A Generic Failure Modes and Effects Analysis (FMEA) Approach for Reliability Modeling of Digital Instrumentation and Control (I&C) Systems

Tsong-Lun Chu*, Meng Yue, Gerardo Martinez-Guridi, John Lehner

Brookhaven National Laboratory, Upton, NY, USA

Abstract: In this paper, a systematic failure modes and effects analysis (FMEA) approach is proposed for creating reliability models for digital instrumentation and control systems. The FMEA approach is at the level of detail where data are available or at least potentially available, i.e., at a level of generic components. The proposed FMEA approach envisions a digital system as consisting of modules, each comprising common generic components, such as an analog/digital converter or a multiplexer. The failure modes of a generic component are defined in terms of their impact on the signal(s) carried by the component, and used to evaluate their impact on the module's input and output signals based on the component's interconnection in the module, that, in turn, determines the status of the entire system. This approach was applied to a digital feedwater control system (DFWCS), consisting of several modules that perform different functions. An automated FMEA tool was created based on the source code of the software and used to propagate failures through the system to determine the system status. The proposed approach is considered a generic one that can support the reliability modeling of any digital system, and can provide a practical solution to addressing the complexity of digital systems with the aid from the automated tool. It should be noted that the implementation of the FMEA approach described in this paper did not involve detailed analysis of software; instead, two generic software failure modes were included as placeholders in the DFWCS example.

Keywords: FMEA, Digital I&C Systems, Automated FMEA Tool.

1. INTRODUCTION

Failure modes and effects analysis (FMEA) is a method used to identify the failure modes of a system and their effects or consequences upon it. One use of FMEAs is to support the development of system reliability models. FMEAs are usually conducted at different levels of detail, with the highest level of detail being the entire system. The system can be further decomposed into different sub-levels (e.g., subsystems or components). An FMEA performed at lower levels (e.g., subsystem level) can be propagated through the higher level(s) up to the entire system because the failure effects of one level indicate the failure modes at the level immediately above it. The process of decomposition may continue until the available information cannot support a more detailed analysis, or the objectives of performing the FMEA are met. While these discussions apply to all system FMEAs, two important generic issues exist with digital-system FMEAs: (1) There is no well-established definition of the failure modes of, and their effects on, digital systems; and (2) there is no specific guidance on how to undertake FMEAs for such systems. Despite these existing issues several reliability studies of digital systems have been completed, e.g., the studies reviewed in NUREG/CR-6962 [1]. In general, those studies were not conducted with sufficient detail for the approach described here; i.e., the failure modes of a component either were not explicitly defined or often were implied simply as "failures to perform its dedicated function," so that the only identified effect of failure on the system is that the system has failed.

* Email address: chu@bnl.gov.

Current digital systems consist of many components and are highly complicated. Theoretically, all the relevant interactions between the components of a digital system should be captured by its reliability model. In practice, these interactions are hard to capture in the model unless a sufficiently detailed FMEA is performed.

In nuclear power plants, digital systems mainly are employed to control specific equipment or a process, or perform safety-related functions, such as tripping the reactor or actuating an emergency safety feature. The differences in desired functions and the uniqueness of individual industrial processes require specific digital system design features. This implies that the data for a specific digital system of interest is likely to be very scarce, owing to the uniqueness of the system designs employed at nuclear power plants. The collecting of suitable data or system-specific data is exacerbated by the fact that digital systems are likely to be upgraded frequently.

In this paper, a systematic FMEA approach is proposed for creating reliability models for digital instrumentation and control (I&C) systems. The FMEA approach is at the level of detail where data are available, e.g., in the PRISM database developed by the Reliability Analysis Center [2], or at least potentially available, e.g., data collected by component manufacturers. Furthermore, the proposed FMEA approach considers that different failure modes of the same component may have different impacts on the system.

Although the designs and complexity of digital systems can greatly differ from each other, they have basic similarities. A digital system is envisioned as consisting of modules, each comprising common generic components, such as an analog/digital (A/D) converter, a multiplexer (MUX), a microprocessor and its associated components (e.g., random access memory [RAM] and buses), a demultiplexer (DEMUX), and a digital/analog (D/A) converter. There is not a standard list of failure modes for digital components, but, in general, the failure modes of a generic component can be defined in terms of their impact on the signal(s) carried by the component. Therefore, a consistent set of failure modes can be applied to components of the same type, even if they are of different makes or models. Failure modes are postulated at the level of generic components, and their effects propagated up through modules to the overall system. The proposed FMEA approach is applied to a digital feedwater control system (DFWCS), consisting of several modules that perform different functions. The main central processing unit (CPU) module[†] of the system is used as an example to illustrate the implementation of the FMEA. It should be noted that the implementation of the FMEA approach described in this paper did not involve detailed analysis of software; instead, two generic software failure modes were included as placeholders in the DFWCS example.

