MIT-AM-19-194

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Assessing Vulnerabilities in Model-Centric Acquisition Programs: Phase 2

1 August 2019

Dr. Donna H. Rhodes

Mr. Jack Reid

Massachusetts Institute of Technology

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. HQ0034-18-1-0013. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School

Abstract

Digital transformation changes how systems are acquired and developed through model-centric acquisition approaches and digital engineering practices and toolsets. Enterprises face new challenges in this transformation, including emergent vulnerabilities within digital engineering environments. While vulnerability analysis of products and systems is standard practice, examining vulnerabilities within the enterprise itself is less common. This report presents findings and results of a second phase of research on uncovering cascading vulnerabilities as related to digital engineering practice and supporting environments, taking a special focus on cybersecurity-related vulnerabilities.

The approach applies Cause-Effect Mapping (CEM) in vulnerability assessment as a means to better enable program leaders to anticipate and respond to vulnerabilities within the enterprise. With CEM, vulnerabilities are described using causal chains, where an external trigger initiates cascading intermediary events that leads to a terminal event. Interventions can be applied to break the causal chain in appropriate places.

Phase 1 investigated uncertainties and related decisions that may lead to vulnerabilities in model-centric acquisition programs. An initial reference model for aiding program managers in detecting, assessing and mitigating vulnerabilities as related to the program's model-centric engineering practices and environment was developed. A step-wise process was defined for applying the reference model. This Phase 2 research further developed and tested the vulnerability assessment reference model and process, resulting in a baseline Reference CEM. Cybersecurity vulnerabilities are of particular concern given digital transformation and increasing threat actors. Accordingly, a deeper investigation of cybersecurity within programs and enterprises was performed given its importance and urgency.

This research is responsive to the 2018 DoD Digital Engineering Strategy, which calls for enterprises to mitigate cyber risks and secure digital engineering environments against attacks from internal and external threats, mitigate known vulnerabilities that

present high risk to DoD networks and data, and to mitigate risk posed by collaboration and access to vast amount of information in models. The technical approach for the research began with literature survey and gathering results of research studies of relevance, recent workshop findings, and related work on vulnerability assessment that may have implications for this work. This informed refinement of the reference model and process, which were further validated in this phase. Dynamic models were examined as a means to understand the cascading vulnerabilities and potential intervention options. A concept for an interactive demonstration prototype was also explored.

Phase 2 research results are: (1) Reference CEM and process to guide vulnerability assessment, (2) empirically-grounded cybersecurity vulnerabilities related to model-centric acquisition programs and enterprises, and (3) initial concept for an assessment prototype.

Keywords: model-centric, vulnerabilities, cause-effect mapping, cybersecurity, interventions

About the Author

Dr. Donna H. Rhodes – Donna H. Rhodes is a principal research scientist at the Massachusetts Institute of Technology, and director of the Systems Engineering Advancement Research Initiative (SEAri). Dr. Rhodes conducts research on innovative approaches and methods for architecting complex systems and enterprises, designing for uncertain futures, and human-model interaction. Previously, she held senior management positions at IBM, Lockheed Martin, and Lucent. Dr. Rhodes is a Past President and Fellow of the International Council on Systems Engineering (INCOSE), and INCOSE Founders Award recipient. She received her Ph.D. in Systems Science from T.J. Watson School of Engineering at Binghamton University.

Massachusetts Institute of Technology
77 Massachusetts Avenue, E17-361
Cambridge, MA 02139
Tel: (617)-324-0473
rhodes@mit.edu

Mr. Jack Reid – Jack Reid is a graduate student with the Space Enabled Research Group at the Massachusetts Institute of Technology. Reid is currently a doctoral student at MIT with research interests concerning the design and management of complex sociotechnical systems, particularly with regard to the anticipation of emergent and cascading behavior. While a master's student, he was a research assistant in the Systems Engineering Advancement Research Initiative (SEAri), performing research on vulnerability assessment methods, model-centric enterprises, and complexity and emergence. He received an MS in both Aeronautics & Astronautics and Technology & Policy at MIT.

Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: 512-350-5261
jackreid@mit.edu

THIS PAGE INTENTIONALLY LEFT BLANK

MIT-AM-19-194

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Assessing Vulnerabilities in Model-Centric Acquisition Programs: Phase 2

1 August 2019

Dr. Donna H. Rhodes

Mr. Jack Reid

Massachusetts Institute of Technology

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. HQ0034-18-1-0013. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

List of Figures

THIS PAGE LEFT INTENTIONALLY BLANK

Executive Summary

Digital transformation changes how systems are acquired and developed through model-centric acquisition approaches and model-based (digital) engineering practices and toolsets. Enterprises face new challenges in this transformation, including potential for emergent vulnerabilities within digital engineering environments. While vulnerability analysis of products and systems is standard practice, examining vulnerabilities within the enterprise itself is less common. This report presents findings and results of a second phase of research on uncovering cascading vulnerabilities as related to digital engineering practice and supporting environments, taking a special focus on cybersecurity-related vulnerabilities. The approach applies Cause-Effect Mapping (CEM) in vulnerability assessment as a means to better enable program leaders to anticipate and respond to vulnerabilities within the enterprise.

Phase 1 of this research investigated uncertainties and related decisions that may lead to potential vulnerabilities in model-centric acquisition programs. An initial reference model for aiding program managers in detecting and assessing vulnerabilities as related to the program's model-centric engineering practices and environment was developed. The research defined a step-wise process for applying the reference model in assessing and mitigating model-centric vulnerabilities. (Rhodes & Reid, 2018, Reid & Rhodes, 2018a, Reid & Rhodes, 2018b). This Phase 2 research has further developed and tested the vulnerability assessment process and model, resulting in the baseline *Reference CEM*. Additionally, research has included a deeper investigation of cybersecurity within programs and enterprises given its importance and urgency.

Cybersecurity vulnerabilities are of particular concern given digital transformation and increasing threat actors, making vulnerability assessment essential throughout acquisition program lifecycle (Rhodes & Reid, 2019). Not only are end-systems highly vulnerable to cyber threats, so too are their enabling environments and digital assets. Early detection of vulnerabilities, and possible

interventions, can mitigate potential disruptions and failures. This research is responsive to many of the imperatives of the newly released DoD Digital Engineering Strategy. The strategy calls for enterprises to mitigate cyber risks and secure digital engineering environments against attacks from internal and external threats, mitigate known vulnerabilities that present high risk to DoD networks and data, and to mitigate risk posed by collaboration and access to vast amount of information in models.

This research seeks to provide program leaders with the means to identify vulnerabilities within model-centric programs and enterprises, and to determine where interventions can most effectively be taken. The technical approach for the research began with literature survey and gathering results of past research studies of relevance, recent workshop findings, and related work on vulnerability assessment that may have implications for this work. This informed refinement of the reference model and process, which was further validated in this phase. Dynamic models were examined as a means to understand the cascading vulnerabilities and potential intervention options. A concept for an interactive demonstration prototype was also explored.

Phase 2 research results are: (1) Reference CEM and process to guide vulnerability assessment, (2) empirically-grounded cybersecurity vulnerabilities related to model-centric acquisition programs, and (3) initial concept for an assessment prototype.

