A Complexity Assessment Methodology for Programmable Electronic Mining Systems

John J. Sammarco, P.E.; National Institute for Occupational Safety and Health (NIOSH); Pittsburgh, Pennsylvania

Keywords: system safety, Normal Accident Theory, complexity metrics, programmable electronics

Abstract

Mining, traditionally a low-tech industry, is now utilizing surprisingly complex programmable electronic (PE) systems. The functional safety of PE-based mining systems is an international issue and concern. From 1995 to 2001, there were 11 PE-related mining incidents reported in the U.S. and 71 PE-related mining incidents reported in Australia. These incidents are due, in part, to unprecedented levels of system complexity. The National Institute for Occupational Safety and Health (NIOSH) is addressing this issue of system complexity by conducting research to develop a quantitative complexity assessment methodology based on Normal Accident Theory (NAT). The methodology models the behavioral interactive complexity at the level of system requirements. A graph-theoretical approach is used for creating quantitative metrics from Software Cost Reduction (SCR) dependency graphs. This complexity assessment methodology will help realize simpler, safer systems that will be easier to validate and verify. The methodology will benefit mining and other industries as well.

Introduction

There is an increasing trend of embedding PE into a wide variety of systems. Some reasons to embed PE technology are to provide increased functionality, improve reliability and to make systems more cost competitive. Thus, traditional hardwired electro-mechanical and analog systems are often replaced with PE hardware and software. This trend will continue due to global market pressures and industry's quest for "improved" systems increasing functionality.

Mining, traditionally a low tech industry, is now utilizing surprisingly complex computerized systems. Today, PE technology is embedded in diverse mining systems such as "driver-less" underground and surface haulage vehicles, continuous mining machines, hoists and elevators and mine atmospheric monitoring systems. A recent survey reported that over 95 percent of all longwall mining systems are PE based (ref. 1). This widespread embedding of PE increases our dependence on and exposure to PE-based systems; more importantly, it can impact safety by creating new hazards that could result in injury or death.

The NIOSH, Pittsburgh Research Laboratory has a project addressing the functional safety of PE-based mining systems. The project consists of two major, yet overlapping parts. The first concerns mining industry specific best practice recommendations and guidance for PE-based mining systems (refs. 2-5). This work is largely based on IEC 61508 (ref. 6). The objective for the project's second part is to develop a complexity assessment methodology for PE-based mining systems. This paper describes the approach and methodology for complexity assessment research.

The next section describes how the research is motivated by the concern for safety. Next, the relationship between complexity and safety is discussed. The related research section draws distinctions between the research and related works in NAT and complexity metrics. The research methodology section describes the experiments and measurement methods. The system model section describes how the Software Cost Reduction (SCR) method is used to enable graph-theoretical approach for quantifying interactive complexity. Finally, the research status and contributions are given.

Research Motivation

The functional safety of PE-based mining systems is an international issue and concern (ref. 7). From 1995 to 2001, there were 11 PE-related mining incidents in the United States; four of these were fatalities (ref. 8). Most likely, the total numbers of incidents are under reported in the U.S. because near misses and some

mishaps are not reported. Australia reports all mining incidents; from 1995 to 2001 there were 71 incidents documented. In both countries, many of the incidents involved unexpected movements or startups of PE-based mining systems. This unpredictable system performance can and has created hazards resulting in injury or death.

PE functional safety is a major concern for many industries besides mining. PE-related mishaps causing mission failures, injuries, and fatalities have occurred in the chemical process industry, commercial and military aviation, public mass transit, and the medical electronics industy. Neumann (ref. 9) documented over 400 PE-related incidents and MacKenzie determined that about 2000 deaths were PE-related (ref. 10).

The general problem: Our ability to understand and manage the complexities of PE-based systems have not kept pace with the technology's utilization. As a result, PE-related incidents causing mission failures, harm to the environment, injuries, and fatalities have occurred.

