Operationalizing normal accident theory for safety-related computer systems

John J. Sammarco *

Pittsburgh Research Laboratory, Mining Injury Prevention Branch, National Institute for Occupational Safety and Health, P.O. Box 18070, 626 Cochrans Mill Road, Pittsburgh, PA 15236, USA

Abstract

Computer-related accidents have caused injuries and fatalities in mining as well as other industries. Normal accident theory (NAT) explains that some accidents are inevitable because of system complexity. NAT is a classic argument in organizational sociology although it has been criticized as having imprecise definitions and lacking criteria for quantifying complexity. These limitations are addressed by a unique approach that recasts this organizational theory into an engineering-based methodology to quantify NAT complexities of computer-based systems.

In this approach complexity is categorized as external or internal. External complexity is defined by the external behavior of a system, and is quantified by these dependent variables: system predictability, observability, and usability. Dependent variable data contain the perceptions of 32 subjects running simulations of a system. The system's internal complexity is characterized by modeling system-level requirements with the software cost reduction (SCR) formal method. Model attributes are quantified using 15 graph-theoretical metrics—the independent variables. Five of 15 metrics are correlated with the dependent variables as evidenced by structure correlations exceeding 0.25, with standard errors <0.10 and a 95% confidence interval. The results also show that the system predictability, observability, and usability decreased as NAT complexities increased. This research takes a step forward in operationalizing NAT for computerized systems. The research benefits mining and other industries as well.

* Tel.: +1 412 386 4507; fax: +1 412 386 6710.
E-mail address: JSammarco@cdc.gov

1. Introduction

Increasingly, computer technology is being embedded into a wide variety of systems. This technology can enable added flexibility, provide new functionality, and help make systems more cost-competitive. Thus, traditional hardwired electro-mechanical and analog systems, having well-known and predictable failure modes, are often replaced with computer hardware and software. This widespread use increases our dependence on and exposure to computerized systems—more importantly, it greatly impacts safety.

Computer-related accidents have caused harm to the environment, injuries, and fatalities. Over 400 computer-related accidents were documented up to 1995 (Neumann, 1995); it was estimated that 2000 deaths were computer-related as of 1994 (MacKenzie, 1994). The safety issues of computerized systems have extended to the mining industry. Traditionally thought of as low-tech, the industry is now using complex, computerized mining systems such as "driverless" underground and surface haulage vehicles, longwall mining machines, hoists and elevators, and mine atmospheric monitoring systems. From 1995 to 2001, 11 computer-related mining incidents in the US were reported by the Mine Safety and Health Administration; 71 computer-related mining incidents were reported in Australia (Sammarco, 2003).

The problem is that we are ill-equipped to identify, understand, and manage the particular safety issues of computerized systems. Systems utilizing computer technology are more complex; as a result, new hazards are created that are difficult to recognize or mitigate with traditional safety techniques. Traditional safety engineering techniques are being stretched to the limit because of many factors, including the "fast pace of technological change," "new types of hazards," and "increasing complexity and coupling" (Leveson, 2004).

To engineer safer computer-based systems, new approaches are needed. One approach establishes a new accident model based on systems theory (Leveson, 2004). The model is intended as a theoretical foundation for new safety analyses and approaches. Another approach uses an interdisciplinary complexity model encompassing the six domains of mathematics, computer science, economics, psychology and cognitive sciences, social science, and system science (Coskun and Grabowski, 2001).

Addressing complexity is important in safety analysis because as computer-based systems proliferate, system sophistication and complexity escalate and increase the likelihood of design errors and the introduction of new hazards (Littlewood and Strigini, 1992). Normal accident theory (NAT) explains that some system accidents are inevitable because complex systems are highly interconnected, highly interactive, and tightly coupled (Perrow, 1999). Although NAT is a classic approach in organizational sociology, it remains theoretical rather than empirical. To our knowledge, only one attempt to operationalize NAT has been made, not for computer-based systems but for a specialized application to petroleum refinery processing.

This paper presents a new approach to operationalize NAT as an engineering-based methodology, with the goal of quantifying system-level complexities of computer-based systems. A methodology is presented for early complexity identification and quantification of system requirements. This enables an early assessment of NAT complexities that can impact safety *before* they are propagated to other life-cycle phases. Also, changes are generally easier and less costly to implement at the requirements phase. Armed with an effective complexity assessment, one can compare options, target the requirements to simplify, and measure simplification efforts.

Table 1
A summary of the research hypotheses and associated rejection criteria

Null hypothesis H_0	Rejection criteria
(1) There is no correlation between NAT metrics and system predictability	Structure correlation $\geqslant .250$
(2) There is no correlation between NAT metrics and system observability	Standard error $\leqslant .100$
(3) There is no correlation between NAT metrics and system usability	95% Confidence interval does not cross zero
(4) Increasing complexity does not decrease system predictability	Wilcoxon sign-ranks test
(5) Increasing complexity does not decrease system observability	$Z \leqslant -1.645$
(6) Increasing complexity does not decrease system usability	p-Value $\leqslant .05$

The specific aims of this research to operationalize NAT as follows:

1. Identify a formal modeling method for system requirements which will afford quantification of NAT attributes.
2. Identify the NAT attributes to be operationalized with respect to system requirements.
3. Identify potential metrics for each NAT attribute to be operationalized.
4. Identify the metrics that are useful measures or indicators of NAT complexity.