The research resulted in one published paper (Rhodes & Reid, 2019) for the 2019 Naval Postgraduate School Acquisition Research Symposium, presented in a panel session, and conference presentation (Rhodes & Reid) given at the 2018 NDIA Systems Engineering Conference.

Background

Digital transformation changes how systems are acquired and developed through model-centric acquisition approaches and model-based (digital) engineering practices and toolsets. While offering great benefit, new challenges arise from both technological and non-technical dimensions. This drives the need to examine and address vulnerabilities not only for products and systems, but also for the model-centric (digital engineering) environments necessary for their acquisition and development. Foundational research on cause-effect mapping for vulnerability assessment has included commercial and defense sectors. This NPS research is primarily focused on the defense sector.

Vulnerability, Risk, and Hazard Analysis

Vulnerability, risk and hazards analysis are three interrelated terms that have different definitions depending on the field and on the method of analysis. In this research, a *hazard* refers to a system or environmental state that has the potential to disrupt the system. Examples include the existence of an iceberg at sea and tired operators. Hazards may not result in system failure, partly depending on the design of the system. A *vulnerability* is the means by which the hazard might disrupt the system. It is through the vulnerability that the system is susceptible to the hazard. Vulnerabilities are best expressed as the causal series of events connecting a hazard to system failure. This is a generalization of common, field-specific usage of the term. *Risk* is a measure of the probability of a system disruption and the consequences of that disruption. Sometimes risk is instead expressed as a multiplication of likelihood and consequence.

Numerous methods for analyzing vulnerabilities, risks, and hazards exist. Common means of analysis include Fault-Tree Analysis (FTA), Failure Modes, Effects, and Criticality Analysis (FMECA, though sometimes reduced to FMEA), Systems Theoretic Process Analysis (STPA), and Event Tree Analysis (ETA). A discussion and comparison of these methods can be found in Reid and Rhodes (2018).

Cause-Effect Mapping (CEM)

Cause-Effect Mapping (CEM) consists of a mapping of causal chains that connect an exogenous hazard to a system degradation or failure, termed a *terminal event*. Each chain represents a vulnerability, sometimes called a *vulnerability chain* in order to emphasize that vulnerabilities are not discrete events. Terminal events are broadly defined and include any form of value loss. Similar to fault tree analysis, CEM is easily read in either direction, but it also allows for the simultaneous consideration of multiple failures and multiple hazards. The hazards are external to control of the defined user, and are thus sometimes called *external triggers*. An *intermediary event* is any unintended state change of a system's form or operations which could jeopardize value delivery of the program.

CEM has previously been applied in a case study of a Maritime Security System of Systems (Mekdeci, et al., 2012) and in a supply chain case (Rovito & Rhodes, 2016). More recently, phase 1 of this research developed a preliminary reference model for use by program managers to assess enterprise-level vulnerabilities in the digital engineering/model-centric environment (Reid & Rhodes, 2018a). Potential use cases are discussed in Reid & Rhodes (2018a). Key benefits include increased understanding of the causal path and the interrelationships between vulnerabilities.

Terminology

The following terms are defined as used in this research project.

acquisition	Conceptualization, initiation, design, development, test, contracting, production, deployment, integrated product support (IPS), modification, and disposal of weapons and other systems, supplies, or services (including construction to satisfy DoD needs, intended for use in, or in support of, military missions.
analysis	An evaluation, quantitative and/or qualitative; synonymous with assessment.
assessment	Synonymous with analysis.
causal chain	A series of events, with each event causing or being an integral part of the cause, or the next "link" in the chain.
causal factor	Any aspect of the system which, when removed or changed, is likely to reduce the occurrence of emergent behavior, or, when induced, is likely to increase the occurrence of emergent behavior.
classification	A generic term for sorting a set by some defined metric, either quantitative or qualitative. Includes both taxonomies and typologies.
cyber attack	An attack, via cyberspace, targeting an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information.
cyberspace	An interdependent network of information systems infrastructures (e.g., internet, computer systems, software, embedded processors and controllers, intercommunications, etc.).
dynamic model	A model that represents the behavior over time (e.g., network model, system dynamics model).
external trigger	A hazard external from the perspective of a defined user; sometimes referred to as a spontaneous event.
hazard	A system or environmental state that has the potential to disrupt the system.
intermediary event	Any unintended state change of a system's form or operations which could jeopardize value delivery of the program and is situated along a causal chain connecting an external trigger to a terminal event.
intervention	A means of disrupting or mitigating a vulnerability chain.

intervention point	The point in a causal chain where disrupting/mitigating occurs.
model-centric engineering (digital engineering)	An overarching digital engineering approach that integrates different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle.
program	A directed, funded effort that provides a new, improved, or continuing materiel, weapon or information system, or service capability in response to an approved need.
terminal event	Any form of system degradation or failure, or of value loss.
vulnerability	The means by which the hazard might disrupt the system.
vulnerability assessment framework	A framework for assessing and analyzing hazards, vulnerabilities, and risks in a comprehensive manner, as well as determining appropriate mitigations and countermeasures.
vulnerability chain	A conceptualization and representation of a vulnerability as a causal chain.

Literature Investigation

Phase 2 continued review of the most recent literature. Knowledge gathering was performed as relevant to model-centric enterprises, and cybersecurity within programs and enterprises.

Digital Engineering

The DoD Digital Engineering Strategy (DoD, 2018) establishes goals and focus areas for digital transformation. Digital engineering involves use of integrated models across disciplines, subsystems, lifecycle stages, and analyst groups. It uses models as "authoritative source of truth," to reduce document handoff and allow for more continuous evaluation. By collaborating through models, there is reduced communication time and rework in response to requirement changes. Most discussions of digital engineering, to date, focus on engineering practices and methods to overcome implementation difficulties. In any system, however, non-technical factors (human factors, business, and organizational) influence engineering effectiveness and model-centric decisions (German & Rhodes, 2017).

Current program leaders have significant experience with processes for acquiring and developing systems, and use this experience to identify and mitigate vulnerabilities. Minimal experience exists with digital engineering practice and model-centric supporting environments, however. This situation, coupled with the increased model integration and model longevity, means that emergent uncertainties (policy change, budget cuts, disruptive technologies, threats, changing demographics, etc.) and related programmatic decisions (e.g., staff cuts, reduced training hours) may lead to cascading vulnerabilities within digital engineering enterprises, potentially jeopardizing program success (Reid & Rhodes, 2018a). New practices and enablers are needed to assist enterprise leaders in identifying vulnerabilities within the digital engineering environment, and to determine where interventions can most effectively be taken.

Model-Centric Environments

Model-centric environments have many elements, including computing infrastructure, networks, software tools, models, data sets, data storage, and human actors. These environments may come under attack from internal and/or external threats. Some of these elements exist in traditional engineering, but some are new or changed under digital engineering practice (Reid & Rhodes, 2016).

New modes of collaboration through models and data are emerging (DoD, 2018). The quantity of and types of models, digital artifacts, and data has greatly increased. The collaboration between the many enterprises involved through digital engineering (government agencies, contractors, suppliers, etc.) results in significant increases in data flowing across networks. As new toolsets are introduced into enterprise, there are potential risks related to how proficient the workforce is in using these tools and whether there are sufficient controls in place in the management of the digital artifacts produced, as well as the overall supporting infrastructure. The DoD Digital Engineering Strategy (2018) calls for the mitigation of these risks and vulnerabilities (Figure 1).