We are ill equipped to understand and manage these complexities because there is no scientific methodology to identify and quantify the safety-related complexities of a PE-based system. Quantification is very important; Lord Kelvin stated,

> When you can measure what you're speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced it to the stage of understanding.
>
> William Thomson (Lord Kelvin) (1824-1907)
> Popular lectures and Addresses, 1891-4.

Research Objective and Specific Aims

The research objective is to develop a complexity assessment methodology for safety-critical PE-based mining systems. The first aim is to target the early life cycle stage of requirements. This enables early assessment so that complexities impacting safety can be better understood and reduced before they propagate to other development phases. Most errors occur at the requirements stage (refs. 11-12) and errors are much less costly to correct early rather than later (refs. 11, 13). The second aim is to create a generalized methodology that can be utilized by other PE-based systems because the safety of PE-based systems is not an issue unique to the mining industry; it is an issue for many industrial sectors. Specifically, unpredictable system performance, during normal or foreseeable abnormal conditions, is a common safety issue (ref. 14).

Complexity and Safety

Webster's dictionary defines complexity as follows:

> Complexity: (a) having many varied interrelated parts, patterns, or elements and consequently hard to understand fully; (b) marked by and involvement of many parts, aspects, details, notions, and necessitating earnest study or examination to understand or cope with (ref. 15).

Complex, PE-based systems directly impact safety because of the relationships between complexity, design errors, new hazards, and system accidents.

As PE utilization proliferates, escalating levels of system sophistication and complexity increase the likelihood of design errors. Littlewood (ref. 16) states, "The problems essentially arise from complexity, which increases the possibility that design faults will persist and emerge in the final product". Leveson (ref. 17) expands upon the consequences of PE-induced system complexity:

> Many of the new hazards are related to increased complexity (both product and process) in the systems we are building. Not only are new hazards created by the complexity, but complexity

> makes identifying analyzing hazards more difficult. This complexity can also lead to a system accident.

Perrow (ref. 18) defines a system accident as "the unintended interaction of multiple failures in a tightly coupled system that allows cascading of the failures beyond the original failures." Perrow theorizes that system accidents are inevitable or "normal" for complex, tightly coupled systems; therefore, system accidents are explained by NAT. This theory has much support (refs. 17, 19-23).

Quantification of NAT complexity would enable meaningful comparison of systems and options, help identify areas for simplification and be utilized in prediction models. With respect to safety, a complexity assessment must address system-level complexities. Safety cannot be assured if efforts are focused only on a part of the system because *safety is an emergent property of the entire system*. Safety emerges once all subsystems have been integrated. For example, the subsystem software can be totally free of "bugs" and employ numerous safety features, yet the system can be unsafe because of how software interacts with the other parts of the system. In other words, the sum might not be as safe as the individual parts.

Related Research

This section reviews related works in complexity metrics and NAT such that limitations and inadequacies are made evident with respect to NIOSH's complexity research.

Complexity Metrics: There is a plethora of complexity metrics. Zuse (ref. 24) characterizes 98 complexity measures. Ince (ref. 25) describes three categories of software complexity measures: lexemical counts, graph theoretical and system design structure. NIOSH researchers extended these by adding a category for "integrated" thus giving four categories:

Lexemic counts: Count key language entities such as keywords and operators.
Graph theoretical: Graph based system models are created and key graphic characteristics are calculated.
System design structure: Structure is defined in terms of internal and external module coupling.
Integrated: Integrated metrics synthesize existing metrics as multi-metric composites.

Coskun (ref. 26) takes an integrated metrics approach by using an interdisciplinary complexity model encompassing the domains of mathematics, computer science, economics, psychology and cognitive sciences, social science and system science. This model is used to measure architectural/structural complexity, data processing/reasoning/functional complexity, user interface complexity, and decision support/explanation complexity. It is quite desirable to define and measure multiple types of complexity because a single measure cannot capture the notion of complexity. However, these complexity metrics are for various *software layers*; hence, *system level* complexity is not addressed.