Several hypotheses (Table 1) are formed to help realize these specific aims.

2. Normal accident theory

Perrow, an organizational theorist, is the originator of NAT. His work emerged in 1979 when he was advising a Presidential commission investigating the accident at Three Mile Island (TMI Harrisburg, PA). In essence, Perrow identified system complexity as the primary accident cause; thus, the TMI accident was labeled a normal accident because this type of accident is inevitable with complex technological systems (Perrow, 1999).

NAT identifies two important system characteristics—interactive complexity and tight coupling—that make complex systems especially prone to system accidents. Interactively complex systems have the potential to generate many branching paths among subsystems. These interactions can be unexpected, unplanned, incomprehensible, and even unperceivable to system designers or system users. 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.

2.1. NAT limitations

NAT is limited in its applicability. First, it addresses a narrow category of accidents—industrial *disasters* of unforeseen events resulting in great damage and loss. Thus, it has not been extended to more commonly encountered accidents of limited scope. Secondly, NAT addresses safety in the context of organizational structures for complex, industrial systems such as nuclear power plans, oil refineries, and chemical plants. Thus, it does not focus on the details of the system and its components. Thirdly, the theory has not been extended to computerized systems using software. This limitation is realized by Perrow: "The metaphor

of an accident residing in the complexity and coupling of the system itself, not in the failures of its components has seeped into many areas where I never thought to apply it" (Perrow, 1999, p. 354). Perrow cites software as a neglected or new area to consider.

NAT is also limited by a lack of refinement in defining and quantifying its terms and concepts. "Ill-defined concepts" and "the absence of criteria for measuring complexity and coupling" have been cited as significant limitations (Hopkins, 1999). Quantitative measures of interactive complexity and coupling would address these limitations and could serve to promote the theory in new areas.

2.2. Related NAT research

The validity and application of NAT to petroleum refineries has been researched (Wolf, 2000). A refinery system was modeled as a hierarchy of system units, links, and nodes. Links are the system pipes that carry raw material, byproducts, final product, and wastes. Nodes are points of connection and interconnection between unit processes and links. They are also the points for control and monitoring of parameters such as flow, pressure, and temperature. Using this system model, a "refinery-specific" index of complexity was created based on refinery process knowledge and the number of unit processes, links, and nodes. This index, C_{iplant}, served to quantify and estimate the interactive complexity for a refinery. C_{iplant} represented the maximum number of states the system could exist.

Wolf's conclusions support the validity of NAT. Refineries characterized by high complexity and tight coupling had more occurrences of accidental releases of hazardous materials and more fires and explosions. However, two limitations are evident in this research. First, the index of complexity C_{iplant} is specific to refineries and is not generalized to other applications. Second, NAT was validated for a narrow spectrum of accidents: refinery disasters involving untoward releases of hazardous material, fires, and explosions. Therefore, NAT was not expanded to other types of accidents besides disasters.

Coskun and Grabowski (2001) addressed the challenge of measuring complexity by using an integrated metrics approach. This approach used an interdisciplinary complexity model encompassing six domains. This interdisciplinary complexity model was used to measure the complexity of software. Software complexity is important to address but software metrics alone are not sufficient to address safety because safety is an emergent property of the *system.*

3. Operationalizing NAT

Operationalizing NAT transfers the theory to practice by establishing concrete, quantifiable measures of system complexity. The operationalization process involves establishing a conceptualized system model, identifying which NAT attributes to measure, and defining multiple metrics to measure or indicate the NAT attributes.

3.1. System model

The first challenge is to formally model the specified behavior (requirements) of a system. System requirements define *what* the system shall do, defining system behavior by specifying system inputs (stimuli), system outputs (responses), and the behavioral relationships between the inputs and outputs. The model needs to provide an abstraction to

The Software Cost Reduction (SCR) method was used for specifying and modeling a system model. SCR is based on the Parnas Four-Variable Model. This model is essentially a black box view of system inputs, outputs, and external behavior; thus, the model captures the required external behavior, devoid of implementation or structural design aspects. SCR is based on a finite state machine model of the system where the system $\sum$ is a 4-tuple, $\sum = (E^m, S, s^0, T)$, where E^m = set of input events, S = set of system states, s0 = set of initial states with $s^0 \subseteq S$, and T = the system transform (Heitmeyer et al. 1998).

Additionally, an integrated environment called the SCR toolset was developed for formally specifying, modeling, simulating, and analyzing complex systems. The toolset includes a Dependency Graph Browser that displays dependencies between SCR model variables as a directed graph (Fig. 1). The dependency graph also provides a mapping of controlled variables (outputs) to monitored variables (inputs). Each variable is depicted 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. Another tool is available for creating a user interface for the system model. The user interface can provide transparent control of system simulations.

3.2. NAT attributes

NAT identifies 13 attributes of complex systems and categorizes them as either interactively complex or tightly coupled. Our research established a more abstract categorization of external and internal complexity as part of our inductive process to identify the NAT attributes to operationalize.

