A Summary of Taxonomies of Digital System Failure Modes Provided by the DigRel Task Group[1]

Tsong-Lun Chu[a*], Meng Yue[a], Wietske Postma[b]
[a]Brookhaven National Laboratory, Upton, U.S.A.
[b]NRG, Arnhem, The Netherlands

Abstract: Recently, the CSNI directed WGRisk to set up a task group called DIGREL to initiate a new task on developing a taxonomy of failure modes of digital components for the purposes of PSA. It is an important step towards standardized digital I&C reliability assessment techniques for PSA. The objective of this paper is to provide a comparison of the failure mode taxonomies provided by the participants. The failure modes are classified in terms of their levels of detail. Software and hardware failure modes are discussed separately.

Keywords: Digital I&C, failure modes, PSA, DigRel

1. INTRODUCTION

In its June 2007 meeting, the Committee on the Safety of Nuclear Installations (CSNI) of the Nuclear Energy Agency (NEA) of the Organisation for Economic Co-operation and Development (OECD) directed the Working Group on Risk Assessment (WGRisk) to set up a task group (TG) called DIGREL to coordinate an activity on digital instrumentation and control (I&C) system risk. The focus of this WGRisk activity is on current experiences with reliability modeling and quantification of these systems in the context of probabilistic safety assessments (PSAs) of nuclear power plants (NPPs). During October 21-24, 2009, a technical meeting was held in Paris, France to discuss such experiences. The objectives of this technical meeting were to make recommendations regarding current methods and information sources used for quantitative evaluation of the reliability of digital I&C systems for NPP PSAs, and identify, where appropriate, the near- and long-term developments that would be needed to improve modeling and evaluating the reliability of these systems [1]. One of the recommendations was to develop a taxonomy of failure modes of digital components for the purpose of PSA.

Recently, the DIGREL TG initiated a new task on developing this taxonomy. It is an important step towards standardized digital I&C reliability assessment techniques for PSA. The key approach toward developing the taxonomy is holding a few workshops. The first workshop took place on May 16-19 2011 in Bethesda, Maryland, U.S.A. During the workshop, participants presented information on their taxonomies work and ideas on the development of a failure mode taxonomy. The second workshop was held on October 26-28, 2011 in Espoo, Finland. The taxonomy was further developed, and an outline of the guideline report was drafted. The third workshop was held on February 16-17, 2012 in Paris, France. In this workshop, the levels of detail for the hardware failure modes taxonomy were revisited, and a consensus on naming the levels of detail was achieved among the group. Additional workshops are being planned to further develop the taxonomy, apply it to an example system, and develop guidelines on the use of the taxonomy in supporting reliability modeling and data collection.

A total number of ten organizations from different countries provided inputs. The organizations include:

- BNL (Brookhaven National Laboratory),
- CNSC (Canadian Nuclear Safety Commission),

- EDF (Electricity of France),
- IRSN (Institut de Radioprotection et de Surete Nucleaire),
- JNES (Japan Nuclear Energy Safety Organization),
- KAERI (Korean Atomic Energy Research Institute),
- NRG (Nuclear Research and Consultancy Group),
- NKS (Nordic Nuclear Energy Research),
- OSU (Ohio State University),
- VTT (Technical Research Centre of Finland).

In general, the organizations provided input by summarizing their own experience and findings about digital I&C system failures. Also a report from the Oak Ridge National Laboratories has been used as input [2]. In total, twelve failure mode and effect analyses (FMEAs) from nine organizations have been contributed. All organizations were asked to give the definition of the level of detail they considered and to give the failure modes they defined at that specific level of detail. The detailed taxonomies are not shown here. The taxonomies are split in hardware and software taxonomies. Not every organization provided a taxonomy for software failures in addition to the hardware taxonomy.

The objective of this paper is to provide a summary and comparison of the failure mode taxonomies provided by the participants. Many failure modes were provided, mostly for protection systems as opposed to control systems, without indicating the levels of detail at which these failure modes were defined. A meaningful comparison between failure modes is unachievable without a clear definition. Subsequent input from the participants on descriptions and definitions of the levels of detail have been established. With this information, the failure modes are defined and classified in terms of their levels of detail, and software and hardware failure modes are discussed separately.