Measuring complexity is difficult and there has been limited success. McDermid (ref. 27) supports this by stating:

> Complexity is both a major problem and an enigma, as there are no easy and effective ways of measuring it. …There has been a lot of work on software complexity measures, but it is widely accepted that these are not adequate and in many cases are misleading…

There are numerous reasons for the limited success of metrics. Fenton's (ref. 28) analysis of software metrics identifies a fundamental flaw; most measurements lack a basis in measurement theory. Other problems with metrics include measurement ambiguities, potentially misleading results and the tendency to use concepts without validation.

There is a much broader flaw with respect to safety; many complexity metrics exist for the *subsystem of software*, but they do not address the *entire system*. Complexity must be addressed at the system level, as stated earlier, because safety is an emergent system property.

Normal Accident Theory: Perrow's accident theory identifies two important system characteristics, interactive complexity and tight coupling, that together make complex software driven systems especially prone to system accidents (ref. 18). Tables 1 and 2 respectively list Perrow's attributes for interactive complexity and tight coupling.

Table 1 – Interactive Complexity Attributes
Adapted from reference 18.

Complex System Attributes	Comments
Proximity	Close proximity of physical components or process steps, less underutilized space
Common-mode connections	Many common-mode connections
Interconnected subsystems	Many interconnections
Substitutions	Limited substitutions of people, hardware, or software; exacting requirements
Feedback loops	Unfamiliar or unintended feedback loops
Control parameters	Multiple and interacting control parameters
Information quality	Indirect, inferential, or incomplete information

Table 2 – Tight Coupling Attributes
Adapted from reference 18.

Tight Coupling Attributes	Comments
Time-dependency	Less tolerant of delays
Sequences	Invariant sequences
Flexibility	Equifinality or limited ways to reach the goal or implement a function
Slack	Little or no slack in system structure or behavior

Interactively complex systems have the potential to generate many nonlinear branching paths among subsystems. These interactions can be unexpected, unplanned, incomprehensible, and unperceivable to system designers or system users. Therefore, adverse outcomes are more likely, and it is less likely these situations will be mitigated by human intervention.

Coupling is a measure of the strength of the interconnectedness between system components. Tightly coupled systems have little or no slack thus, they rapidly respond to and propagate perturbations such that operators do not have the time or ability to determine what is wrong. As a result, human intervention is unlikely or improper.

NAT Limitations: There are limitations to Perrow's work. NAT lacks formal measures of interactive complexity and coupling. Secondly, the system characteristics of interactive complexity and coupling are loosely defined. Hopkins (ref. 29) supports this criticism and cites "the absence of criteria for measuring complexity and coupling" as significant limitations. Kates notes the same limitations stating "the absence of clear criteria for measuring complexity and coupling makes his (Perrow) examples seem anecdotal, inconsistent, and subjective." (ref. 30).

Finally, NAT has not been operationalized for PE-based systems. The NIOSH research will be the first to operationalize NAT for PE-based systems. Operationalizion involves quantification of empirical attributes or indicators by measurement or assignment of numbers and scales. It also includes the translation of informal definitions to observable operations and processes. This research also develops the methodology for *early* identification and quantification of complexity at the system level.

NAT Research: Wolf's empirical research strongly supports NAT (ref. 23). He operationalized NAT for petroleum refineries by creating an index of complexity. His conclusions validate NAT; refineries characterized by high complexity and tight coupling had more occurrences of accidental releases of hazardous materials and more fires and explosions. However, the research is limited to refineries and the index of complexity is not generalizeable across industrial sectors.

Methodology

The first phase of the complexity assessment research focuses upon measuring NAT interactive complexity. Complex interactions are those of unfamiliar, unplanned, or unexpected behaviors. In essence, the end user perceives system behavior as unpredictable. Also, these behaviors could be unobservable or not immediately comprehensible by the end-user as stated by Perrow. As a result, the system's usability declines and a hazardous situation is created. Therefore, it is hypothesized that there is a relationship between interactive complexity − the independent variable, and system predictability, observability and usability − the dependent variables.