External complexity was characterized with three variables: system predictability, observability, and usability. These variables were viewed in the context of an operator interacting with the system. Situations of poor system predictability, observability, and

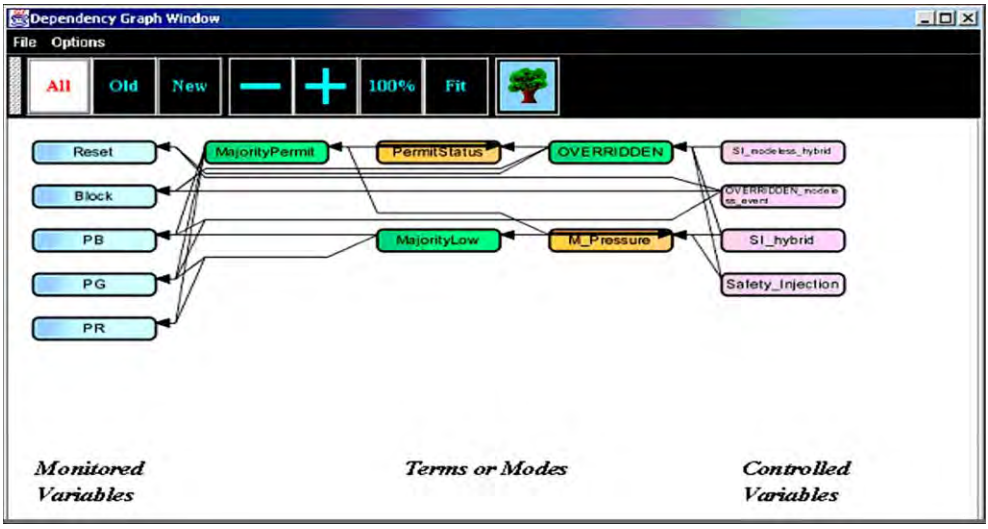

Fig. 1. An SCR dependency graph. *Source:* Naval Research Laboratories.

usability can contribute to human error or worse, mishaps. For instance, predictability concerns unfamiliar, unplanned, or unexpected system behaviors as viewed system's operator. These behaviors can result in unplanned machine movements or unexpected machine startups. A specific example concerns unpredictable mining machine movements that occurred in the US and Australia that in some cases resulted in injury or even fatalities (Sammarco, 2003). Complex system behaviors can also be transparent making them difficult to observe or comprehend by the end-user (Perrow, 1999). Observability also declines if the end-user is overwhelmed with information as happened to operators during the Three Mile Island mishap. Poor predictability and observability can negatively impact system usability. Hence, system complexity can be indicated by an external component characterized with three variables: system predictability, observability, and usability. These were the dependent variables for our research.

Internal complexity concerns a system's internal structure. Internal complexity was characterized by modeling system requirements with SCR, and quantifying NAT attributes represented in a SCR dependency graph with graph-theoretical metrics—the independent variables. Specific NAT attributes to operationalize were identified by deduction: (1) abstracting NAT attributes of complexity to a generalized view of simple (linear) and complex (nonlinear) systems; (2) selecting a subset of NAT attributes pertaining to linear and nonlinear systems; (3) identifying a general set of metrics to measure or indicate the subset of NAT attributes.

3.3. System linearity

Simple systems are linear. A single line of dominos provides an example. A single disturbance of a domino starts a linear chain of events where one domino pushes over the next. This chain of events follows a highly observable, predictable, and linear sequence of events.

Nonlinear systems are complex. They have multiple branching paths to system components and subsystems; hence, nonlinear systems are highly interconnected. A car windshield provides a nonlinear system example. A single disturbance, such as a stone hitting the windshield, results in multitudes of nonlinear, interconnected cracks. The extent and pattern of the crack is unpredictable and incomprehensible.

A high-level abstraction of system linearity was used to select a subset of three attributes pertaining to linearity from the 13 NAT attributes. The resulting NAT attributes were interconnectivity, common-mode connections, and multiple control parameters. A set of 15 metrics were proposed to operationalize these NAT attributes.

3.4. Metrics

Graph-theoretical metrics were used to measure system linearity of SCR dependency graphs. For instance, interconnectivity was indicated by using McCabe's cyclomatic complexity $V(g)$—the number of linearly independent paths. The directed graph of Fig. 2 depicts

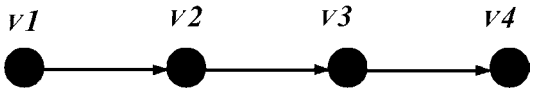

Fig. 2. A simple linear system having one path.

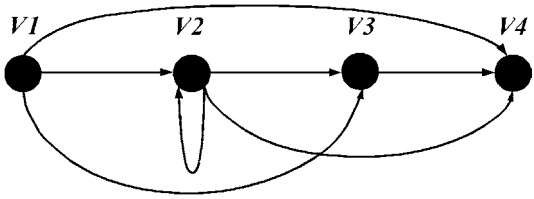

Fig. 3. A nonlinear system having five paths.

a simple system where $V(g) = 1$; thus, the single path from $v1$ to $v4$ indicates very low interconnectivity. A nonlinear system is depicted by Fig. 3, where $V(g) = 5$; thus, the five linearly independent paths from $v1$ to $v4$ indicate more system interconnectivity.

Common-mode connections increase as a system becomes more nonlinear. For instance, vertices $v2$, $v3$, and $v4$ of Fig. 3 have a common-mode connection established by vertex $v1$. Vertex $v2$ is another common-mode connecting $v2$ to $v3$, $v4$, and to itself via a self loop at $v2$. The out-degree metric quantifies a common-mode connection. The out-degrees of vertices $v1$ and $v3$ are $\text{od}(v1) = 3$ and $\text{od}(v2) = 3$. By comparison, $\text{od}(v1) = 1$ and $\text{od}(v2) = 1$ for the simple linear system of Fig. 2.