Section 2 of this paper describes the generic approach to performing digital system FMEAs. A description and a simplified diagram of the DFWCS are presented in Section 3. The application and implementation of the proposed generic FMEA approach and the major assumptions are mainly discussed in Section 4. Example FMEA analyses are presented to illustrate how the approach can be applied to a digital system using the main CPU module of the DFWCS as an example. Section 5 provides additional considerations when implementing the FMEA approach and discusses an automated tool for supporting the FMEA. Concluding remarks are provided in Section 6.

2. A GENERIC APPROACH TO FMEAS OF DIGITAL SYSTEMS

Defining component failure modes in terms of the function performed by the component allows them to be used in reliability analysis as long as the associated failure data are available. It is thus preferable to carry out an FMEA at the level of generic components where reliability failure parameters are available for the failure modes of the functions of concern. Also, the software and

[†] In general, CPU represents a central processing unit, which is a generic component of digital systems. Here, a CPU module includes a CPU and its associated components, such as a multiplexer, analog/digital converter, etc. In this study, a CPU of a CPU module is denoted as a microprocessor in order to avoid confusion with a CPU module, and CPU and CPU module are used interchangeably to represent a CPU module.

hardware interaction, the fault-tolerance characterized by specific software design, and the interactions between digital systems and monitored/controlled processes are reflected by signals transmitted between generic components of the digital system; therefore, all three of these effects potentially can be captured by the FMEA at this level.

Accordingly, a generic approach to undertaking FMEAs of digital systems [3] is proposed here. The entire digital system is decomposed into successive levels of detail, starting at the overall system level and continuing until the level of the generic components is reached. The number of intermediate levels depends on the complexity of, and information available about, the particular system. The actual performing of the FMEA, begins at the lowest level (in this case, at the level of generic components), where failure modes are postulated and their effects propagated back up to higher levels until the impacts on the entire system can be determined.

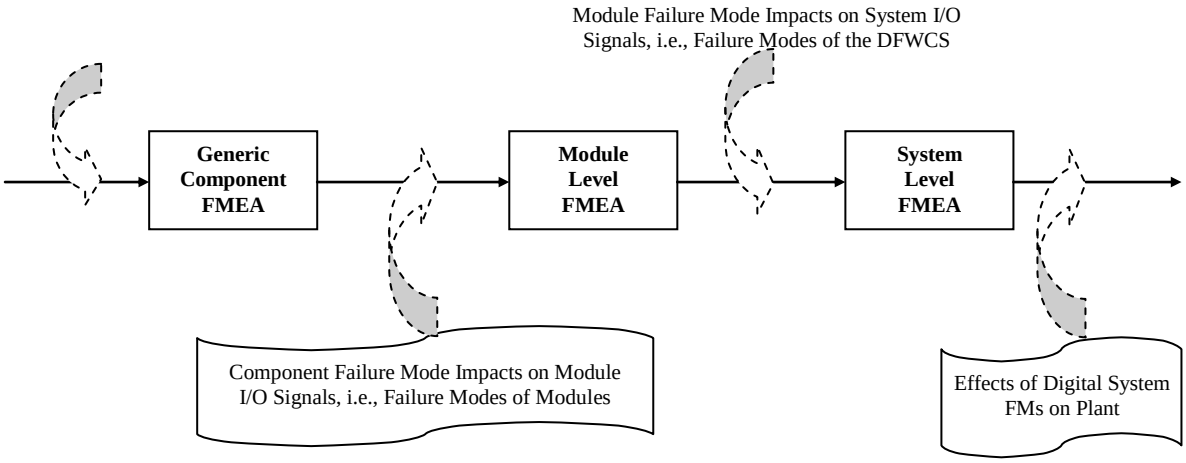

Figure 1: Steps for the Generic Approach to Digital System FMEA

Figure 1 illustrates the proposed FMEA process. The key points in performing FMEAs are that (1) the status of the system eventually is determined by module signals that reflect the interactions between the modules and between the digital system and the plant and (2) these signals are directly affected by the status of the generic components of the modules. Figure 1 depicts a pathway showing how the failure modes of generic components in a specific digital system are used to determine the status of the module and the whole system. That is, the failure modes of generic components are used to evaluate their impacts on the module input and output (I/O) signals, which in turn determine the status of the entire system, i.e., whether or not a system failure takes place.

3. A DESCRIPTION OF THE DFWCS

A DFWCS of a secondary loop of a pressurized water reactor (PWR) is used as an example system. The DFWCS analyzed is described in detail in [1, 3]. Here, a summary description of the system is provided.