Section 2.1 presents the description and definition of the levels of detail for hardware. A summary of failure modes from individual participants is shown and discussed in Section 2.2. Similarly, Sections 3.1 and 3.2 provide the definition of the levels of detail and a summary of failure modes for software, respectively. A summary of observations on the inputs and discussions about related issues is included in Section 4. Section 5 presents the conclusions and future work.

2. COLLECTION OF HARDWARE TAXONOMIES

First, a more descriptive explanation of the levels of detail is presented. After the descriptive explanation, the important terms are defined more strictly to support analysis purposes.

2.1. Levels of Detail for Hardware of Digital System

A digital protection system at a NPP, such as a reactor protection system (RPS), consists of redundant divisions that provide inputs to voting modules, which determine if an actuation signal should be generated. The divisions may be of the same or different architectures but all perform the same functions (in general simultaneously but not in a synchronous mode). Each division consists of multiple I&C units, carrying out specific tasks or functions, like acquisition and processing of data, voting or determining priority. Each I&C unit consists of one or more modules (i.e., circuit boards), performing particular function(s) or a portion of a function for the I&C unit, such as input/output, communication, and processing, etc. Each module may comprise basic components, such as an analog/digital (A/D) converter, a multiplexer (MUX), a demultiplexer (DEMUX) and a microprocessor and its associated components (e.g., random access memory [RAM] and internal buses). Therefore, the module level can be considered the intermediate layer between the I&C unit level and the basic components level, although sometimes an I&C unit may be implemented with a single module.

To clarify the levels of detail, basic components build up modules. These modules carry out specific tasks, like input/output and communication. The modules can be inserted in a sub-rack in order to exchange data via the backplane bus between the modules. These modules together form an I&C unit, which can carry out a specific function, like voting. The subracks of the I&C units can be inserted in a rack or a cabinet, to form a division, which may consist of the pathway(s) from sensors to generation of an actuation signal. The actuation signal can be send to multiple actuators.

In summary, the following levels of detail have been defined for the analysis of hardware failure modes of a digital protection system:

1. **System level:** a collection of equipment that is configured and operated to serve some specific plant function as defined by terminology of each utility.
2. **Division level:** a system can be carried out in redundant or diverse divisions. In this case, a division may consist of the pathway(s) from sensor(s) to generation of an actuation signal. The actuation signal can be sent to multiple actuators. A division can be decomposed further in I&C units.
3. **I&C unit level:** a division consists of one or more I&C units that perform specific tasks or functions that are essential for a system in rendering its intended services. I&C units consist of one or more modules.
4. **Module level:** an I&C unit can be decomposed into modules that carry out a specific part of the process. For example, input/output-cards, motherboard, and communication cards, etc. An I&C unit may contain only a subset of these modules.
5. **Basic components level:** a module is composed of a set of basic components bounded together on a circuit board in order to interact. Consequently, the states of a module are the set of the combined (external) states of its basic components. Failure modes defined at the basic component level should be independent of design or vendor.

In principle, failure modes of a control system can be defined at the same levels of detail, but often control systems do not have redundant divisions, since they are usually not safety critical.

To clarify the definitions, the definitions are also illustrated in Figure 1 for a possible example system. Although it has been tried to provide clear definitions, in practice the distinction between the levels of detail is not sharp. For example, an I&C unit can be implemented using a single circuit board or module, and one may argue that an optical cable should be considered a basic component.