Multiple control parameters are used to determine the paths of control in a graph. The number of control parameters increases as a system becomes more nonlinear. For instance, the number of edges into a vertex (in-degree) is indicative of the quantity of control parameters for that vertex. The in-degree of vertex $v4$ (Fig. 3) is three in comparison to a value of one for vertex $v4$ (Fig. 2).

This section has presented just a few of the metrics for the NAT attributes of interconnectivity, common-mode connections, and multiple, interacting control parameters. Ultimately, a set of 15 metrics $\{X1, X2, X3, \ldots, X15\}$ were identified as candidates to operationalize NAT. This set of metrics was based on three system abstractions and projections (perspectives). The rationale was that a single abstraction or projection could not afford all the necessary metrics because complexity is multidimensional. These abstractions were created from SCR dependency graphs, as follows:

- Scenario subgraph; a course-grained abstraction induced by the dependency graph edges and vertexes that are used for a given set of user tasks.
- Critical-state subgraph; a medium-grained abstraction derived from the scenario subgraph.
- Critical-vertex subgraph; a fine-grained abstraction for each vertex of the critical-state subgraph.

These projections were created for the three subgraph abstractions:

- Input projection; a view of all dependencies with respect to the input vertices (i.e., all the ancestors of a given input vertex).
- Output projection; a view of all dependencies with respect to the output vertices (i.e., all the descendents of a given output vertex).
- All projection; a view of all dependencies with respect to input and output vertices.

4. Methodology

4.1. Research procedures

Dependent variable data were obtained from subject perceptions of a PC-based simulation of a light control system (LCS). The experiment consisted of two major parts. First, subjects learned the operation of the LCS. Next, subjects followed written instructions to run three test scenarios on a PC-based LCS simulator; each scenario was a set of typical user tasks. After each scenario was completed, subjects answered a questionnaire concerning subject perceptions of the LCS, took a short break, and then began the next test scenario.

The questionnaire was used to quantify the dependent variables. It was based on two respected and validated instruments: the questionnaire for user interaction satisfaction (Human Computer Interaction Laboratory, 2004) and the software usability measurement inventory (Human Factors Research Group, 2004). Closed and open-ended questions were used, with a five-level Likert scale from 1 (lowest) to 5 (highest) for the closed-ended questions. A portion of the questionnaire is given in Appendix.

4.2. Research design

The design was based on a cross-over design—a standard design with an established validity. The research also used a standard usability evaluation method called the discount usability engineering method.

The cross-over design used two treatments (A and B) and two washout periods (breaks). The independent variables were manipulated to increase NAT complexity for treatment A; treatment B had the independent variables manipulated to decrease complexity. A washout or waiting period was established between treatments to minimize carryover or residual learning effects from the prior treatments. The washout periods were just a few minutes because the residual effects were not physiological. Also, short washout periods were needed to keep the total test time relatively short. Lengthy washout periods could have confounded data because of subject fatigue or boredom.

The basic sequence was to give half the subjects treatment A, let the subjects rest during the washout period, and then have subjects receive treatment B. The other half of the subjects had the same treatments, but the order was reversed. Thus, given these sequences, the cross-over design had significant advantages: the subjects served as their own control, there was greater sample size efficiency with randomization of treatment order, and all subjects received all the treatments.

Treatments A and B were both applied to each of the three scenarios. The sequence of scenarios and treatments were optimized for a cross-over design (Jones and Kenward, 1998). Tables 2 and 3 list the test sequences. Half the subjects were randomly assigned to sequence 1 and the other half to sequence 2.

The discount usability engineering method was used to evaluate system usability, which was a dependent variable. The method uses three techniques: scenarios, simplified thinking aloud technique, and heuristic evaluation. The simplified thinking aloud technique encourages the subjects to vocalize their thoughts as they perform typical tasks. Observers

Table 2
Sequence 1 ordering of scenarios and treatments

Order	Scenario		Treatment
0		a	
1		b	
2	1		A
3		b	
4	2		B
5		b	
6	3		A
7		b	
8	2		A
9		b	
10	1		B
11		b	
12	3		B

a—Warm-up session.
b—Washout period.

Table 3
Sequence 2 ordering of scenarios and treatments

Order	Scenario		Treatment
0		a	
1		b	
2	3		B
3		b	
4	1		B
5		b	
6	2		A
7		b	
8	3		A
9		b	
10	2		B
11		b	
12	1		A

a—Warm-up session.
b—Washout period.

recorded these thoughts and encouraged the users to vocalize their thoughts and provide user feedback.

4.3. Research vehicle

The light control system (LCS) was used as the research test vehicle. The LCS requirements were formalized as a case study by the Fraunhofer Institute for Experimental Software Engineering for requirements engineering seminars (Queins et al., 2000). The LCS offered several advantages for research. First, "the light control case study is an example of a nontrivial reactive system" (Kronenburg and Peper, 2000). It represented a relatively complex, real-world system in that it required sensors, actuators, software, human machine

interfaces, automatic control functions, manual override functions, and fault management functions for the detection, annunciation, and tolerance of faults. Lastly, it afforded human/computer interaction.