The DFWCS consists of sensors, transmitters, two central processing unit (CPU) modules (the main and backup CPUs), four controller modules (one each for the main feedwater valve [MFV], bypass feedwater valve [BFV], feedwater pump [FWP], and pressure differential indication [PDI]), and associated support systems, i.e., direct current (DC) power supplies and 120v alternating current (AC) buses. The DFWCS sends demand signals to the positioners of the main feedwater-regulating valve (MFRV) and the bypass feedwater-regulating valve (BFRV), and to the turbine controller of the main feedwater pump (MFP). The positioners convert electrical signals into pneumatic pressure that is used to position valves. The PDI controller that normally displays the differential pressure across the MFRV also can serve as a manual control station for the MFRV and BFRV. The digital parts of the system are the CPU modules and controller modules. Each module consists of a microprocessor and its associated components, e.g., A/D converter, MUX, and D/A converter. In addition, each of the

CPU modules contains an external watchdog timer (WDT). In Figure 2, a simplified diagram of the system shows the modules and components considered in the reliability model of the DFWCS and the major signals between them. The solid boxes represent modules and components that are modeled in detail, while the dotted boxes represent those that are either modeled in a simplistic way or not modeled at all because they are beyond the scope of this study or found not to affect the operation of the system at full power.

The system has two modes of operation, automatic and manual. This study assumes that the system is initially operating in automatic mode. The operators can interact with the system by using the controllers that are located in the main control room. If a controller switches from automatic to manual control mode due to a detected failure condition, the operators then can take manual control. The DFWCS also operates in either high-power or low-power mode. Since the plant is assumed to be operating at full power for this study, the system is considered to be initially operating in the high-power mode.

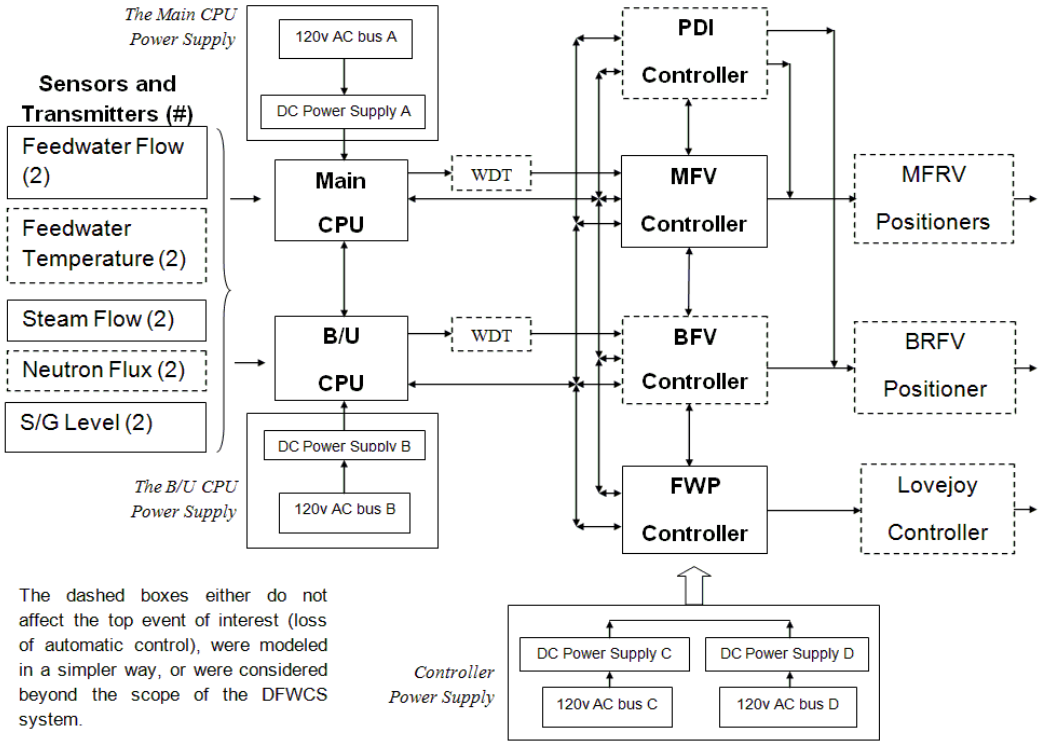

Figure 2: Modules of the DFWCS

4. APPLICATION OF THE GENERIC APPROACH TO FMEAS OF THE DFWCS

4.1. Hierarchical Consideration of the DFWCS FMEAs

The DFWCS consists of: (1) the main and backup CPUs that essentially execute identical software and use the same control algorithms to calculate control demands based on input from the plant and (2) four controllers that receive the corresponding demand signals from the CPUs and forward them to valve positioners or the pump speed controller. The interactions of the two CPU modules and four controller modules are determined from system design information. Each DFWCS module can be considered as a complete digital system with its own A/D input, processing, and D/A output. Therefore, the major components of all modules include A/D converters, D/A converters, microprocessors and their associated peripheral devices (e.g., RAM, read-only memory [ROM], and buses), MUXs, DEMUXs, and some analog I/O devices (e.g., current loop [CL] devices). The major components of the modules are identified based on general architecture of digital systems.