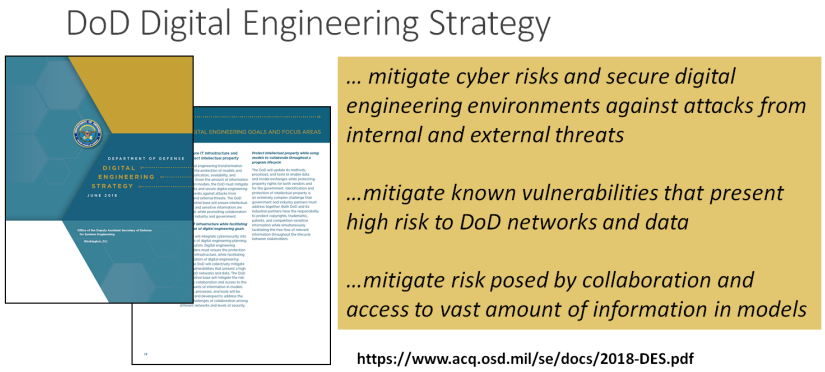


Figure 1. Digital Engineering Strategy - mitigation of risks and vulnerabilities (DoD, 2018).

Cybersecurity

Digital engineering with its focus on digitization, model integration, and collaboration, naturally impacts the potential cybersecurity vulnerability of an enterprise. A *vulnerability* is the means by which the hazard might disrupt the

system, thus it is through the vulnerability that the system is susceptible to the hazard. Vulnerabilities are effectively expressed as the causal series of events connecting a hazard to system failure. This is a generalization of common, field-specific usage of the term. MITRE's Common Vulnerabilities and Exposures (CVE) database defines a vulnerability as "a weakness in the computational logic (e.g., code) found in software and some hardware components (e.g., firmware) that, when exploited, results in a negative impact to confidentiality, integrity, OR availability" (The MITRE Corporation, 2015). In this definition, the same components can be seen: some structural means or "weakness," that can result in system disruption or "negative impact" if a hazard is present or the vulnerability is "exploited." For instance, the infamous Spectre security vulnerability is described by CVE as "Systems with microprocessors utilizing speculative execution and branch prediction may allow unauthorized disclosure of information to an attacker with local user access via a side-channel analysis" (The MITRE Corporation, 2017), providing a concise description of the hazard (an attacker), the means (side-channel analysis using speculative execution and branch prediction), and the disruption (unauthorized disclosure of information).

Risk and vulnerability assessment methods have not failed to adapt to novel cybersecurity concerns. The aforementioned CVE database has been public since 1999. Quality Assurance testing (essentially the verification and validation of software) has been around since the beginning of commercial software. Software penetration testing (where security experts intentionally seek to break a software product) has been the industry norm for more than a decade (Arkin, Stender, & McGraw, 2005). Black-box mutational fuzzing and concolic execution are being used to automatically test for certain types of software vulnerabilities (Schwarz, 2018). Formal verification tools, initially limited to pure software domains such as cryptography (Meadows, 1994), has been rapidly advancing and finding applications in hardware (Kern & Greenstreet, 1999) and business processes (Morimoto, 2008), as well as fields that straddle the software-hardware-environment boundaries (Kamali, et al., 2016). Beyond these specific testing methods, assessment frameworks have progressed as well. System-Theoretic Process Analysis (STPA)

has adjusted, adapted, and been applied to handle cybersecurity vulnerabilities associated with additive manufacturing (Pope & Yampolskiy, 2016), Internet of Things (Pope, 2017), Air Operations (Young, 2013), and Mission Operations (Young & Porada, 2017). More recently, there have been efforts to combine compiler technology with STPA to automatically detect vulnerabilities in software-controlled systems (Pope, 2018).

While cybersecurity vulnerabilities in operational systems remain alarming common, from the trivial (Hanselman, 2012) to the critical (Gressin, 2017), there is some evidence that software is becoming more secure, at least in terms of defects per line of equivalent source code (Pope, 2017). In many cases, however, the acquisition or development process itself needs to be protected from outside threats and endogenous failures. Be it military information or technology-related trade secrets, there is real value in attempting to penetrate much earlier in the life cycle in order to either steal secrets (Hanna, Smythe, & Martin, 2018; Raymond, 2017) or to disrupt production (Statt, 2018).

Defense acquisition programs have already instituted a variety of means of ensuring the security of their work. Some of these means were originally instituted to address other forms of threats but have turned out to be effective in addressing cybersecurity as well. These methods include relying on the security clearance process, the use of Sensitive Compartmented Information Facilities (SCIFs), restrictions on the use of media storage devices, separate networks such as SIPRNet and NIPRNet that are isolated or semi-isolated from the internet, and general compartmentalization of critical information.

Unfortunately, some of these historically successful methods may be in conflict with the implementations within model-centric enterprises. For instance, the use of SCIFs has been quite successful in preventing unauthorized access to data. The typical use of a SCIF in design, where a small number of engineers work on a task isolated from the outside world, is not directly compatible with an MCE environment structured around model integration and collaboration across teams and locations. While this problem has been previously considered and ways to mitigate this conflict have been proposed (e.g., Reid & Rhodes, 2016), no silver bullet to resolving these

tensions exists and it is likely that the increased use of MCE will result in both the exacerbation of current vulnerabilities and the creation of new ones. Furthermore, most means of assessing such vulnerabilities are aimed at assisting software and systems engineers to identify and remove cybersecurity vulnerabilities from the end-system. New methods for enabling program leaders to perform cybersecurity assessments of their enterprise and engineering environment are needed.

THIS PAGE LEFT INTENTIONALLY BLANK

Methodology

The primary objective of this research, over the two phases of effort, has been to formulate an approach and reference map for vulnerability assessment of digital (model-centric) engineering for use by program leadership in DoD enterprises. In Phase 1, the initial data and knowledge gathering was performed and analyzed, which informed an effort to apply a Cause-Effect Mapping (CEM) approach to vulnerability assessment of model-centric programs. The Phase 1 research produced an initial version of a reference map, which was then tested for usability.

In this Phase 2 of the research, additional literature review and knowledge gathered was used as a basis for refinement of the Reference CEM. Given a special focus on cybersecurity vulnerabilities in this phase of the research, the portion of the Reference CEM that pertained to external triggers and terminal events related to cybersecurity was elaborated. Publically available information and expert knowledge was used to validate it, as well as to investigate the potential interventions that could be used to break the vulnerability chains. Fundamental to the methodology is the construct of a vulnerability as a causal chain.

Vulnerabilities as Causal Chains

A *vulnerability* is the means by which a hazard might disrupt the system and/or enterprise. Vulnerabilities are effectively expressed as the causal series of events connecting a hazard to the system and/or failure that results. A *casual chain* can be defined as*:* a series of events, with each event causing or being an integral part of the cause, or the next "link" in the chain. A *hazard* ("spontaneous event") is a system or environmental state that has the potential to disrupt the system. A *vulnerability* is defined as causal means by which one or more hazards results in the system disruption / value loss. *Terminal events* are broadly defined and include any form of value loss. Accordingly, a *vulnerability chain* is defined as a conceptualization and representation of vulnerability as a causal chain, emphasizing that vulnerabilities are not discrete events.