The LCS was to control the interior lighting of a building consisting of various offices, laboratories, hallways, and staircases such that energy was not wasted and such that a safe environment was maintained for normal and abnormal conditions. An office environment is relatively benign with respect to safety. Loss of lighting can result in trip and fall hazards. More dangerous hazards would exist if the LCS was used in an industrial environment such as underground mining where moving equipments, rotating machinery, high-voltage electrical circuits, and unstable roof conditions are common. Several LCS requirements specifically address fault tolerance and safety aspects applicable to safety-related applications.

Briefly, the LCS provides automatic and manual control for two groups of office lights: one group is near the window and the other is near the wall. The control enables the user to set two light scenes named occupied and vacant. The occupied light scene automatically maintains a user-defined lighting intensity and light group configuration when the office is occupied. The office lights are also dynamically controlled to provide a constant level of illumination in spite of variations in sunlight entering the office. The vacant light scene automatically provides a user-defined light intensity and configuration if the office is vacated for an extended time that the user defines. Lastly, the LCS provides manual lighting control to over-ride the automatic controls. Manual pushbutton switches enable on/off control of each light group.

The LCS components consist of sensors, a logic solver, wall and window light actuators, and a user-interface panel. Five sensors are used; a motion sensor detects an occupied or vacant office; an analog sensor measures natural light in the office; a door closed contact indicates the door is open or closed; two status-line sensors indicate if the lights are turned on or off. The logic solver is PC-based and it provides control functions and a user-interface. Manual pushbutton switches enable manual control of each light.

The LCS system-level requirements were modeled using the SCR toolset. These requirements defined end-user needs, nonfunctional needs, and the required behavior of the system hardware components that included five sensors, two actuators, two pushbuttons, and two graphical user interfaces (GUIs). A model of nonideal LCS behavior (Heitmeyer and Bharadwaj, 2000) was expanded to provide new functionality needed by the research. A new PC-based GUI for the LCS was also created. The control requirements for offices and laboratory spaces were identical; therefore, the problem space was scoped to a model for a single office for the research described by this paper.

4.4. Subjects

Thirty-two subjects from the National Institute for Occupational Safety and Health participated in testing. All subjects were recruited as volunteers by word of mouth. Thirty-two subjects participated in the LCS tests and are characterized as follows based upon subject data collected during pre-test activities:

- 78.1%—technical job classification;
- 71.8%—45–65 years old;
- 87.3%—male;
- 100%—no prior involvement in the research;

- 84.4%—no knowledge of the light control system test vehicle;
- 84.4%—PC experience rated at 4 or 5 (expert).

4.5. Observers

Three additional volunteers were test observers that administered the tests. The observers did not know the purpose of the research nor understand the operation of the LCS. This was intentional so as to reduce the potential for observer-induced biases.

The observers gave the subjects instructional material for the using the LCS and GUI. Multiple delivery methods were used for instruction to accommodate subjects who learn by reading, watching, listening, or by hands-on activities. First, subjects watched a narrated PowerPoint presentation giving an overview of LCS. The presentation contained a video that provided a dynamic example of using the LCS and GUI. Next, written instructions were given. Lastly, the observers instructed subjects to run a warm-up session to gain hands-on experience.

Observers also collected the subject questionnaires and qualitative data in the form of observer notes. During the testing, observers took notes on each subject's verbal comments, actions, and body language with respect to predictability, observability, and usability. The qualitative data of observer notes were quantified by using a process of categorizing the data to the dependent variables and mapping the data to a five-point Likert scale. Once the observer data were quantified, the mean values for each category were weighted by 30%, and then combined with questionnaire data for predictability, observability, and usability.

5. Results and discussion

5.1. Subject responses

The frequency of subject responses for each scenario and treatment were depicted by histograms. In general, the treatment B histograms for predictability, observability, and usability are skewed to the right (the highest level 5) more than the histograms for treatment A. This indicates that treatment B (less complex) was generally perceived as having better predictability, observability, and usability than treatment A.

Table 4 lists the median and mode subject responses for predictability, observability, and usability for treatments A and B. Observations of these data also indicate that

Table 4
Mean and mode of each dependent variable and treatment for all scenarios

	Scenario 1 treatments		Scenario 2 treatments		Scenario 3 treatments	
	A	B	A	B	A	B
Predictability median	2.42	4.38	2.5	4.67	2.64	4.95
Predictability mode	3.0	5.0	2.5	5.0	4.67	5.0
Observability median	3.79	4.38	3.99	4.6	4.14	5.0
Observability mode	3.86	4.0	5.0	5.0	4.71	5.0
Usability median	3.75	4.67	4.0	4.5	4.04	4.25
Usability mode	4.25	5.0	4.0	5.0	4.5	5.0

treatment B was more predictable, observable, and usable because the median and mode values for treatment B are all greater than for treatment A with only one exception—the mode values are equal for observability of scenario 2.

5.2. Internal validity analyses

All subjects answered all questions of the questionnaire each time they completed a scenario. The questionnaire data had numerous internal validity checks to identify confounding, or invalid data, to assess data reliability, and to evaluate subject learning and fatigue effects.