Effectively, the DFWCS is broken down into three levels. The highest level is the entire DFWCS system, and the lowest level corresponds to the major (generic) digital components. The single intermediate level (the module level) includes the six modules. The failure modes at the system level are the failure effects of the modules; similarly, the failure modes at the module level are the failure effects of the components within the module. It should be noted that system failure modes usually are defined in terms of the system's functionalities and, thus, are system specific. The FMEA's scope encompasses the internal failures of the system, but excludes external events, such as fire or seismic events.

- System-Level FMEA: For the system-level (top-level) FMEA, the scope of analysis included the entire DFWCS. A loss of automatic control of the DFWCS, given that the plant is in full power operation, is the top event selected for this study.

- Module-Level FMEA: The next level of the FMEA included the major modules of the DFWCS, i.e., the main CPU, backup CPU, MFV controller, BFV controller, FWP controller, pressure differential indicating (PDI) controller, and some related dependencies, such as power supply and sensors. The failure modes of these major modules are represented by the failures of their individual I/O signals (see more discussion in Section 4.3); their impacts on the behavior of the modules were analyzed.

- Major-Component-of-Module-Level FMEA: The lowest level FMEA analyzed the components inside the modules of the DFWCS. The controllers are application-specific integrated circuit (ASIC)-based devices. Since the major components of both controllers and CPUs are similar, they are analyzed in the same way.

Note, the module-level FMEA is not of significant interest since the goal is to determine the system status given the component failures.

4.2. Assumptions for the DFWCS FMEA

In this example implementation of the FMEA approach, steady-state operation of the system is assumed as the initial condition, and loss of automatic control is the system failure condition being modeled. Therefore, conditions arising that result in an operator switching from automatic mode to manual mode are also considered system failures. The following are important assumptions made in performing the FMEAs for the DFWCS:

- All components, including those playing a standby role, e.g., the backup CPU, are operating at all times and can fail at any time.

- Typically, a component can have more than one failure mode with different effects that must be modeled differently. A component is assumed to fail only once in a given failure sequence, i.e., after one failure mode of the component has occurred, other modes cannot occur for the same component (e.g., if an A/D converter experiences the failure mode "all bits stuck at zeros," it will not subsequently experience the failure mode "random bit failure" in the same failure sequence). This assumption is believed to hold for most of the digital components, because available information on digital component failures seems to suggest so, i.e., the hardware failure databases reviewed in [1] did not provide any indication that additional failures may occur subsequent to an initial failure. It would be unrealistic to assume that a component can always fail more than once. It may be possible that a certain component fails to an intermediate failure mode before it reaches one of the other failure modes. If recognized, such a sequence of failures can still be analyzed and modeled using the approach of this study as discussed in [3].

- Due to lack of detailed design information, failures of different components are assumed to be independent of each other (regardless of how they are physically wired together). It is recognized

[1] that determining the effects of component failure modes in a real digital system could be much more complex than what is assumed here. For example, the detailed connection of a digital output to a few digital inputs determines if failure of one input would affect other inputs, which could lead to cascading component failures. On the other hand, built-in mechanisms that may detect and isolate the cascading faults can also be designed, and would need to be accounted for in the FMEAs. The independence assumption is introduced because, otherwise, detailed analyses of the designs at the circuit level, which are unavailable for this example implementation, must be performed for individual components to determine how a specific failure of a component affects the connected components.

- It is assumed that a drifting signal will eventually drift to out-of-range (OOR) high or low. As a result, in the model a drifting signal is always detected by the OOR check, and the system may continue to successfully operate, e.g., using the redundant signal (i.e., the signal that does not drift OOR). This treatment may be non-conservative because, in reality, a drifting signal may not drift OOR, and may cause an undetectable failure that could result in system failure. However, according to the design information of the DFWCS, if a drifting signal is not detected by the OOR check, it still may be detected by the built-in deviation check of the application software. Ideally, for a control system, a thermal-hydraulic model of the plant can be used to capture the effects of a drifting signal. If such a failure mode causes a system failure, the failure mode can be modeled accordingly. Ref. [3] provides more discussion on the significance of the assumption used here regarding signals drifting OOR and on how to better model drifting signals.

4.3. Example FMEA for the DFWCS: Main CPU Module

This section provides detailed discussions on how failure modes are defined, and how they are propagated to the system level. It is anticipated that FMEAs can be carried out for other digital systems using a similar process to that used for the examples covered in this section. Figure 3 shows the "internal" components of the main CPU module of the DFWCS, i.e., the components connected to the main CPU, and considered in the reliability model as its internal parts.