Categories of External Triggers.

Research on engineering practice and program environments has informed three categories of external triggers. This should be viewed as work in progress and possibly customized to the situation. Three specific external trigger categories are used in the methodology in this work. It likely that one or more additional categories may be specified as this work evolves.

Force Majeure: This is a general term for an event that is the result of actions beyond the possibility of the program enterprise (not just the program manager) to influence. Thus it includes both malicious action and general, unforeseeable events such as Technological Change.

Policy: An event that is the result of intentional decisions made at the organizational or enterprise level. In the case of a government-run program, this includes oversight from Congress and the general public. Non-government organizations may still be impacted indirectly by such oversight, but their proximal triggering event would be different.

Private Sector: In a prior supply chain research effort, this category was broadly defined as “Economic/Resource”. In this research project, in order to distinguish the influence within the government section, this category is termed “Private Sector”. The external trigger is any event that is the result of the actions of one or more private-sector firms outside the program enterprise.

Extrapolation of Vulnerabilities

The Reference CEM was generated through a combination of methods. At the start of this research there was little literature on programmatic vulnerabilities posed by MCE. Most negative case studies, that is those that depict failures (Software Engineering Institute, 2007), and lessons learned databases (NASA Office of the Chief Engineer, 1994) are from prior to the rise of MCE and thus deal with general vulnerabilities. Over the two phase of research, there have been additional experiences and findings related to model-centric program execution. The existing case studies that directly deal with MCE tend to be largely positive, likely due to the

rising popularity of the paradigm (Conigliaro, Kerzhner, & Paredis, 2009; Maley & Long, 2005; Martz & Neu, 2008). As a result, extrapolations from extant vulnerabilities had to be made, along with hypothetical inversions of the positive instances of MCE. Additional vulnerabilities were contributed through sessions with graduate students with practitioner experience, and these where supplemented and confirmed using expert interviews.

Cause-Effect Mapping for Vulnerability Assessment

Cause-Effect Mapping (CEM) has been demonstrated as a useful approach to vulnerability analysis for programs and enterprises (Reid & Rhodes, 2018a, Reid & Ross, 2018b). An example CEM for a supply chain case vulnerability assessment (Rovito, 2016) is shown in Figure 2. The hazards are external to the perspective of the defined user, and in this method are called *external triggers*. An *intermediary event* is any unintended state change of a system's form or operations which could jeopardize value delivery of the program and/or enterprise. *Interventions* are actions that eliminate or mitigate a vulnerability to break the causal chain. The text boxes on the left side are the exogenous factors, or external triggers, beyond the control of the program leader. The text boxes on the right side are the terminal events that could result. The unshaded boxes are the intermediary events (or conditions). The directed arrows show pathways from the external trigger, cascading to intermediary events, and the resulting terminal event.

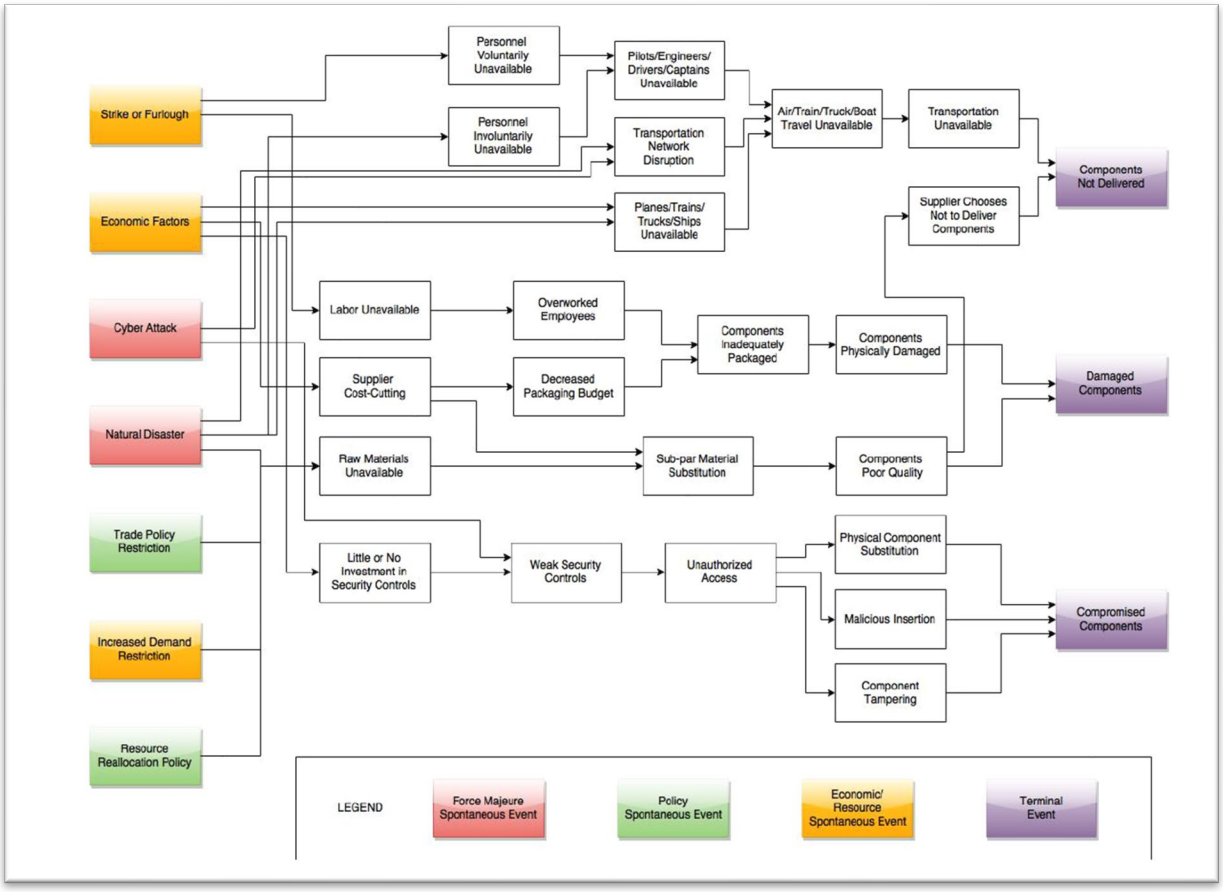


Figure 2. Vulnerability Assessment CEM of a supply chain. (Rovito & Rhodes, 2016)

A CEM is created for a specific class of decision-maker (e.g., program manager). The external triggers (referred to in Figure 2 as “spontaneous events”) are exogenous from the point of view of the decision-maker for which the CEM was constructed. In this way, the cause-effect mapping approach avoids “blaming someone else” by making all hazards exogenous. The decision-maker only has control over the intermediary events. While not necessarily at fault for any of the vulnerabilities, the decision maker has the responsibility and authority to choose if, and how, to address these.

As shown in Figure 3, a causal chain may have multiple points for breaking the chain, for instance to correct weak security controls and/or prevent unauthorized access. The first might be a policy/process intervention and the latter might be a

technology intervention. The decision to execute one/both of the interventions will depend upon unique factors, such as cost to implement, color of money available, specifics of the situation, etc.