Each potential threat is listed and discussed as follows:

- *Data confounding from the scenario instructions.* Subject responses for the dependent variables predictability, observability, and usability could be biased due to subjects having difficulty following and understanding scenario instructions. This seems unlikely based on the warm-up data for variable W1—the mean value for the subject's ease of following and understanding the warm-up instructions. Of 28 subjects, 24 rated W1 very favorably with a greater than 3.94 out of a maximum of 5.0. The distribution for W1 had a positive skew to the right (the highest score) as depicted by Fig. 4.
- *Data confounding from the graphical user interface (GUI).* Biased responses for the dependent variables predictability, observability, and usability could be due to subjects having difficulty with the GUI. This seems unlikely based on the warm-up data for variable W2—the mean value for the subject's ease of using the GUI to run the warm-up. Of 28 subjects, 24 rated W2 greater than 3.58 out of a maximum of 5.0 score. The distribution for W2 had a positive skew to the right (the highest score) as depicted by Fig. 4.
- *Invalid data.* All subjects answered all questions of the questionnaire; however, data from four subjects were eliminated because of consistent strings of high ratings and because these data were inconsistent with observer data. For instance, out of 36 questions, 34 were rated 5.0 (highest rating) and two questions were rated 4.0. This contrasted with the observer's data which indicated much lower ratings.

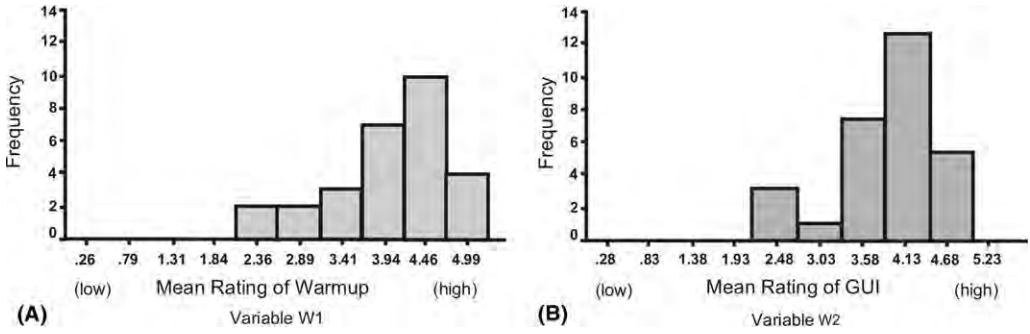

Fig. 4. Mean subject responses for the warm-up session. Graph (A) depicts the ease of following and understanding the warm-up instructions. Graph (B) depicts the GUIs ease of use.

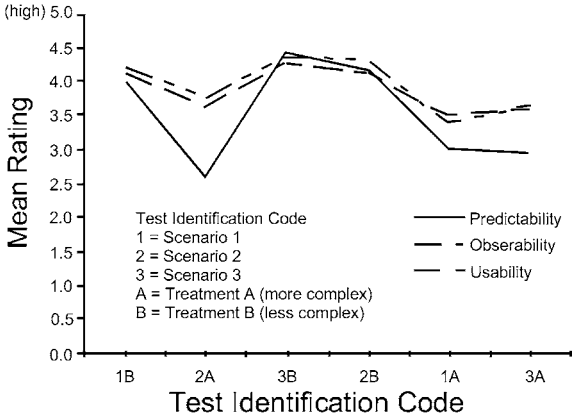

Fig. 5. The mean values of each dependent variable for the sequence 1 test order. $N = 14$ subjects.

- *Data reliability*. The data reliability was accepted given Cronbach's $\alpha = .811$. An α of .70 or higher is a typical benchmark of acceptability.
- *Learning effects or subject fatigue*. From inspection of data trends, one can infer learning effects and fatigue. A positive-sloped trend could be an indication that subjects are learning more as time progresses; thus, they would rate dependent variables with a higher value. A negative-sloped trend could be an indication of subject fatiguing as time increases, so they would rate dependent variables with a lower value. The data trends for the sequences 1 and 2 test orders were used to infer data confounding from learning effects or fatigue. Fig. 5 depicts the data trends for sequence 1. Data confounding was not detected in either graph of sequence 1 or 2 data trends.

5.3. Hypotheses testing

Hypotheses 1–3 concern the existence of correlations between subject perceptions of the system (the dependent variables of predictability, observability, and usability) and the set of 15 NAT metrics of system complexity (the independent variables). Testing of hypotheses 1–3 used canonical correlation analysis (CCA) and structure correlations.

CCA is a multivariate analysis technique used to identify multiple correlations between sets of independent and dependent variables. CCA produces a set of paired canonical variates representing the independent and dependent variables so as to maximize the correlation. The canonical variates consist of weighted sets of the original variables. The weightings are called canonical coefficients.

These raw canonical coefficients can be difficult to interpret, but structure correlations are very useful to facilitate their interpretations (Cliff, 1987; Shafto et al., 1997). Structure correlations are derived from the raw canonical coefficients and represent the Pearson correlation of each original variable to the canonical variate.

The results showed structure correlations exceeding .25 for five metrics. The five NAT metrics that correlated with the dependent variables are listed in Table 5. The structure correlations for the first pair of canonical variates are depicted in Fig. 6.