Experiments: The research includes experiments with subjects using a PC-based simulation of a PE-based system. This enables measurement of the dependent variables. The experiments use two versions of the PE-based system to facilitate hypothesis testing. The first version serves as a baseline and the second manipulates the independent variable. For each version, subjects execute three test scenarios. This gives a total of six tests, all randomly assigned for each user. After each test, the subjects fill out a questionnaire designed to elicit their perceptions of the system.

Measurement Methods: A hybrid design is used for research hypothesis testing and the collection of quantitative and qualitative data. A weakness encountered in metrics research concerns the failure to measure what is *needed* (ref. 28); therefore, the Goal-Question-Metric (GQM) paradigm (ref. 31) is used to determine what needs to be measured. GQM is widely accepted as a very effective approach for this task.

The dependent variables are predictability, observability, and usability and they are measured subjectively from the perspective of the test subjects. For the collection of dependent variables, multiple methods are used. The methodology uses the Discount Usability Engineering method (ref. 32), a human subject questionnaire instrument, and computer-based simulations and scenarios of case study examples. The Discount Usability Engineering method is a very cost effective method having a maximum benefit-cost ratio with as few as three to five subjects. Another advantage lies with the method's simplicity; it is much less likely that errors and biases will be introduced from misapplication of complex methods. A questionnaire instrument is also used to collect the dependent variables of system predictability, complexity, and usability. The questionnaire is based on guidelines by Creswell (ref. 33). Portions of the questionnaire are adaptations of two highly respected and validated instruments, the Software Usability Measurement Inventory (SUMI) developed at the University College Cork, Ireland (ref. 34) and the Questionnaire for user Interaction Satisfaction (QUIS) developed at the University of Maryland (ref. 35).

The independent variables are measured by using a graph-theoretical approach for quantifying interactive complexity. An appropriate system modeling method is needed to accommodate this approach and is described in the next section.

System Model

There are numerous types of system models, each having different strengths, weaknesses, capabilities and purposes. Broadly speaking in the context of this research, a model must enable direct or indirect measurement of interactive complexity as captured by the system requirements. System requirements define "what" the system does – this defines system behavior by specifying system inputs (stimuli), system outputs (responses), and the behavioral relationships between the inputs and outputs. Typically, system requirements include nonfunctional requirements or constraints. Finally, the model must be suitable for safety critical embedded systems. Therefore, criteria were established to guide the selection of an appropriate model. The following subset of criteria was established for model selection:

- *Abstraction level*: The model must be applicable to the level of the system.
- *Projection*: The projection needs to capture the external viewpoint of an end-user.
- *Real-world applicability*: The model needs to handle the complexity and size of real-world systems.
- *Safety*: Undesirable or unexpected system behavior can create a hazard. This type of behavior can occur during normal or abnormal conditions; thus, both conditions must be modeled.

- *Simulation*: Capabilities must exist for the end-user to interactively execute the system such that end-user perceptions of complexity can be measured.
- *System type*: The model must be applicable to reactive systems, so the model must capture the system's behavior to external stimuli from the environment and humans.

Based upon the complete set of criteria, SCR formal method was selected.

SCR: This is a powerful method for the formal specification, analysis, and validation of complex, embedded systems. SCR was extended for system requirements; the extensions included the incorporation of *nonfunctional requirements* such as timing and accuracy constraints (ref. 36).
SCR is based on the Parnas four variable model that partitions a system as four elements: the environment, input devices, software, and output devices. The model's projection captures synchronous behavior, as viewed externally from the perspective of the user although the model does not explicitly depict the user as part of the system. Because, in the context of this work, complexities are from the user's perspective, we extend the Parnas four variable model to include the "liveware" element of the SHEL model (ref. 38) as depicted by Figure 1.