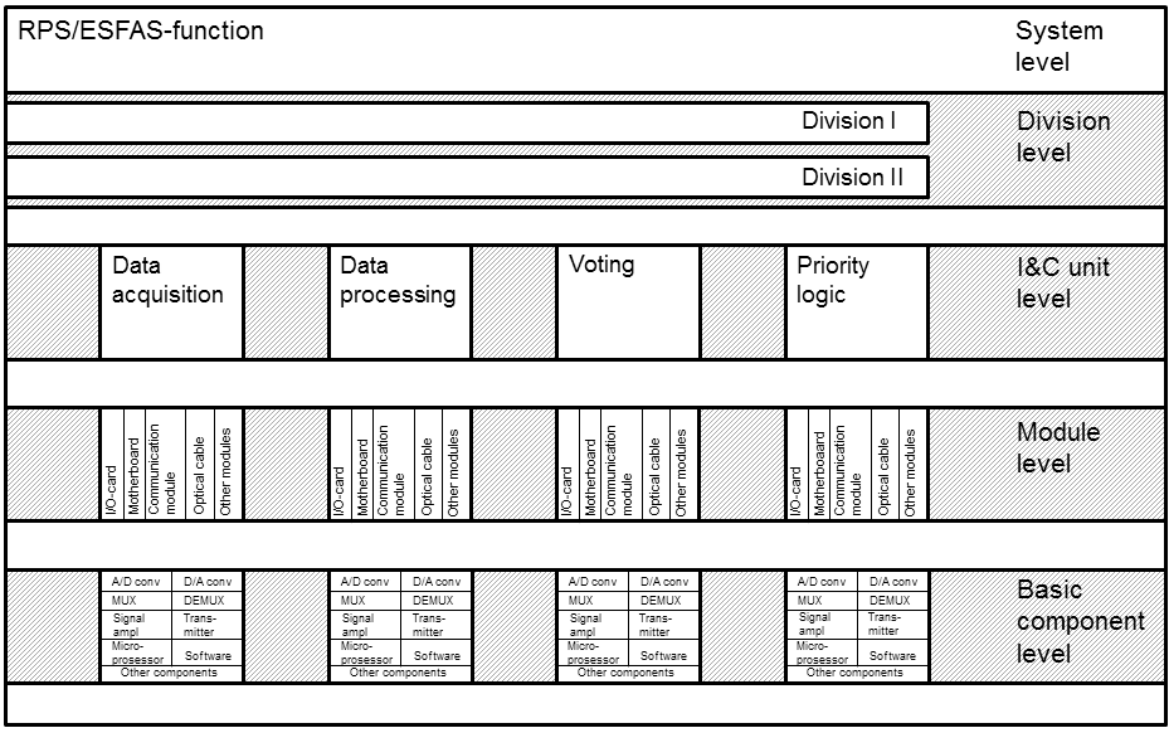

Figure 1. An Illustration of Levels of Detail for Hardware Failure Mode Taxonomy
In some systems, data acquisition and data processing are regarded as one I&C unit.

In general, failure effects for components[2] at a particular level of detail become failure causes of the components at the immediate higher level of detail. A clear definition of levels of detail facilitates the propagation of lower levels failure modes to the entire system.

2.2. Input Summary of Hardware Failure Modes

In Table 1, the contributions of the different organizations are summarized. For every level of detail, the components that are considered are given (in Column 2) as well as the failure modes at this level of detail (in Column 3). Failure modes are usually defined according to functionalities of the digital units (e.g., a component or module) characterized by input/output signals for the units.

Participants had different ways to define the failure modes. Some provided more general failure modes, like "failure to actuate" or "spurious failure". Others defined more descriptive failure modes, like "frozen sensor" or "amplifier adjustment too low". The more descriptive failure modes are more like failure causes and need to be evaluated in order to determine their the failure effects. The more general failure modes do give information about the effect, but not about the causes. Therefore, in the summary of the contributions of the participants, the general and descriptive failure modes are shown.

Table 1: Summary of the Components Considered at Each Level of Detail and the Corresponding Failure Modes

Level of detail	Components	Failure modes
System level	Entire system	Failure to actuate Failure to actuate in time Spurious actuation Descriptive failure modes: • Failure of support system; • Failure of acquisition • Failure of treatment and communication

Division level	Single division of RPS	• Undetected failures
		  o Loss of function
		  o Spurious function
		• Detected failures
		  o Loss of function
		• CCF
		• Corrective maintenance

I&C unit level	• Acquisition and processing unit (APU)
	• Logic processing module
	• Signal conditioning module
	• Actuation logic unit (ALU)
	• Voting processing module
	• Hardwired output logic for actuation
	• Digital trip module
	• Trip logic unit
	• Safety logic unit
		• Undetected failures
		  o Loss of function
		  o Spurious function
		• Detected failures
		  o Loss of function
		• CCF
		• Corrective maintenance