Table 5
The NAT attribute metrics and their associated abstractions and projections

NAT attribute	Metric	Abstraction	Projection
Interconnectivity	*X*13—cyclomatic complexity	Critical vertex	All
Common-mode connections	*X*7—out-degree	Critical state	Output
Control parameters	*X*5—number of state changes for a given input	Critical state	All
Interconnectivity	*X*2—cyclomatic complexity	Scenario	Output
Control parameters	*X*6—in-degree	Critical state	Input

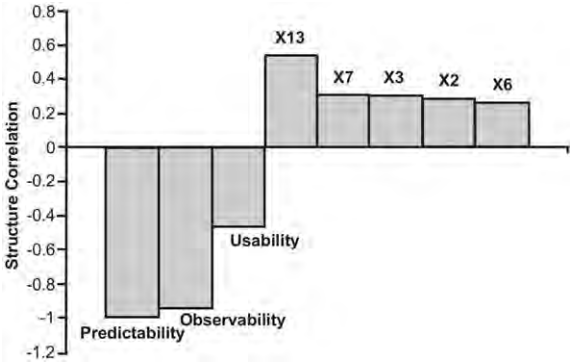

Fig. 6. A graphical depiction of structure correlations for the first pair of canonical variates. Note the negative correlation between the canonical variates.

The bootstrap re-sampling method (Efron and Tibshirani, 1993) was used to obtain the estimates of standard error and the 95% confidence limit for the structure correlations. The bootstrap method takes repeated samples to approximate the distribution of the original population. The re-sampling was done preserving the treatment group sample sizes. The results of 1000 bootstrap samples were standard errors of less than .10, and a statistically significant 95% confidence interval as listed in Table 6.

A Wilcoxon signed-ranks test was used to test null hypotheses 4–6—increasing NAT complexity does not decrease system predictability, observability, and usability. The results (Table 7) showed that subjects perceived the test scenario outcomes of the complex system (treatment A) as less predictable, observable, and usable in comparison to the simpler system (treatment B). The statistical significance measure was determined by using 1-tailed *p*-values.

Table 6
Structure correlations and statistical significance measures for the first pair of canonical variates

Independent variables	Structure correlation	Standard error	Confidence limits	
			5%	95%
*X*13	0.542	0.07596	0.420	0.665
*X*7	0.316	0.08953	0.175	0.463
*X*5	0.315	0.08834	0.174	0.461
*X*2	0.294	0.08835	0.148	0.438
*X*6	0.272	0.09122	0.127	0.421

Table 7
Wilcoxon signed-ranks test results for treatments A and B

Wilcoxon signed-ranks test (1-tailed)	Predictability (treatments B–A)	Observability (treatments B–A)	Usability (treatments B–A)
Z	−7.230[a]	−6.014[a]	−5.574[a]
p-Value	.000*	.000*	.000*

* Statistical significance <.001 (1-tailed).
[a] Based on negative ranks.

5.4. Discussion

Test results for hypotheses 1–3, as depicted by Fig. 6, show negative structure correlations for the canonical variate composed of the original dependent variables (note that a perfect negative correlation is −1). Therefore, as the independent variables $X13$, $X7$, $X5$, $X2$, and $X6$ increase, the dependent variables of predictability, observability, and usability decrease.

The statistical test results (Table 7) for hypotheses 4–6 indicate that treatment A was perceived by subjects as more complex than treatment B. These results were statistically significant given that the p-values exceeded the statistical significance level of 0.05.

In summary, the null hypotheses 1–6 were all rejected given the statistical significance of test results and the steps taken to guard internal validity.

5.5. Implications

A methodology for the quantification of NAT complexities for system-level requirements was presented. Early quantification of NAT complexities impacting safety could help system designers identify, analyze, and mitigate safety-related system complexities before they are propagated to subsequent life-cycle phases. Safety is an emergent property of the system, so safety must be addressed at the system level as done by this research. This is in contrast a safety approach that quantifies attributes of the software subsystem. This does not address the system directly and also would take place much later in the system life cycle when the software is already written. Thus, system modifications would be more difficult and costly to correct.

6. Conclusions

This work is a promising and significant step in meeting the research objective: to operationalize NAT for the system-level requirements of safety-related computer systems.

The research objective was partially realized. This claim is qualified as partial because the research was limited to one system and 32 test subjects; more empirical research is needed to establish external validity. Two arguments support this qualified claim. First, there was a statistically significant, negative correlation between five NAT metrics of complexity and the externally visible system attributes of predictability,

observability, and usability. Secondly, each of the specific aims for operationalizing NAT was realized.

- *Specific aim 1*. Identify a formal modeling method for system requirements which will afford quantification of NAT attributes.
  The SCR models and simulations successfully served the research needs for modeling, simulating, and analyzing human-computer interactions in the context of NAT. The SCR dependency graphs accommodated multiple levels of system abstraction and the multiple projections needed for specific aim 3. Secondly, the SCR toolset successfully supported simulation of our model. Subjects were able to quickly learn (in about 10 min) to run the simulation and effectively understand the simulation such that useful data were collected.
- *Specific aim 2*. Identify the NAT attributes to be operationalized with respect to system requirements.
  A process of deduction enabled us to ascertain that three of 13 NAT attributes can be observed in SCR dependency models of system requirements. Our premise was that NAT attributes could be generalized to linearity. Complex systems are nonlinear; simple systems are linear. From this premise, our reasoning led us to identify three NAT attributes to operationalize: interconnectivity, common-mode connections, and multiple control parameters.
- *Specific aim 3*. Identify potential metrics for each NAT attribute to be operationalized.
  This aim was satisfied as evidenced by 15 metrics proposed to measure or indicate the three NAT attributes from specific aim 2. We infer a degree of validity to the proposed metrics because our selection process addressed the multidimensional aspects of complexity by using multiple system abstractions and perspectives to obtain our metrics.
- *Specific aim 4*. Identify the metrics that are useful measures of NAT complexity. Analysis results showed that five out of the 15 proposed metrics had structure correlations exceeding .25, standard errors of less than .10, and statistically significant confidence intervals.