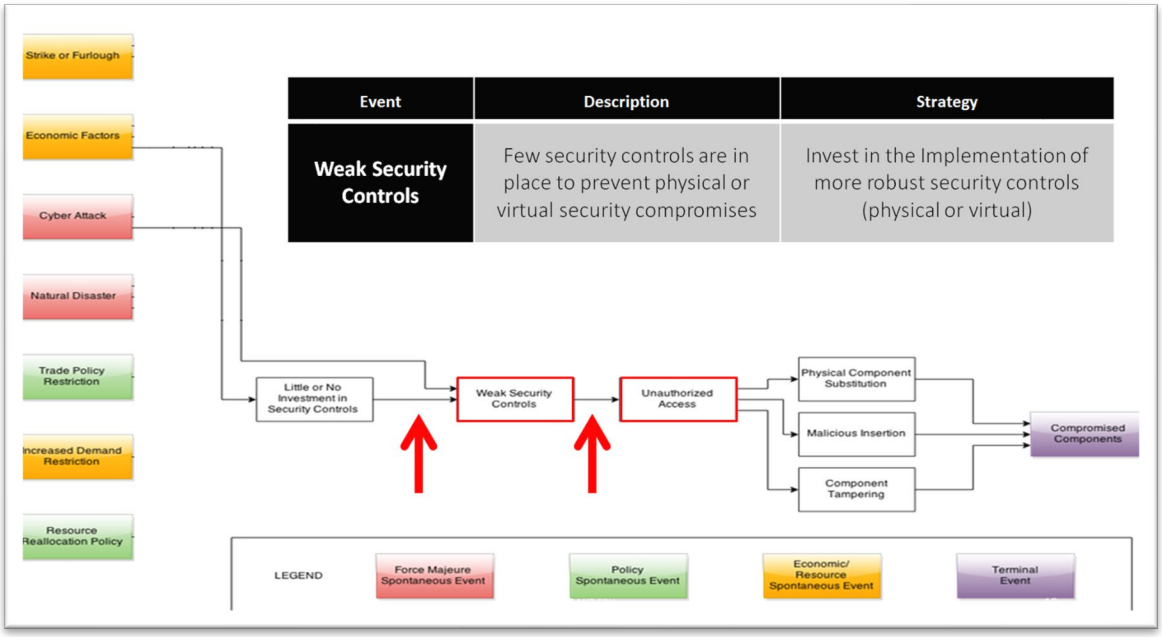


Figure 3. Example alternative placements of an intervention in causal chain.

Reid & Rhodes (2018a) discusses the basic steps to create a new CEM (steps are not application specific), which derives from earlier work (Rovito & Rhodes, 2016). The stakeholder (e.g., program decision maker) generates the CEM (or tailors a Reference CEM) by listing potential hazards posed to the program, and then traces the consequences of each of these hazards through the intermediary events to the final terminal events. The process is then done in reverse: taking the terminal events, adding in any that are still missing, and working backwards on how these might come about. The causal connections between each intermediary event are examined to see if there are any additional connections not previously noticed. Multiple additional sources (lessons learned databases, case studies, and other experts) are consulted to generate additional hazards, intermediary events, causal connections, and interventions, as well as to verify existing ones. It is envisioned that any of these steps can take place either formally, using automated tools to

enumerate possible vulnerabilities, or informally, relying upon the stakeholder's own experience. CEM is fundamentally a qualitative analysis method, though it has potential to be adapted into a more quantitative form, by specifying probabilities of transition to each intermediary (Reid & Rhodes, 2018a).

Findings

This research provides several contributions to the body of knowledge concerning vulnerability assessment and the assessment of programmatic/enterprise model-centric engineering vulnerabilities in particular. It advances the conceptualization of vulnerabilities as causal chains and uses results from usability testing to demonstrate the effectiveness of this conceptualization. The causal chain concept also enables new means of sorting and categorizing vulnerabilities, enabling the identification of effective interventions and heuristics.

The subsections below summarize specific Phase 2 findings, including:

- Reference CEM for model-centric programs and enterprises
- Cybersecurity vulnerabilities related to model-centric programs and enterprises
- Preliminary investigation of dynamic methods and concept for an interactive prototype.

Reference CEM for Model-Centric Programs and Enterprises

CEM provides an effective way to describe cascading vulnerabilities within a digital engineering enterprise. Figure 4 shows the Reference CEM generated in this research. The Reference CEM has been developed based on the research over the two phases of effort. It integrates the findings resulting from the investigation and analysis of vulnerabilities and interventions.

For improved readability of the results, see Appendix A, which has a breakdown of the Reference CEM in four sections, along with the detailed information on the associated external triggers, terminal events, and intermediary events.

The subsection that follows discusses the external triggers, shown in the green, orange and red boxes in the Reference CEM (left side). The next subsection discusses the intervention points that are marked with numbered circles on the Reference CEM, with the corresponding detail.