Requirements are commonly conveyed and documented by a requirements document written in prose. The SCR methodology translates this prose into a formal SCR specification. Figure 2 depicts the major components of a formal SCR specification.

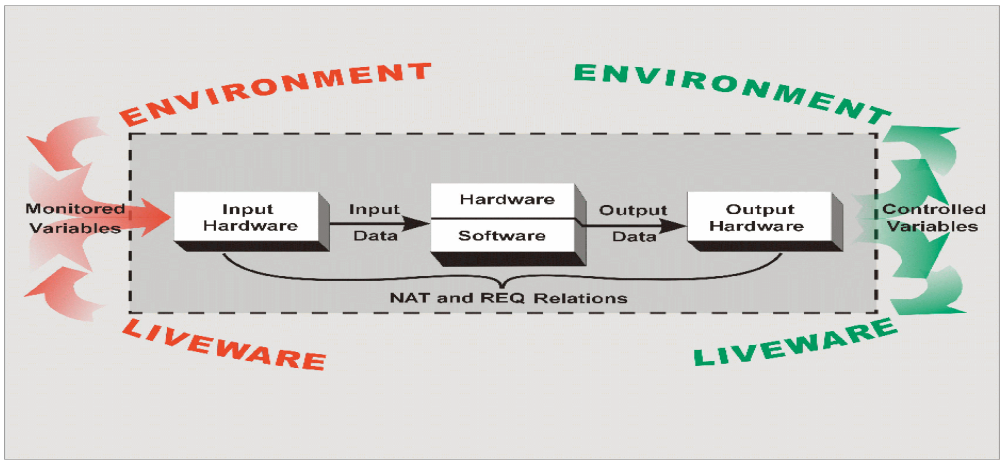

Figure 1 - The Parnas four variable model extended to incorporate the human component of the SHEL model.

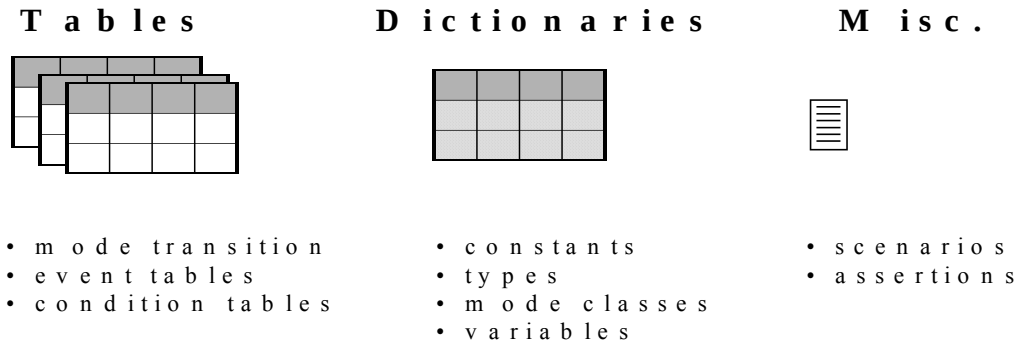

Figure 2 - Components of an SCR specification.

SCR is based on a finite state machine model of the system where the system $\sum$ is a 4-tuple (ref. 36),

$$\sum = (E^m, S, s^0, T), \qquad \textit{where} \tag{1}$$

E^m = set of input events
S = set of system states
s^0 = set of initial states where $s^0 \subseteq S$
T = the system transform

The Parnas four variable model is event driven. An event *e* is an instance in time when a mode, term, or variable changes value. For instance, an input event is when a monitored quantity changes; an output event is when a controlled quantity changes.