6.1. Limitations

Limitations of this research are as follows:

- *Predictive limitations*. The research did not develop mathematical models and inference procedures to identify and assign a probability to future outcomes; one thus cannot make outcome predictions based solely on the metric values.
- *Limited subject diversity*. The data from our volunteer subject characterizations indicates a relatively homogenous group of people. This can potentially threaten external validity with respect to generalizations to other populations. We infer that it was more difficult to elicit negative subject perceptions of system predictability, observability, and usability (the dependent variables) because the typical subject was an engineer with considerable analytical abilities and experiences with technical systems.
- *Unknown external validity*. The resulting set of independent variables $X13$, $X7$, $X5$, $X2$, and $X6$ and the rejection of the null hypotheses were based on statistically significant test results specific to the data set. It is not known if these independent variables are

useful for other systems, or if the same inferences concerning the six research hypotheses pertain to other systems.

Appendix

This appendix contains a portion of the subject questionnaire. The questions are for scenario 1, treatment B. All scenarios and treatments had identical questions.

System predictability

3.1 What is your *initial* opinion of the system's behavior?

3.1.1 confusing — understandable

1 2 3 4 5

3.1.2 unpredictable — predictable

1 2 3 4 5

3.1.3 unstable — stable

1 2 3 4 5

3.2 How difficult is anticipating *the system's* output or behavior?

difficult — easy

1 2 3 4 5

System observability

3.3 Does the system keep you informed about its status or state?

3.3.1 never — always

1 2 3 4 5

3.3.2 inappropriately — appropriately

1 2 3 4 5

3.4 Recognizing a change in the system's status is

difficult — easy

1 2 3 4 5

3.5 Understanding the meaning or implications of a change in system's status is

difficult — easy

1 2 3 4 5

3.6 Recognizing changes in the display information is

difficult — easy

1 2 3 4 5

System usability

3.7 The ability to find information is

difficult — easy

1 2 3 4 5

3.8 Can the scenario be performed in a straight-forward manner?

never — always

1 2 3 4 5

3.9 Rate the scenario's complexity.

high — low

1 2 3 4 5

3.10 Please write any comments. You may use the back of this page.

References

Cliff, N., 1987. Analyzing Multivariate Data. Harcourt Brace Jovanovich, San Diego, CA.

Coskun, E., Grabowski, M., 2001. An interdisciplinary model of complexity in embedded intelligent real-time systems. Information and Software Technology 43, 527–537.

Efron, G., Tibshirani, R., 1993. An Introduction to the Bootstrap. Chapman and Hall, New York.

Heitmeyer, C., Bharadwaj, R., 2000. Applying the SCR requirements method to the light control case study. Requirements Engineering. Journal of Universal Computer Science 6 (7) (special issue).

Heitmeyer, C., Kirby, J., Labaw, B., Bharadwaj, R., 1998. SCR*: A toolset for specifying and analyzing software requirements. In: Proceedings of the 10th Annual Conference, Computer-Aided Verification.

Hopkins, A., 1999. The limits of normal accident theory. Safety Science 32, 93–102.

Human Computer Interaction Laboratory. QUIS Webpage (http://www.cs.umd.edu/hcil/quis/) viewed: 22 April 2004.

Human Factors Research Group. SUMI Webpage (http://www.ucc.ie/hfrg/questionnaires/sumi/) viewed: 22 April 2004.

Jones, B., Kenward, M.G., 1998. Design and Analysis of Cross-Over Trials. Chapman and Hall, New York.

Kronenburg, M., Peper, C., 2000. Application of the FOREST approach to the Light Control Case Study. Journal of Universal Computer Science (Special Issue on Requirements Engineering) 6 (7).

Leveson, N., 2004. A new accident model for engineering safer systems. Safety Science 42, 237–270.

Littlewood, B., Strigini, L., 1992. The risks of software. Scientific American(November), 62–67.

MacKenzie, D., 1994. Computer-related accidental death: an empirical exploration. Science and Public Policy 21, 233–248.

Neumann, P.G., 1995. Computer Related Risks. ACM Press. Addison Wesley Publishing Co., New York.

Perrow, C., 1999. Normal Accidents: Living with High-Risk Technologies. Princeton University Press, Princeton.

Queins, S., Zimmerman, G., Becker, M., Kronengurg, M., Peper, C., Merz, R., Schafer, J., 2000. The light control case study: problem description. Journal of Universal Computer Science (Special Issue on Requirements Engineering) 6 (7).

Sammarco, J.J., 2003. Addressing the safety of programmable electronic mining systems: lessons learned. In: Proceedings of the 2002 IEEE Industry Applications Conference, 37th IAS Annual Meeting, Pittsburgh, PA.

Shafto, M.G., Degani, A., Kirlik, A., 1997. A canonical correlation analysis of data on human–automation interaction. In: Proceedings of the 41st Annual Meeting of the Human Factors and Ergonomics Society. Human Factors and Ergonomics Society, Albuquerque, NM.

Wolf, F.G., 2000. Normal Accidents and Petroleum Refining: A Test of the Theory. Doctoral dissertation, Nova Southeastern University.