Module level	• Remote multiplexing unit
	• Input/output devices
	  o Digital I/O-modules
	  o Digital I/O channels
	  o Analog I/O modules
	  o Analog I/O channels
	• Load driver
	• Optical cable
	• PLC-module
	• Communication card
		• Undetected failures
		  o Loss of function
		  o Spurious function
		  o Malfunction
		• Detected failures
		  o Loss of function
		  o Malfunction
		• CCF
		• Corrective maintenance

[2] Note, components here do not mean the basic components. They simply represent a collection of components that constitute the equipment at a level of detail.

Level of detail	Components	Failure modes
	• Termination module • DC- power supply • Power battery • AC-power supply • Subrack	Example descriptive failure modes: • Loss of one sensor input • Intermittent sensor signal failure • Loss of an output • Loss of internal power supply • Internal power overshoot • Round-off/truncation/sampling rate errors • Unable to meet response requirements • Skipping software functions due to hardware/software faults or too fast triggered WDT • WDT fails to activate • WDT activates when computer has not failed • Arbitrary value output • Setpoint corrupted • Malfunction alarm of the PLC module of blackout diesel system (BDS) • Termination module D/I fails to close/open when energized/de-energized. • Card failure detected/undetected by software. • Card failure detected on panel check • Network interface Card fails to establish communication
Basic components level	• Current loop • A/D converter and D/A converter • Multiplexer • Demultiplexer • Sensors • Signal amplifier • Transmitters • Etc.	Failure modes defined for individual components according to their output. Example descriptive failure modes (Failure modes for some other components can be found in [3]): • Transmitter fails/drifts high/low • Amplifier output fails low • Amplifier output fails low due to CCF • Amplifier adjustment too low • Power supply output fails low • Sensor signal fails low • Transducer spuriously fails high • Termination module A/I fails/drifts high/low

The defined generic failure modes are almost identical at every level of detail except the basic component level. First, the meaning of these failure modes is discussed.

Failures denoted as “undetected failures-spurious function” are defined as failures not automatically detected, which result in a signal that will contribute to a spurious actuation; also known as nuisance failures.

Failures denoted as “undetected failures-loss of function” are defined as failures not automatically detected, which result in a failure to send a signal that can contribute to actuation when needed; for example, a failure to issue an actuation signal; also known as fail to danger.

The last category of failures represents detected failures. These failures are defined to be automatically detected and, more importantly, it is assumed that after a detected failure, the functional module will go to a predefined state; for example, generate a trip signal. The relevance of the detected failure is dependent on the predefined state of the module. Note that it may not be possible to correctly determine a predefined state for every possible failure that is detected.

Both loss of function and spurious function can be subdivided into several failure modes. A component can fail high, fail low, give erroneous outputs or get stuck. Depending on the design and the plant condition, these failure modes will result in either a spurious function or a loss of function. The impact of the failure modes is dependent on the design of the system. Moreover, in a digital system, it may be possible to detect failures but it may not be possible to react on the failures, e.g., by setting the output of a functional module to a predefined state. These are considered undetected failure modes in this case.

Although in an FMEA the generic failure modes and the descriptive failure modes can be used both, in a reliability model they are likely to be combined, so that both the cause and the effect can be interpreted. Therefore it is important to translate a descriptive failure mode to signal level to analyze the effects of a failure mode, i.e., to determine the behaviors of the system or the functional unit. To do this, the choices made in the design are very important.

3. COLLECTION OF SOFTWARE FAILURE MODES

3.1. Levels of Details for Software of Digital Systems

The levels of detail for software failure modes can be defined by considering the architecture of the digital system software, similar to the above definitions for digital hardware.

At the system level, digital protection system software consists of the collection of software running on various microprocessors of the system and failure modes can be defined at this highest level.

Considering the redundant divisions of an RPS, the collection of software running on the microprocessors of a single division may also fail and cause the failure of that division. Therefore, failure modes of all software belonging to a single division can also be defined at the division level as division level failure modes.