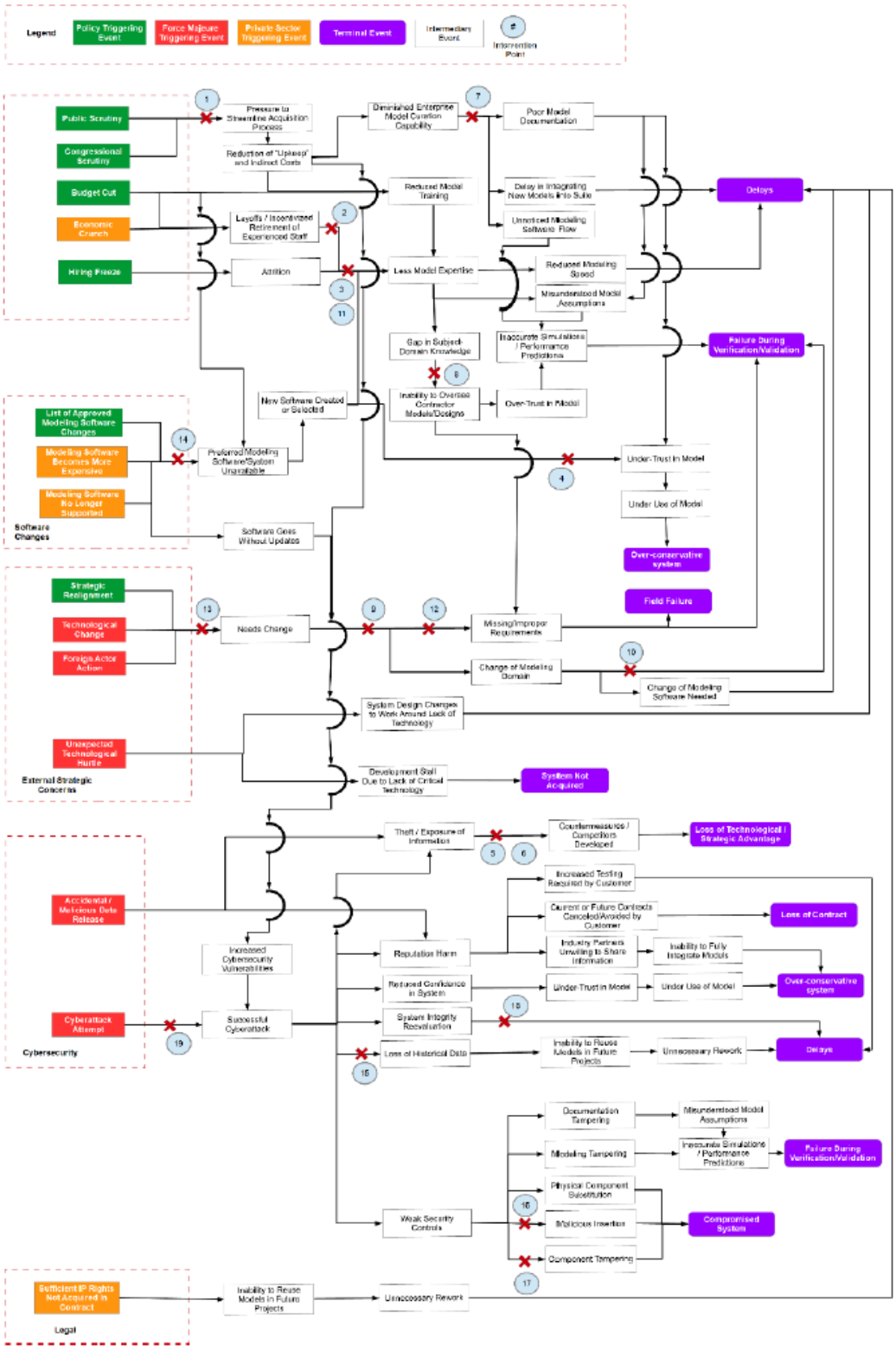


Figure 4. Reference CEM for Vulnerability Assessment.

Vulnerability Chain Illustrative Example

Figure 5 shows a very simple example of a vulnerability chain, where an external trigger disrupts effectiveness of engineering activities, as triggered by increased cost of the commercial software used by the enterprise. This is illustrative of how a rather simple external change may cascade into interim impacts, and ultimately lead to a failure later in the program.

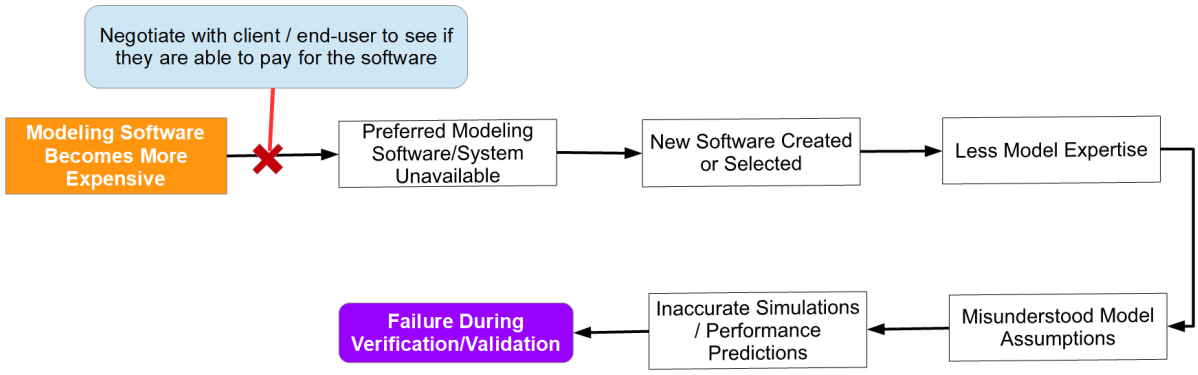


Figure 5. Example Vulnerability Chain with Intervention Action (in blue box)

Describing this as a vignette, the vulnerability is as follow:

> A particular piece of simulation software that your company has used on similar projects in the past is licensed from commercial software vendor. The license contract is up for renewal soon and the price goes up significantly. This could result in the preferred modeling software being unavailable for use in this program leading to the selection of an alternate software tool that the team has less (or no) experience with. Due to this lack of experience with the new software, assumptions underlying the model may be misunderstood by analysts and thus inaccurate simulation results generated. This may not be noticed until either verification or validation when the system or subsystem does not behave according to the predicted performance levels.

One identified intervention point is shown in the blue box in Figure 5. Executing this intervention would require that program leadership recognizes when the external trigger is imminent or occurring and act quickly to avoid loss of modeling capability. Alternately, there may be other points of intervention along the chain. While this analysis is quite simple, more sophisticated applications of graph theory

and probabilistic modeling can be conducted using a well-developed Reference CEM. For instance, if probabilities, likelihoods, or time scales of each event transition are known, techniques such as Markov Chain Modeling, Monte Carlo Analysis, and Bayesian Networks can be brought to bear, weighting each arc of the graph instead of treating them equally (Reid, 2018).

External Triggers

The research has defined fifteen external triggers that may lead to terminal events in model-centric programs and enterprises. This should be viewed as an incomplete list, and depending on the specific program and particular enterprise it would be expected to vary. The identified external triggers are:

Accidental/Malicious Data Release	Either accidentally or intentionally, some amount of sensitive data involving the program has been released to individuals or groups not cleared to access such information
Budget Cut	Current or projected funding for this program or for the enterprise as a whole is being reduced
Congressional Scrutiny	Congress has become increasingly concerned with the management or status of this program or of defense programs in general
Cyberattack Attempt	Some individual or group attempts to disrupt or surveil the program using a cyberattack
Economic Crunch	Reduced consumption, reduced willingness to lend, raising unemployment or other forms of economic recession or depression are occurring
Foreign Actor Action	A foreign actor has taken some significant action that impacts the intended use of system
Hiring Freeze	The program or enterprise has ceased all new hiring for some period of time
Approved Software List Changes	The organization maintains a list of software approved for use in programs and has changed (added/removed) certain software from this list

Modeling Software Becomes More Expensive	Some modeling software used by the program which is purchased or licensed from an external provider has become more expensive in the upcoming version/renewal
Modeling Software No Longer Supported	Maintainer/developer of in-use modeling software has ceased issuing updates and/or new versions
Public Scrutiny	The public and/or news media has become increasingly concerned with management of program or of defense programs in general
Strategic Realignment	The strategic interests of the client or other stakeholders in the program have changed
Sufficient IP/Data Rights Not in Contract	Sufficient intellectual property rights not acquired for system components, product, data.
Technological Change	A significant new technology has been developed or put into use that impacts program
Unexpected Technological Hurtle	During the program, some desired technology either is unexpectedly unavailable or is taking an unexpectedly long time to develop

Intervention Points and Actions

The research has identified nineteen intervention points and actions that may be used to break casual chains in model-centric programs and enterprises. This should be viewed as an incomplete list, and depending on the particular program and the enterprise, it would be expected that additional intervention points will be identified. The intervention points are shown below on a reference map with a numeric identifier (see Figure 6).

The CEM Reference Map in Figure 4 (and in Appendix A) includes nineteen interventions, as listed below.