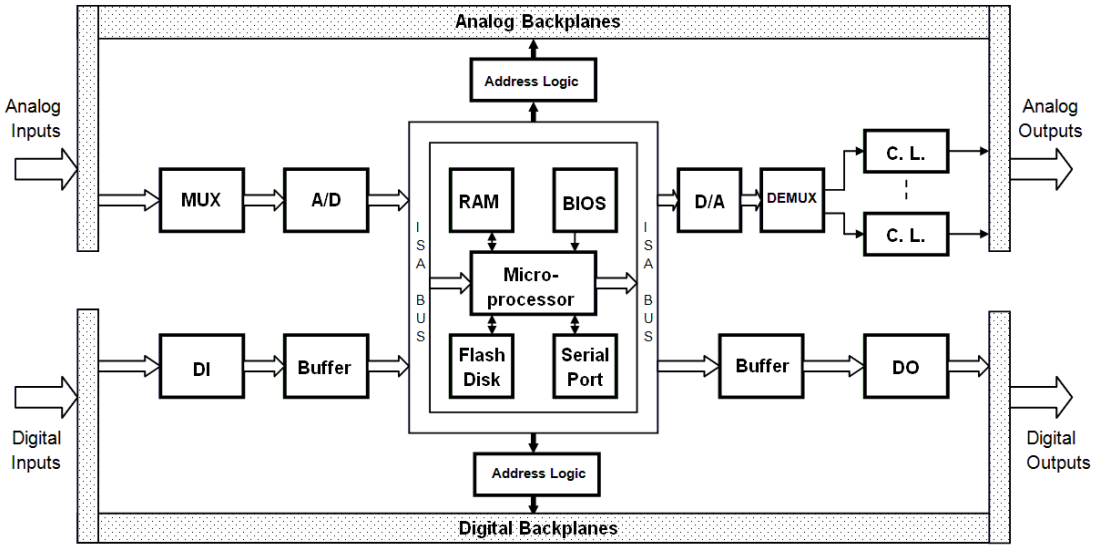

Figure 3: Major Components of the Main CPU Module

In the diagram, analog backplanes and digital backplanes are buses that interface with all I/O of the main CPU module. An Industry Standard Architecture (ISA) bus is used for the microprocessor of the main CPU module to interact with components connected to the backplanes. A current loop device produces a current output (usually 0-20 milliamperes [mA]). Each analog output is assumed to use one current loop. The figure does not depict the current loops for analog input signals, but it also is assumed that each analog input uses a current loop. "DI" and "DO" indicate the digital input and

output modules, respectively. Other components are all standard in digital systems. The arrows represent signal flows between different components.

The failure modes for individual components of the main CPU module are summarized below, and the sources of the failure modes are cited. It is noted that the failure modes of components discussed in the references cited below may not perfectly match the component failure modes that were used in the example implementation of the FMEA approach because, in some cases, the failure mode from a reference did not clearly specify the impact on the signals associated with the component and the impact had to be assumed (e.g., "no output" and short-circuit" were assumed as "all bits stuck at zero"). Nevertheless, they were the best approximation found at present.

1. Hardware Common-Cause Failure (CCF): The hardware of the main CPU and backup CPU is identical. The occurrence of a hardware CCF may fail the entire system.

2. Software: The main and backup CPUs run the same software and a software CCF may occur and fail the entire system. Two CCF failure modes are considered: (1) the software on the CPU modules seems to be running normally but sends erroneous output and (2) the software halts and, hence, the CPU modules stop updating output. In addition to the CCFs of software, the above failure modes are also considered for the individual software running on the CPU modules considering the fact that the main and the backup CPU are in different modes (controlling and tracking modes) and might be running different portions of the software at any given time. More information on the completeness of software failure modes is provided in [3].

3. Microprocessor of the Main CPU: Failure modes considered are (1) the microprocessor seems to be running normally but sends erroneous output and (2) the microprocessor stops updating output [4].

4. Associated Components of a Microprocessor, such as the ISA bus, RAM, ROM, BIOS (basic input/output system), flash disk, buffer, and serial port: It is conservatively assumed that each component has only one failure mode, i.e., a loss of the component, which entails the loss of the functions performed by the component.

5. Address Logic: This is a generic digital component, also called a decoder. A microprocessor uses the address logic to access the information transmitted on the backplanes. The failure mode is assumed as a loss of the address logic, so that the microprocessor cannot access the intended information.

6. Voltage Input Module: The voltage regulators are assumed to be the major component of the voltage input module of the main CPU. The failure modes are fail-high and fail-low of the associated voltage input signal [4].

7. MUX and DEMUX: Failure modes of MUXs and DEMUXs are defined in [5] in terms of the analog signals they process, which include a loss of one or all signals. No other failure modes of MUXs or DEMUXs were mentioned in [5], and, therefore, a loss of signal is modeled here as signal fails low.

8. A/D and D/A converters: Both A/D and D/A converters are linear integrated circuits (ICs), i.e., the I/Os are proportional to each other; all analog I/Os of the same module share them. The failure modes of an A/D converter include all bits of the A/D stuck at zeros, all bits stuck at ones, and a random bit-failure of the A/D converter [6]. The failure modes of a D/A converter include output fails (drifts) high or low [6]. It is assumed that if the D/A converter output starts drifting, it will eventually reach the high or low detection threshold.

9. Current I/O Modules: The major components of the current I/O modules are current loops that essentially are linear transmitters/receivers. They also are linear ICs and their failure modes are

current signal fails (drifts) high or low [6]. It is assumed that if the current starts drifting, it will eventually reach the high or low detection threshold.