A more detailed level for failure mode definition is the microprocessor level (which may also be called module level because a module usually has only one microprocessor) for the software program running on a particular microprocessor. At this level of detail, the software is treated as an individual component like the microprocessor of a module. The software runs on a microprocessor and interacts with other hardware components via the microprocessor. This enables modeling of interactions between system hardware and software and capturing some fault-tolerance features. For example, the external watchdog timer of the main CPU in a digital feedwater control system can detect a failure of the CPU software to update its output and cause a fail-over to the backup CPU.

The software that runs on a microprocessor may be complicated enough such that it can be further decomposed, to a sub-level. That is, the software running on a microprocessor can be considered in elements such as input, output, communication etc. The sub-module level is the most detailed level in this study.

To summarize, the following levels of detail have been defined for software:

1. **System level:** for a digital protection system, at the system level, the software consists of the collection of software running on various microprocessors of the system and failure modes can be defined at this highest level.
2. **Division level:** for the redundant or diverse divisions of an RPS, the collection of software running on the microprocessors of a single division may also fail and cause the failure of that division. Failure modes of all software belonging to a single division can be defined at this level as division level failure modes.
3. **Module level/microprocessor level:** for the software program running on a particular microprocessor, the software is treated as an individual component like the microprocessor of a module.
4. **Sub-module level:** the software that runs on a microprocessor may be complicated enough such that it can be further decomposed, to a so-called sub-module level.

3.2. Summary of the Software Failure Modes

In Table 2, the software failure modes at different levels of detail achieved at the Bethesda workshop are provided. This represents preliminary results and will be further enhanced.

Table 2: Summary of Software Failure Modes at Each Level of Detail

Level of detail	Failure modes
System level	For an RPS • Failure to actuate (including failure to hold) • Spurious failure • Adverse effects on other functions (systems, operators) • (and others, dependent upon additional functions judged to be safety related) For load sequencing: • Failure to actuate in time For ESFAS: • Failure of trip signals such as a high reactor pressure level;
Division level	No failure modes were defined for this level of detail at the Bethesda workshop.
Microprocessor level / module level	Erroneous operation for data acquisition: • Incorrect value/incorrect validity • Incorrect value and incorrect validity • No value • No validity • Above failure modes may be subdivided, e.g. incorrect high or low Erroneous operation for logic processing: • Failure to actuate (including failure to hold) • Spurious failure Erroneous operation for voting logic: • Incorrect voting • No vote • Above failure modes can lead to a failure to actuate (including a failure to hold) or to a spurious failure Erroneous operation for priority actuation logic: • Incorrect priority • No priority • Above failure modes can lead to a failure to actuate (including a failure to hold) or to a spurious failure;
Sub-level	Failure modes are defined for software functional modules related to individual signals to hardware components such as pumps and valves.

Most participants considered software failure as a potential source of CCF, and some participants defined different levels of software failure due to CCF:

- Loss of the complete software system;
- Loss of (multiple) division(s);
- Loss of one or more specific software modules.

Almost all participant defined software as a source of CCF rather than go into more detail by defining CCF software failure modes at the lowest level of detail

4. OBSERVATIONS AND DISCUSSION OF ISSUES ON DIGITAL SYSTEM FAILURE MODES

4.1. Some Observations on Participants' Input

Some participants included information on failure effects (as a part of an FMEA); others did not. This is mainly because failure effects are not required when inputs were requested from participants. However, it is also partially due to the fact that failure effects are strongly dependent on system designs, especially the hardware and software fault tolerance features that vary from vendor to vendor.

In general, RPSs are of interest from a PSA point of view; and, therefore, most of the failure modes are developed specifically for protection systems, except for the failure modes at the basic component level. For hardware, the failure modes at the basic component level are applicable to both protection systems and control systems because they are all built upon the basic components regardless of the designs. The higher level failure modes are defined for protection systems based on the understanding of the simplicity of the architecture, functions, and data exchange within the system, and that the plant conditions become irrelevant once the trip signal is issued by the protection system. These characteristics of the protection systems are not shared by a digital control system. The failure modes of a control system can be much more complicated at a high level. For example, functionality of a control system may be complex and feedback from the controlled process needs to be considered when developing control system failure modes.

