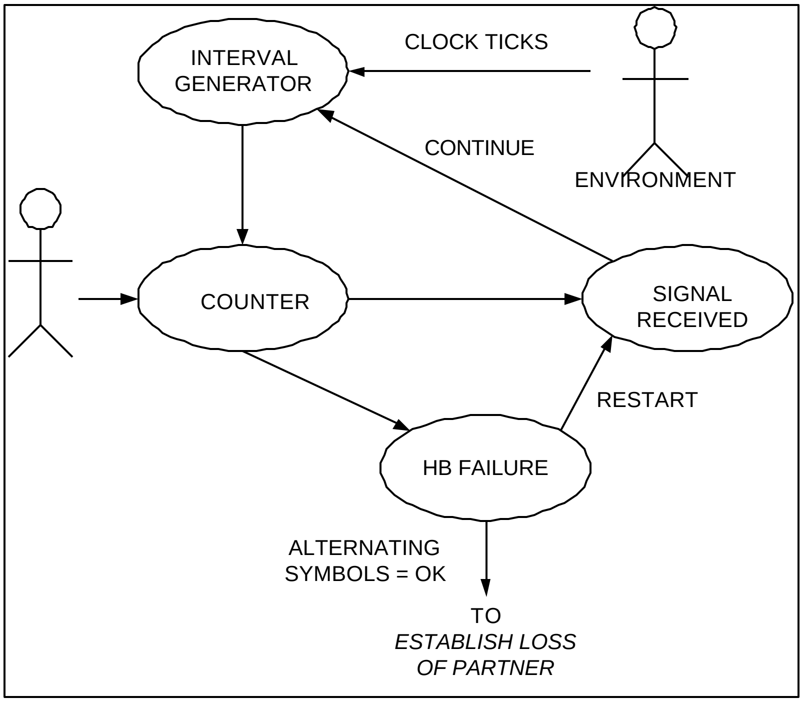


Figure 3 RECEIVE HB USE CASE DIAGRAM

A valid heartbeat consists of three evenly spaced pulses over a defined interval. The *Counter* method transmits the number of pulses received to the *HB Failure* and the *Signal Received* methods. If the number is three, a new interval is started by *Signal Received*. If it is not three, *HB Failure* sends Restart to *Signal Received* (permits restarting an interval) and it also stops sending alternating symbols (thus declaring failure) to the *Establish Loss of*

Partner method at the higher level (see Figure 2). The latter method waits three HB cycles before initiating actions appropriate to loss of partner. Thus, the failure modes that prevent recognition of a loss of partner are (i) spurious generation of exactly three pulses per interval in *Counter* and (ii) spurious transmission of defined alternating symbols by *HB Failure*. Both of these conditions will have to exist for at least three HB cycles to affect the actions at the *Active/Standby* level.

An FMEA worksheet for the *Receive HB* method based on Figure 3 is shown in Figure 4. The level of detail is compatible with a parts approach for hardware portions of the system.

ID	Item/Function	Failure Mode & Causes	Local Fail. Effect	Failure Detection	Compen-sation	Seve-rity
1.1.1.1	Interval Generator	No interval started. Loss of clock ticks or internal failure	HB failure	External	Note 1	IV
1.1.1.2	Interval Generator	Long interval. Missing clock ticks or internal failure.	HB failure	External	Note 1	IV
1.1.2.1	3-Pulse Counter	No count. External or internal failure	HB failure	External	Note 1	IV
1.1.2.2	3-Pulse Counter	Spurious count ?3 per interval. Internal failure	HB failure	External	Note 1	IV
1.1.2.3	3-Pulse Counter	Spurious count, exactly 3 per interval. Internal failure	Spurious HB	External	Note 1	II
1.1.3.1	HB Failure	Does not send Restart. Internal Failure	None	External	Note 2	
1.1.3.2	HB Failure	Spurious Restart. Internal Failure	HB Failure	External	Note 1	IV
1.1.3.3	HB Failure	No or random output to *Loss of Partner*. Internal failure	HB Failure	External	Note 1	IV
1.1.3.4	HB Failure	Spurious defined alternating signals	Spurious HB	External	Note 1	II
1.1.4.1	Signal Received	No *Continue* output. External or internal failure.	HB Failure	External	Note 1	IV
1.1.4.2	Signal Received	Spurious *Continue* output. Simultaneous errors in input and Restart processing	None	External	3-pulse counter	None

Note 1: Temporary failure effects are suppressed because the *Loss of Partner* method waits for three intervals to activate.
Note 2: Will cause Severity IV effect under all conditions when count ? 3 and no effect when count = 3.

Figure 4. WORKSHEET FOR *RECEIVE HB* METHOD.

The worksheet reveals both strengths and weaknesses of the *Receive HB* method. The important strength shown is that the Spurious HB failure effect will occur only under highly unlikely circumstances An important weakness is that there is no internal fault detection. The method depends primarily on the Plant Control to establish whether a reported HB Failure resulted from external (actual loss of partner) or internal events.