10. Digital I/O Modules: Digital I/O is implemented via a solid-state switch [7]. The status of a digital signal is controlled by opening or closing the switch. The solid-state switch may fail to operate (fail as is) and spuriously operate (fails to the opposite state), as stated in [4].

In summary, failure modes of components that carry analog signals include "signal fails high" and "signal fails low" (a loss of signal is modeled as signal fails low, as indicated above). Failures of drifting analog signals, such as random signals, are assumed to either drift high or drift low, i.e., the same as fail high or fail low. Failure modes of components that carry digital signals include normally closed, fails closed (NCFC), normally closed, fails open (NCFO), normally open, fails closed (NOFC), and normally open, fails open (NOFO). These failure modes will cause the corresponding digital I/O signals of a module to fail to operate (NCFC and NOFO) or fail to the opposite state (NCFO and NOFC).

Based on the physical meaning of the failure modes of a specific component, their impact on signals associated with this component can be determined, as described above. The faulted signal(s) is processed by the software running on individual modules and propagated between different modules that are interconnected (see Figure 2). The impacts of the postulated component failures on the modules and the system are represented by the values of the signals the modules and system process, and therefore, the interactions between the components modeled can be captured. If the failure effects of a component failure mode are unknown due to a lack of knowledge about that failure mode, it is conservatively assumed that the associated module is failed. Thus, the effects of component failure on the main CPU module and the entire system can be evaluated. Note, the above FMEA analyses heavily rely on the application software of individual modules, and can be potentially performed by using an automated FMEA tool given the availability of sufficient software details, e.g., the source code, as will be further illustrated in Section 5.

Table 1 lists the representative failure modes of a limited set of generic component types and their potential impact on the main CPU module and the DFWCS (a more comprehensive listing of failure modes can be found in [3]). The impacts of single component failures on the main CPU module and the system were determined by the FMEA performed manually and validated with the automated FMEA tool. Impacts of some of the failure modes were postulated based on understanding of the function and design of the components, e.g., a loss of BIOS is assumed to be an undetectable failure that will fail the system. The table does not provide a complete FMEA of the main CPU module; instead, there is an explanation of the meaning of the failure modes of generic component types, and an illustration of the way these failure modes propagate to the entire system via the intermediate module level. Details about the FMEAs of the DFWCS modules and the system are provided in [3].

It is important to consider fault-tolerance features in each CPU module. If the main CPU module fails and is detected by these features, the backup CPU module will assume control of the system. This process is named a "failover to the backup CPU," or simply a "failover." Each CPU module (main and backup) has available two types of fault-tolerance features. The first one is failure-mode detection by the application software running on the CPU, and the second one is monitoring by an external watchdog timer (WDT). Each feature can initiate a failover from the "controlling" CPU module (normally the main) to the "tracking" CPU module (normally the backup).

Column 1 in Table 1 presents the failure modes for individual components (including software[‡]) in the main CPU module. Column 2 (heading "Failure Mode Detected by") indicates whether the failure modes can be detected by the application software or the external WDT, which represent fault-tolerance features of the main CPU. The impacts of the failure modes on the main CPU module are

[‡] This example implementation of the FMEA approach did not involve detailed analysis of software; instead, two generic software failure modes were included as placeholders in the DFWCS example.

indicated in Column 3. Column 4 establishes whether a failure mode triggers system failure. Note that failures of different signals carried by a particular type of component, e.g., current loop, may have different impacts on the main CPU or the DFWCS system, i.e., the impact is signal dependent. The actual failure effects for each specific component are provided in the FMEA tables of [3]. Finally, Column 5 provides explanations on each failure mode.

Considering the failure modes of the main CPU, an undetectable failure will result in failure of the DFWCS (i.e., loss of automatic control) because the main CPU is assumed to be the controlling CPU. A failure mode detectable by the application software indicates that the main CPU can detect the failure, so that the application software initiates a failover to the backup CPU, if needed. A WDT detectable failure signifies the detection of the failure mode by the external WDT of the main CPU, and the resulting failover to the backup CPU. For a failure mode of a component that does not impact the main CPU, e.g., a loss of serial port, the main CPU will continue to carry out the DFWCS control function. If a failure mode does not cause the main CPU module to fail, this module will continue to operate with a latent failure present, i.e., a failure that may subsequently lead to failure of the DFWCS if combined with other component failures.