$e = @T(c)$; a basic event where condition *c* changes to true (2)
$e = @F(c) = @T(\neg c)$; a basic event where condition *c* changes to false (3)
$e = @T(c)$ WHEN $d = a \ \wedge \ a'$; conditional event where *a* = next state, *a'* = old state (4)

Additionally, an integrated environment called the SCR* tool set was developed (ref. 37). One of the environment's tools is a dependency graph browser that displays dependencies between SCR model variables (the controlled and monitored variables, modes, and terms) and gives a graphical overview of the specification as shown by Figure 3. The dependency graph also provides a mapping of controlled variables to monitored controlled variables. The graph depicts each SCR variable as a node; an arrow represents a dependency between nodes where value of the variable at the tail depends on the value of the variable at the head. As the system behavioral dependencies increase, the interactive complexities increase. Therefore, the dependency graph is used for direct and indirect measurements of interactive complexity.

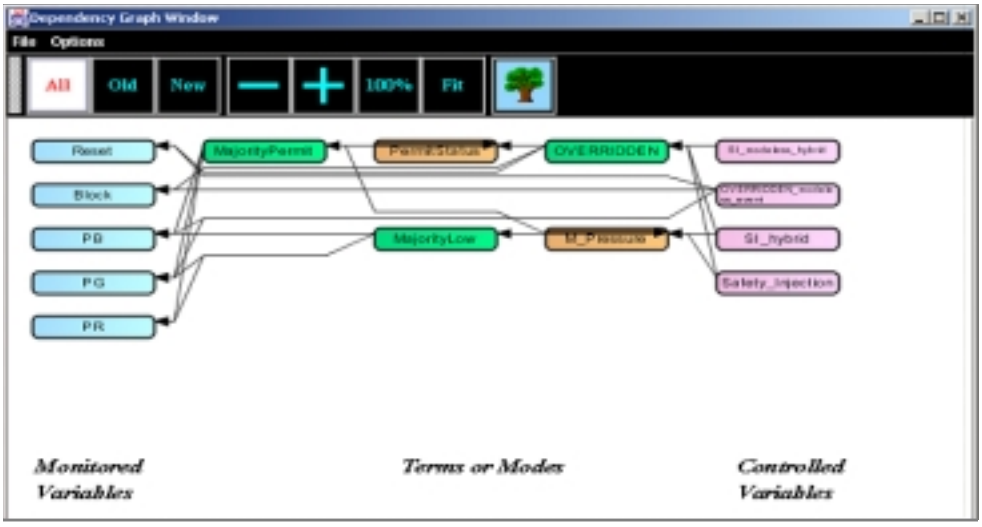

Figure 3 - An SCR dependency graph. *Source: Naval Research Laboratories.*

Research Status

The experimental design and research hypotheses are completed. The next stage is to conduct a "dry run" with users running a PC-based simulation of a generic type system. The Light Control System (LCS) case study was selected for this purpose. This case study was designed by the Fraunhofer Institute for Experimental Software Engineering in Kaiserslautern (ref. 39) for a seminar on Requirements Capture, Documentation and Validation. The dry run stage serves to facilitate refinement of the experiments and measurement methods. Testing of real-world system is scheduled to begin mid 2002.

Research Contributions

This work will be the first to operationalize NAT for PE-based mining systems. It addresses a major a weakness and limitation; NAT lacks scientific quantification, and it has not been applied directly to computer-based systems. These are addressed by a novel approach for quantifying NAT interactive complexity. The approached is summarized.

First, complexity metrics have traditionally been used to predict testing cost, development time and effort, number of errors, and numerous quality attributes. This work uses complexity metrics to address safety. The approach creates a measurement framework enabling the quantification of *system behaviors* that negatively impact safety. This negative impact of complexity is from the end-user's perspective of system predictability, observability, and usability. Thus, this avoids two pitfalls in metrics research: addressing a single dimension or representing multiple dimensions of complexity with a single number.

The quantification of system behavior takes place early in the system life cycle from models of system-level requirements. This enables complexities impacting safety to identified and analyzed *before* they are propagated to subsequent life cycle phases. This is in contrast to other approaches that target the *structural* aspects of the software subsystem. The structural aspects are not measurable until the design or test stages of the life cycle.