Table 1. Intervention Points and Actions

Point #	Intervention Action
1	Initiate internal assessment and a PR strategy
2	Initiate various non-monetary benefits (e.g., 9/80 schedule) to retain employees
3	Seek to share resources and employees with other programs
4	Hire employees with prior experience with the new software
5	Compartmentalize sensitive information
6	Obfuscate sensitive data with false or misleading information
7	Create documentation and curation processes within the program
8	Institute handover periods to benefit from contractor expertise
9	Reevaluate the training regime and needed fields of expertise
10	Increase the amount of testing conducted
11	Increase use of contractors/consultants/former employees to maintain expertise level
12	Reevaluate the requirements with the client and other stakeholders
13	Design for modularity to minimize impact on system
14	Negotiate with client / end-user to see if they are able to pay for the software
15	Maintain isolated but readily accessible back-ups of data
16	Conduct reviews/comparisons of models between lifecycle stages
17	Use multiple, independent simulations or component checkers
18	Maintain isolated, independent backup equipment while primary equipment evaluated
19	Conduct regular “red-team” / penetration test exercises

Observations Regarding Intervention Points

Reid (2018) found that intervention points identified in the Reference CEM tend to be in the first half of the vulnerability chains, with several immediately after an external trigger. This suggests the need for monitoring for potential or imminent

external triggers and being ready to respond as soon as, or even in advance of, their manifestation.

The Reference CEM can be used to guide the attention to various vulnerabilities. For instance, it should be noted that within the “active modeling” set of intermediary events (inside the blue box of Figure 6) there are relatively few intervention points identified, despite the high number of vulnerability chains that pass through that section of the Reference CEM. The primary intervention point identified in that section, #7, is “Create documentation and curation processes within the program.”

This relative lack of intervention points may represent the unfamiliarity of program leaders with digital engineering processes and how to intervene in them. This suggests that further work would be useful in identifying potential interventions in this section of the map, and educating program leaders concerning their availability and use.

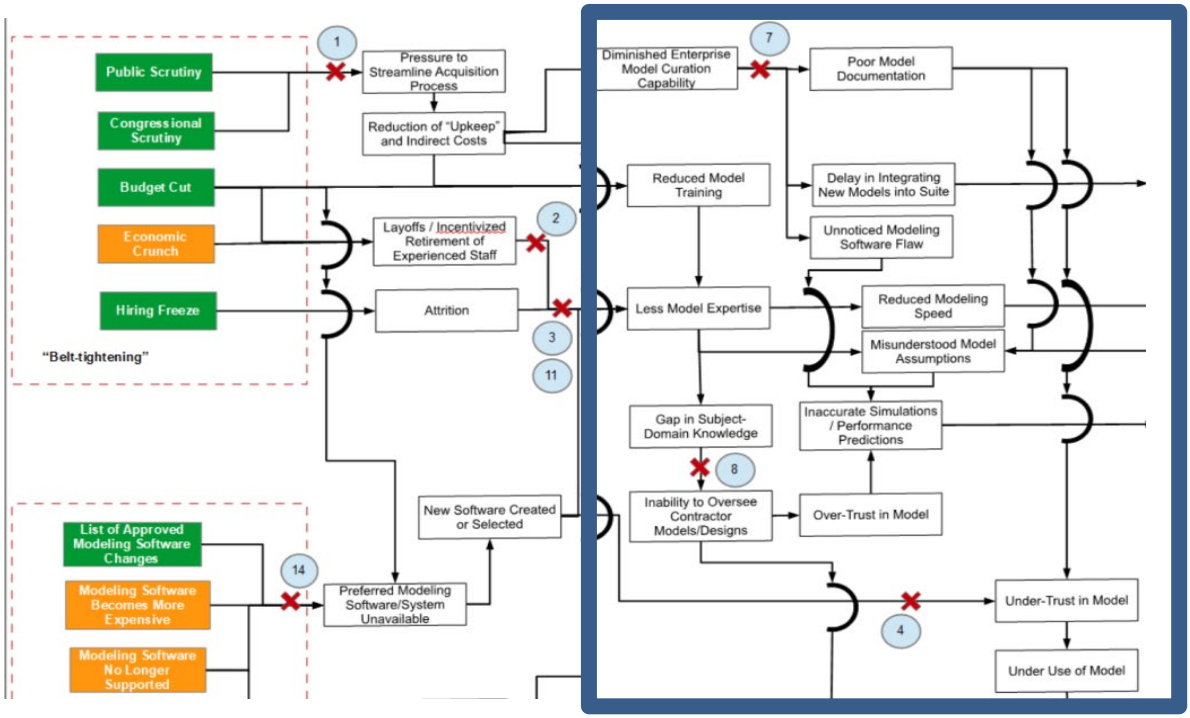


Figure 6. Excerpt of Reference CEM highlighting “active modeling” portion (Reid, 2018)

While this portion of the chain has one intervention identified, certain vulnerability chains have multiple intervention points identified at multiple stages. For instance, several of the vulnerability chains that pass through the *Needs Change* event have three intervention points each (and the others have at least two), as shown in Figure 4. According to Reid (2018), this suggests there may not be as much of a concern about these vulnerabilities, due to the multiple options of intervention available and the fact that several are positioned multiple events into the chain, giving significant time for response.

An experienced program leader will likely find some of the listed intervention points to be common sense. For instance, one of the interventions (#12) following the *Needs Change* event is "Reevaluate requirements with the client and other stakeholders." This degree of occasional obviousness is not unique to CEM, but is true of all vulnerability assessment techniques. The point of these techniques is not just to identify new vulnerabilities and interventions, but to consistently track and assess them so that all options are available.

It should be noted that the Reference CEM shown in this report does omit vulnerabilities and interventions that are entirely unchanged. For example, practices like the security clearance system and restricting the use of digital storage media will remain necessary, effective interventions that are not significantly impacted by model-centric environments. Some historically successful methods may be conflict with these environments, for example, the use of SCIFs has been quite successful in preventing unauthorized access to data. The typical use of a SCIF in design, where a small number of engineers work on a task isolated from the outside world, is not directly compatible with an model-centric environment structured around model integration and collaboration across teams and locations. While this problem has been previously considered and ways to mitigate this conflict have been proposed (e.g., Reid & Rhodes, 2016), tensions do exist and it is likely that the increased use of digital engineering will result in both the exacerbation of current vulnerabilities and the creation of new ones.

Cybersecurity Vulnerabilities

As the initial research progressed, the importance and urgency of considering the cybersecurity vulnerabilities shaped the second phase of study to focus more specifically on these. Literature review and interview-based research has provided useful insights. In the second phase of this research, interviews were conducted with systems engineers and program managers from a variety of fields, including defense, aerospace, manufacturing, and semiconductors (Reid, 2018). The interviews explored these program cybersecurity vulnerabilities in general, and in context of model-centric approaches.

Interviews and knowledge gathering revealed commonly cited issues, including:

- Cybersecurity needs to be thoroughly considered much earlier than it commonly is, preferably in the proposal generation stage.
- At present, cybersecurity assessment of the enterprise itself is often ad-hoc and minimally performed.
- Program managers and systems engineers are sometimes intimidated by cybersecurity issues and thus seek to pass them onto specialists later in the acquisition process.
- Model-based engineering toolset developers have not sufficiently considered programmatic cybersecurity vulnerabilities, though the tools are thought to be quite effective at designing for cybersecurity in regard to end-systems.
- Traditional programmatic cybersecurity defensive practices tends to be quite effective in traditional engineering programs, but the increased use of digital engineering, particularly for multi-site collaboration, could change this. (Reid & Rhodes, 2018b).
- Model-centric approaches relay on much more infrastructure (computing, data storage, software packages, etc.) that could be compromised in myriad ways.
- Model sharing is becoming increasingly important, increasing potential for human actor induced vulnerabilities.
- With increased use of software toolsets, there may be secondary vulnerabilities that are more difficult to detect.
- Cybersecurity vulnerabilities can result in non-technical impacts such as reduced trust of models.