Table 1: Illustrative Example of Performing FMEA at Component Level of the Main CPU Module

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
The microprocessor seems to be running normally but sends erroneous output	No	No	Undetectable Failure	Yes	This is considered an undetectable failure of the main CPU and will fail the entire system.
The microprocessor stops updating output	No	Yes	WDT Detectable Failure	No	When the WDT no longer receives a toggling signal, it will cause a failover of the main CPU to the backup CPU provided that the status of the WDT is normal.
Loss of BIOS	No	No	Undetectable Failure	Yes	The input and output operations of the CPU rely on BIOS routines. However, it is unknown whether a loss of BIOS will cause a complete loss (or a partial loss) of inputs to and outputs from the application software and CPU; hence, the failure is conservatively assumed to be undetectable.
Fail (drift) high or fail (drift) low of current loop device	Signal dependent	No	Signal Dependent	Signal dependent	The current loop is a linear device that may fail high or low, resulting in the associated I/O signal failing high or low. Fail low includes failures of fail to zero. The main CPU processes different signals differently.
All 16 bits of A/D converter stuck at zeros or ones	Yes	No	Application Software Detectable Failure	No	1. Both A/D and D/A converters are linear ICs. The A/D converter is shared by all analog inputs, and its loss will entail the loss of all analog inputs. 2. Stuck at zeros or ones indicates that all analog signals fail low or high. The main CPU software can detect failures of some input signals, and then cause a failover.

Failure Mode	Failure Mode Detected by		Failure Effects on Main CPU	Fails the DFWCS?	Comments
	Application Software	WDT			
Random bit failure of A/D converter	No	No	Undetectable Failure	Yes	Although the main CPU software can detect some random failures, they are conservatively assumed to be undetectable and will fail the whole system.

Following the procedures and methods illustrated in this section using the Main CPU module as the example, one should be able to perform the complete FMEA analyses on other modules and components of the DFWCS and the whole system. Similar to the main CPU, if a component failure causes an undetectable failure of any of the controllers, the entire DFWCS will fail. Note, an undetectable failure of the backup CPU does not directly lead to loss of automatic control of the system (i.e., DFWCS failure), but the automatic control will be lost if there is a need for failover from the main CPU to the backup CPU.

5. ADDITIONAL CONSIDERATIONS AND THE AUTOMATED FMEA TOOL

Due to the flexibility, variety, and complexity of digital systems, the difficulties in performing FMEAs at the proposed levels also are obvious. The previous description of the FMEA process requires a thorough knowledge of digital systems and their associated components, as well as specific design information on the particular digital system to be analyzed. Moreover, it is prohibitively difficult in manually relating different pieces of information to determine the effects of individual postulated failures and their sequences (i.e., combinations of ordered failures). While it is not a straightforward task to gain a detailed understanding of underlying principles of digital systems/components, i.e., the principles of generic digital components and physical meanings of their failure modes and their potential effects, the more difficult part in the analysis is acquiring and using design information of the specific digital system. The design information is system specific, and must be collected and reviewed extensively to undertake the FMEAs of the system. The FMEAs of the DFWCS were mainly accomplished manually and a significant effort was expended in doing so. The system designers certainly will have the necessary design information but may not perform the detailed analysis performed in this example.

The FMEAs for the DFWCS identify the component failure modes that individually result in (or are assumed to result in) system failure. However, the sum of the failure probabilities for these failure modes only represents a part of the overall DFWCS failure probability. A component failure that does not cause the system to fail is called a latent failure. There are several latent failures for each DFWCS module. Since a latent failure does not, by itself, trigger system failure, the impact of combinations of latent failures on the system must be evaluated, and the number of the combinations of failures may be extremely large. Accordingly, an automated FMEA tool was developed to provide an efficient way to resolve this problem by making use of the system's design details to examine its responses to combinations of postulated failures.

The automated FMEA tool was developed from the original source code of the CPUs, and from re-creating the controller software interfaced by input- and output-variables representing the physical connection signals between the modules, the system, and the controlled process. The tool was employed to crosscheck the results of the manual FMEA for individual failure modes of individual components, and to assess the impacts of higher order failure sequences, i.e., double- and triple-failure combinations.

Use of the automated FMEA tool revealed that the same set of failure modes, but with a different order of occurrence, may have a different impact on the system. As mentioned previously, one of the fault-tolerant features of the example DFWCS is that when it detects a failure associated with the main

(or controlling) CPU, it will automatically switch control over to the backup (tracking) CPU. As an example, consider a double sequence consisting of two failures, fail out-of-range high of one feedwater flow analog input to the main CPU, and all-bits stuck at 1 of the A/D converter of the backup CPU. Neither one of these two failures by itself will cause the system to fail. If the failure of the backup CPU occurs after the failure of the input signal to the main CPU, the system will fail because the input signal failure will cause a failover to the backup CPU and this, in turn, will be failed by its A/D converter failure. However, reversing the order of this double sequence, the backup CPU will be failed first and the response of the main CPU to the subsequent feedwater flow input signal failure would be to use the other feedwater flow input as opposed to failing over to the backup CPU, because the main CPU knows the failure status of the backup CPU. Use of the automated FMEA tool identified 510 double sequences of this type for the DFWCS model.

See Refs. [3,8] for more details on the automated FMEA tool.