Lastly, the work can be generalized to PE-based systems in other industrial sectors.

References

1. Fiscor, S. The Goal Question Metric Approach, U.S. Longwall Census, pp. 22-27, 1998.

2. Sammarco, John J. Safety Framework for Programmable Electronics in Mining. Society of Mining Engineers, pp. 30-33, 1999.

3. Sammarco, John J., T.J. Fisher, J.H. Welsh, and M.J. Pazuchanics. Programmable Electronic Mining Systems: Best Practice Recommendations (In Nine Parts); Part 1: Introduction. *IC 9456,* NIOSH, Pittsburgh, PA, 2000.

4. Sammarco, John J., and T.J. Fisher. Programmable Electronic Mining Systems: Best Practice Recommendations (In Nine Parts); Part 2: 2.0 System Safety. *IC 9458,* NIOSH, Pittsburgh, PA, 2001.

5. Fries, E. F., T.J. Fisher, and C.C. Jobes. Programmable Electronic Mining Systems: Best Practice Recommendations (In Nine Parts); Part 3: 2.2 Software Safety. *IC 9460,* NIOSH, Pittsburgh, PA, 2001.

6. International Electrotechnical Commission (IEC). Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related System, Parts 1-7. *IEC 61508-1 to IEC 61508-7*. International Electrotechnical Commission, 1997.

7. Sammarco, J. J., J. L. Kohler, T. Novak, and L. A. Morley. Safety Issues and the Use of Software-Controlled Equipment in the Mining Industry. *Proceedings of IEEE Industry Applications Society 32nd Annual Meeting*, 1997.

8. MSHA. Fatal Alert Bulletins, Fatalgrams* and Fatal Investigation Reports. Available at www.msha.gov/fatals/fab.htm. Web page (accessed May 2001).

9. Neumann, Peter G. Computer Related Risks. The ACM Press, 1995.

10. MacKenzie, Donald. Computer-Related Accidental Death: An Empirical Exploration. *Science and Public Policy*. Vol. 21, pp. 233-248, 1994.

11. Davis, Alan M. Software Requirements: Objects, Functions, and States. Prentice Hall, 1993.

12. Lutz, Robyn R. Targeting Safety-Related Errors During Software Requirements Analysis. *The Journal of Systems Software*. Vol. 34, pp. 223-230, 1996.

13. Nelson, Mike, James Clark, and Martha Ann Spurlock. Curing the Software Requirements and Cost Estimating Blues. pp. 54-60, 1999.

14. Herrmann, Debra S. Software Safety and Reliability. IEEE Computer Society, 1999.

15. Gove, Philip B., Chief Editor. Webster Third New International Dictionary; Merrian Webster Inc.; 1996.

16. Littlewood, Bev, and Lorenzo Strigini. The Risks of Software. pp. 62-27, 1992.

17. Leveson, N. G. Safeware: System Safety and Computers. Addison Wesley Publishing Co., 1995.

18. Perrow, Charles. Normal Accidents: Living with High-Rish Technologies. Princeton University Press, Princeton, NJ, 1999.

19. Leveson, N.G. System Safety in Computer-Controlled Automotive Systems. *SAE Congress*, 2000.

20. Mellor, Peter. CAD: Computer-Aided Disaster. *High Integrity Systems*. Vol. 1, No. 2, pp. 101-156, 1994.

21. Rushby, John. Critical System Properties: Survey and Taxonomy. *Reliability Engineering and System Safety*. Vol. 43, No. 2, pp. 189-219, 1994.

22. Sagan, Scott D. The Limits of Safety. Princeton University Press, 1993.

23. Wolf, Frederick G. Normal Accidents and Petroleum Refining: A Test of the Theory. Nova Southeastern University, 2000.