This example has demonstrated that the use of UML techniques permits generation of a software FMEA that is

- Unique – the items analyzed are defined by the use case diagrams; the analyst is not required or permitted to partition the program into functions.
- Complete – if it is accepted that an item will work if all its methods work, then the type of analysis shown in Figure 4 meets the completeness criterion.
- Meaningful – the failure modes examined are recognizable by system engineers and the presentation is compatible with that used for hardware.

Table 1 acknowledges that the cost of the parts approach to FMEA is higher than that of the functional approach. But the cost can be much reduced by taking advantage of the extensive databases maintained by UML development tools such as Rational Rose™ and Rhapsody™[*]. These tools can be used to

[*] Rational Rose is a Trademark of Rational Software Corp., Rhapsody is a Trademark of iLogix.

generate use case diagrams (like Figures 2 and 3) and will then maintain information on structure and properties in their database from which it can be retrieved by a program for computer aided FMEA.

UML is also applicable to the development of custom ICs, and the methodology for generating an FMEA for these is analogous to that described here for software. For commercial ICs the vendor's test specification may be used as a starting point for determining the functions that are implemented. The analyst must then use judgment in assessing which of these functions are utilized in the application and possibly grouping those that produce similar effects for entry into an FMEA worksheet.

FRIEND OR FOE - REVISITED

In the preceding section we have shown that some of the perceived obstacles to making FMEA meaningful for digital I&C systems can be overcome, and that commercial tools can reduce the effort and the cost. But even a reduced cost is still a cost, and unless FMEA produces a benefit that is greater than the cost it must be regarded as "foe".

Let us then return to the FMEA worksheet of Figure 4. It shows in a systematic manner that Severity II failure effects can occur only when internal failures produce spurious generation of signal patterns that were selected to preclude spurious generation. That these occurrences are highly unlikely can be deduced without recourse to (usually subjective) estimates of failure probability. The usual objective of an FMEA, evaluation of the probability of high severity failures, has thus been achieved.

As mentioned earlier, the value of FMEA to planning for the O&M phase is often overlooked, and O&M costs tend to be the major cost factor in I&C systems. The failure detection column clearly shows the absence of internal diagnostic provisions in the *Receive HB* method. At the least such diagnostics could generate a failure code that identifies *Receive HB* and possibly a method within it as having caused the failure. In the absence of such a diagnostic the analyst will most likely have to review the system log, identify contributors to the failure, and then troubleshoot each of them to determine the real cause. These measures will use up valuable analyst time and may even involve shut-down of the system. Recognition of this condition from the FMEA can avoid these costs by providing internal diagnostics or modifying the failure detection capabilities at the next higher level to isolate failures in the *Receive HB* method. This use of the FMEA provides an important benefit that will be recognized by the management of I&C systems and will truly make FMEA a friend.

REFERENCES

[1] Cunningham, T. J. and Greene, K., "Failure Mode, Effects and Criticality Analysis", *Proceedings of the 1968Annual Symposium on Reliability"*, January 1968, pp. 374 - 384

[2] McCready, K. F. and Conklin, D. E., "Improved Equipment Reliability Through a Comprehensive Electron Tube Surveillance Program", *Proceedings of the First National Symposium on Reliability and Quality Control"*, November 1954, pp. 81 - 94

[3] Automotive Industry Action Group (AIAG), "Potential Failure Modes and Effects Analysis" (technical equivalent of SAE J-1739), 3rd edition, July 2001

[4] Goble, W. M., *Control System Safety Evaluation and Reliability*, 2nd Edition,
Published By ISA, 1998

[5] Department of Defense, “Procedures for Performing a Failure Modes, Effects and Criticality Analysis”, AMSC N3074, 24 Nov 1980

[6] Hall, F. M., Paul, R.A and Snow, W. E., “Hardware/Software FMECA”, *Proc. of the 1983 Reliability and Maintainability Symposium,* Orlando, FL, January 1983, pp. 320-327

[7] Bowles, J. B. and Chi Wan, “ Software Failure Modes and Effects Analysis for a Small Embedded Control System”, *Proc. of the 2001 Reliability and Maintainability Symposium,* Philadelphia PA, January 2001, pp. 1 – 6.

[8] Goddard, P. L. “Software FMEA Techniques”, *Proc. of the 2000 Reliability and Maintainability Symposium,* Los Angeles CA, January 2000, pp. 118 – 122.

[9] Bondavalli, A., Majzik, I. and Mura, I., "Automatic Dependability Analysis for Supporting Design Decisions in UML", *Proc. of 4th IEEE International Symposium on High-Assurance Systems Engineering (HASE’99)*, Washington D.C., Nov., 1999.

[10] Rosenberg, D., *Use Case Driven Object Modeling with UML,* Addison-Wesley, 1999