6. CONCLUDING REMARKS

The FMEAs of the DFWCS modules and other DFWCS components afford several observations: (1) many failure modes of components of modules will not fail the system; (2) the impacts of different failure modes for a specific component may be very different from each other; (3) the failure impacts of the same failure modes of the same components on different modules can be significantly different (not shown in this paper but can be found in [3]); and (4) fault-tolerance features implemented via specifically designed hardware (e.g., an external WDT) or hardware redundancy (e.g., the main CPU and the backup CPU), or application software, play a vital role in determining the effect of each component failure mode on its respective module and on the entire system.

The proposed FMEA approach and its implementation make the following simplifying assumptions: (1) drifted analog signals are assumed to eventually drift high or low and can be merged with the failure modes of signal fails high or low, and (2) only one failure mode (i.e., loss of component) is assumed for some components, such as the ISA bus, RAM, ROM, BIOS, flash disk, serial port, address logic, and buffer. Furthermore, for the latter assumption, in most cases, the failure impact on the module from loss of the component was considered as an undetected failure due to difficulty in precisely evaluating the impacts. For example, some of the lower level failure modes of memory may be detectable, while some other failure modes are not. This is an issue that can be addressed using the concept of coverage. More detailed modeling, such as through the use of fault injection analysis, as discussed in the next paragraph, is needed to determine if lower level faults can or cannot be detected. While a more systematic treatment of the detectability of component failure modes is desirable, it should also be recognized that detectability of a failure mode is design specific and coverage values obtained for one system will often not be applicable to other systems.

Other assumptions made for this example implementation include: (1) a component can only fail to one of its failure modes and (2) failures of different components are independent of each other whether or not these components are physically wired together, i.e., individual failures are localized. For example, a failure of component A can be propagated to component B to which component A is connected, but this does not introduce a new failure of component B. The former assumption probably can be relaxed by reviewing failure experience and modeling the physics of failure of the components (i.e., considering root causes of failure, such as fatigue and fracture, to study the physical processes that bring about failures), an up-front approach adopted in many countries [9]. The latter assumption is due to lack of design details. If the design details are available, then the assumption may not be necessary because whether or not a failure is localized can be determined manually or by performing supporting analyses using tools, such as fault injection methods. Ref. [10] discusses use of a fault-injection method to study the dependability of a digital system by modeling its internal logic in detail, and applied the method to estimate the coverage of the main CPU. This method might be useful for refining the FMEAs described here. Using the detailed model of a digital system/component considered in a fault injection method, the effects of non-localized failures can be accounted for. The completeness of the failure modes also is an issue. Clearly, the role that failure modes and the

associated data play in studies such as this is vital. There are very few public references that describe failure modes of generic digital components and the associated distributions of failure modes (mainly [4] and [6]). Refined definitions of failure modes of digital components and associated data are desirable, and further efforts in this area are needed.

The proposed FMEA approach is based on the generic architecture of digital systems from which the major components of the system can be identified. Depending on the level of detail required for a reliability analysis, the list of the major components may be expanded, and the failure analysis can be performed in the same manner. Therefore, the proposed approach is considered a generic one that can support the reliability modeling of any digital system. Implementing the proposed FMEA approach using the automated tool is especially important for digital systems as it offers a practical solution to addressing the complexity of these systems.

References

[1] T. L. Chu, G. Martinez-Guridi, M. Yue, J. Lehner, and P. Samanta, "Traditional Probabilistic Risk Assessment Methods for Digital Systems," NUREG/CR-6962, July 2008.

[2] Reliability Analysis Center, "Electronic Parts Reliability Data," EPRD-97, 1997.

[3] T. L. Chu, M. Yue, G. Martinez-Guridi, and J. Lehner, "Modeling Digital Feedwater Control System Using Traditional Probabilistic Risk Assessment Methods," NUREG/CR-6997, August 2009.

[4] Reliability Analysis Center, "Failure Mode/Mechanism Distributions," Department of Defense Information Analysis Center, FMD-97, December 1997.

[5] Aeroflex, "Reliability Failure Mode Effects and Predicted Failure Rate Analysis for the ACT8500 64-Channel Multiplexer Module," Application Note AN8500-1, September 15, 2005.

[6] V. Meeldijk, *Electronic Components Selection and Application Guidelines*, John Wiley & Sons, 1996.

[7] Eurotherm Ltd., Using 2604/2704 Fixed Digital I/O, Technical Information, No. TIN 137, pg. B-113, 2000.

[8] M. Yue, T. L. Chu, G. Martinez, and J. Lehner, "An Automated Tool for Supporting FMEAs of Digital Systems," *ANS Probabilistic Safety Assessment and Analysis*, September, 2008.

[9] M.G. Pecht, and F. R. Nash, "Predicting the Reliability of Electronic Equipment," *Proceedings of the Institute of Electrical and Electronics Engineers*, Vol. 82, No. 7, July 1994.

[10] C. Elks, et al., "Quantitative Dependability Assessment of the Benchmark Digital Feed-Water Control System: Final Report," University of Virginia, Technical Report UVA-CSCS 2007-003, January 20, 2008.