24. Zuse, Horst. Software Complexity: Measures and Methods. New York, 1991.

25. Ince, D. C. The Influence of System Design Complexity Research on the Design of Module Interconnection Languages. *SIGPLAN Notices*, Vol. 20, No. 10, pp. 36-43, 1985.

26. Coskun, Erman, and Martha Grabowski. An Interdisciplinary Model of Complexity in Embedded Intelligent Real-Time Systems. *Information and Software Technology*, Vol. 43, pp. 527-537, 2001.

27. McDermid, John. Issues in the Development of Safety-Critical Systems. *Safety-Critical Systems: Current Issues, Techniques and Standards*. Felix Redmill, and Tom Anderson: Chapman and hall, 1990.

28. Fenton, Norman, and S. L. Pfleeger. Software Metrics: A Rigorous and Practical Approach. PWS Publishing Co., 1997.

29. Hopkins, A. The Limits of Normal Accident Theory. *Safety Science*, Vol. 32, pp. 93-102, 1999.

30. Kates, R. Book Review: Normal Accident. pp. 121-122, 1986.

31. Basili, Victor R., Gianluigi Caldiera, and H. Dieter Rombach. The Goal Question Metric Approach. *Encyclopedia of Software Engineering*. Vol. 2, John Wiley and Sons, pp. 528-532, 1994.

32. Nielsen, J. Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. In *Cost-Justifying Usability*. R. G. Bias, and D. J. Eds. Mayhew, Academic Press, pp. 245-272, Boston, MA 1994.

33. Creswell, J.W. Research Design: Qualitative and Quantitive Approaches. Sage Publications, 1994.

34. Human Factors Research Group. "SUMI Homepage." Web page, [accessed 7 February 2002]. Available at http://www.ucc.ie/hfrg/questionnaires/sumi/index.html.

35. Human Computer Interaction Laboratory. "QUIS " Web page, [accessed 7 February 2002]. Available at http://www.cs.umd.edu/hcil/quis/.

36. Heitmeyer, Connie. Requirements Specifications for Hybrid Systems. *Proceedings, Hybrid Systems Workshop III, Lecture Notes in Computer Science*. R. Alur, T. Henzinger, and E. SontagSpringer-Verlag, 1996.

37. Heitmeyer, C., J. Kirby, B. Labaw, and Bharadwaj. SCR*: A Toolset for Specifying and Analyzing Software Requirements. *10th Annual Conference, Computer-Aided Verification*, 1998.

38. Hawkins, Frank H. Human Factors in Flight. Ashgate Publishing Company, 1993.

39. Queins, S., G. Zimmerman, M. Becker, M. Kronengurg, C. Peper, R. Merz, and J. Schafer. The Light Control Case Study: Problem Description. *Journal of Universal Computer Science*. Vol. 6, No. 7, Special Issue on Requirements Engineering, 2000.

Biography

John J. Sammarco, P.E., NIOSH, P.O. Box 18070, Cochrans Mill Road, Pittsburgh, PA 15236-0070, USA, telephone – (412) 386-4507, facsimile – (412) 386-6764, e-mail – zia4@cdc.gov.

John J. Sammarco is an Electrical Engineer at the National Institute for Occupational Safety and Health (NIOSH), Pittsburgh Research Laboratory. His research has covered various aspects of mine safety since 1987; the safety of computerized mining systems has been the focus of this research since 1997. He leads a project to address the safety of programmable electronic systems and protection devices in mining. The scope includes development, test, approval/certification, maintenance, human/machine interfaces, and configuration management. The results of this project include the establishment of NIOSH Best Practice Recommendations for Programmable Electronic Mining Systems that serve to guide the mining industry in system development and safety assurance.

John is also a Ph.D. candidate in the Lane Computer Science and Electrical Engineering Department at West Virginia University, Morgantown, WV. He is also a member of the IEEE's Committee for Technology Accreditation Activities and he is an accreditation program evaluator for Computer and Electrical Engineering Technology Programs.