Another observation is that the participants provided more information on the hardware failure modes than on the software failure modes, but a preliminary set of software failure modes was developed during the Bethesda workshop.

The levels of detail at which failure modes were defined are very different, and at the same level of detail the failure modes from different participants may still be different.

Although considered one of the most important contributors of digital systems, CCFs are not mentioned separately by every organization. This is due to the fact that not all organizations regard CCF as a separate failure mode, but as a combination of failure modes due to a common cause. In this way CCF does not get too much attention in the taxonomies, but is an important factor in modeling.

Although there is no consensus on the modeling method, event tree/fault tree approach appears to be the most popular one.

4.2. Discussion of Issues in the Failure Mode Summary

The failure mode taxonomy summarized in Tables 1 and 2 represents work in progress, and further development/improvement of the taxonomy is needed. Regardless of differences between descriptive and generic failure modes, it is still questionable whether some of the generic failure modes for hardware are meaningful, and some of these issues are raised here including whether (1) CCFs should also be classified into the detectable and undetectable; (2) "malfunction" should be in parallel with failure modes "loss of function" and/or "spurious function"; and (3) "corrective maintenance" should be considered a failure mode.

It is also noted that there is no intermediate level between the division level and the module level for software failure mode taxonomy in Section 3.1. It might be worthwhile adding an I&C unit level, similar to hardware, by considering a collection of software running on an I&C unit and defining the related failure modes at this level of detail.

Furthermore, it needs to be considered whether failure modes can be defined unambiguously, clearly, completely/exhaustively, hierarchically, exclusively, and analogously between different components. On the other hand, due to the relative simplicity of RPS functions, the defined failure modes may be sufficient for performing FMEA and reliability evaluation of an RPS. In addition, data availability that is needed to support the defined failure modes is of concern; however, the data issue is considered out of the scope of this task.

5. CONCLUSIONS AND ONGOING WORK

The taxonomy of failure modes summarized here represents preliminary results collected from participants. The working group is developing a consensus failure mode taxonomy for both hardware and software based on the input of the participant and the discussions during the workshops. The development and improvement of such taxonomy is still an evolving process.

The developed taxonomy is expected to be used by PRA analysts with a potential of being a useful input to digital I&C system designers. The taxonomy needs to be defined and maintained at various levels of detail to provide options to PRA analysts and/or I&C system designers such that the taxonomy at a particular level of detail can be selected based on their own FMEA and/or reliability modeling need.

Modeling methods are another important topic. It is generally agreed that a modeling method needs to capture dependencies and fault tolerant features, use meaningful failure modes, and propagate failure effects. However, there is no commonly accepted method for modeling digital systems although event tree/fault tree method appears to be the most popular in PRA applications. Selection of a modeling method is, therefore, beyond the scope of this task, and the failure mode taxonomy should not eliminate useful modeling methods based on level of detail or degree of difficulty. Furthermore, it may be possible that a reliability model be developed with a mixture of levels of detail.

In order to demonstrate the usefulness of the developed taxonomy, the failure modes at different levels of detail are being or have been applied to an example digital I&C system by the group. For example, in the case of basic component failure modes, the demonstration can be performed by decomposing a selected module into basic components, identifying the failure modes of individual components, and propagating their failure effects to the module level, which become the failure modes of the selected module. A comparison between the failure modes obtained this way and the failure modes defined at the module level will be performed to remove any gaps, if any.

Improvements on the development and application of the failure mode taxonomy will be continuously pursued in this task to develop best practice guidelines on using a common taxonomy in risk and reliability assessments.

Acknowledgements

This work was performed under the auspices of the U.S. Nuclear Regulatory Commission.

References

[1] Nuclear Energy Agency, Committee on the Safety of Nuclear Installations, "Recommendations on Assessing Digital System Reliability in Probabilistic Risk Assessments of Nuclear Power Plants," December 17, 2009, available online at http://www.oecd-nea.org/nsd/docs/2009/csni-r2009-18.pdf.

[2] Korsah, K., Cetinerm, S. M., Muhlheim, M. D., and Poore III, W. P., "An Investigation of Digital Instrumentation and Control System Failure Modes," ORNL/TM-2010/32, March 2010.

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