Non-technical Influences and impacts

One set of vulnerabilities that came up repeatedly in both the interviews and experiment sessions in our research (Reid and Rhodes, 2018b) were those that passed through *the reputation harm intermediary event*, as shown in Figure 7. Despite the frequency that the potential for this vulnerability was raised by experts, few interventions were proposed for post-breach. According to Reid (2018), this suggests that leaders of digital engineering enterprises may need better understanding of potential vulnerabilities leading to breaches in contact of digital engineering, as well as more knowledge on how to respond to breaches, particularly prominent ones, instead of solely how to prevent them. While in the private sector there is evidence suggesting that the reputation harm incurred by a prominent breach does not significantly impact the firm (Lange & Burger, 2017), contractors to the government are known to suffer significant financial penalties due to breaches, even when such a breach is unrelated to their government duties (Braun, 2014; Overly, 2017). In a defense acquisition environment, there is thus significant incentive to having program leadership (and the enterprise as a whole) well-prepared to respond to major breaches.

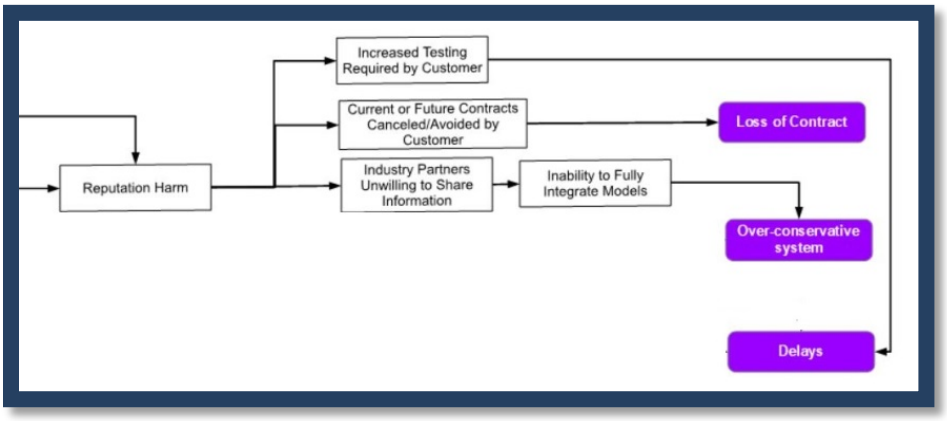


Figure 7. Reputation Harm Vulnerabilities (Reid, 2018)

Relevant Research from Other Fields

Huff et al. (2018) present a methodology for performing vulnerability assessment and decision analysis of critical infrastructure using the approach of

model-based systems engineering. The work focuses on physical security of critical infrastructure. Some of their findings may provide useful insights for vulnerability assessment of infrastructure within model-centric enterprises.

The literature from the manufacturing sector offers interesting observations and new research of relevance to vulnerability assessment of model-centric enterprise environments. Burnson (2017), discussing a recent Deloitte study on cyber vulnerabilities in manufacturing supply chains, states “one-third of all manufacturers sampled admitted to not having performed any cyber risk assessments of the industrial connected devices operating on factory floors”. While data is not available, from discussions with experts in the engineering domain it seems likely that there would be a similar situation in regard to whether cyber risk assessments have been performed for model-centric engineering environments with connected hardware and software.

DeSmit et al. (2016) discuss research on cyber-physical vulnerability assessment in manufacturing systems that uses an approach that employs *intersection mapping*. According to these authors, “no literature is aimed at assessing cyber-physical vulnerabilities for manufacturing systems”. With similarities of manufacturing facilities with facilities used in model-centric enterprises, their research may offer useful insights to our research. DeSmit et al. describe their approach as “…based on the principle that vulnerabilities in manufacturing systems occur at intersections (and intra-sections, referred to collectively as intersections) of cyber, physical, cyber-physical and human entities that embody a manufacturing system”. Similar to the CEM approach, their method maps intersections and assesses the impact at intersection notes. They evaluate five characteristics: loss of information, inconsistency, relative frequency, lack of maturity and time until detection. In their method, vulnerability impact assessment (Low, Medium High) is assessed for the characteristics at each of the nodes. This offers an interesting approach to qualitative assessment measures for vulnerability. Another noteworthy facet of their work that resonates with our research is that human entities are included in defining intersections.

Preliminary Investigation

Given limitations of the research, preliminary investigation of dynamic methods and an interactive prototype were conducted. The investigations suggest further research in the future would be very valuable.

Dynamic Methods

The use of dynamic methods has been explored in the research as adjunct analytic approaches for cause-effect mapping (Reid, 2018). System Dynamics is a method particularly useful for this due to the preexisting models of many organizational phenomena (Rouwette & Ghaffarzadegan, 2013). For instance, the *Attrition*, *Reduced Model Training,* and *Less Model Expertise* can be modeled by adapting the rookie fraction model shown in Figure 8 into the more model-centric relevant model shown in Figure 9. In this model, it is apparent that a hiring freeze (which would set the "Growth Rate" variable to zero) has no immediate impact, as rookies will continue to develop into experienced employees and model expertise will continue to accumulate. Over time, however, the dearth of new rookies will result in fewer experienced employees, increasing the error rate. These kinds of long-term, indirect impacts are likely to become more common with increased use of MCE.

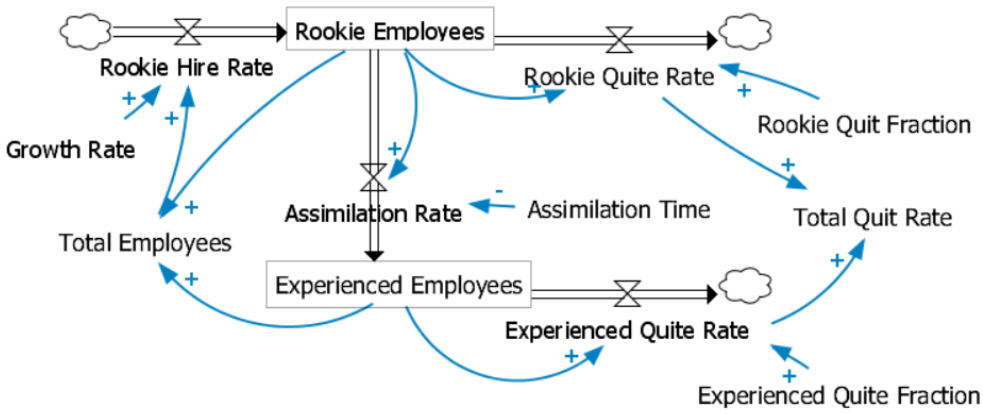


Figure 8. System Dynamics Model of Employee Training Rate. Adapted from (Sterman, 2000)

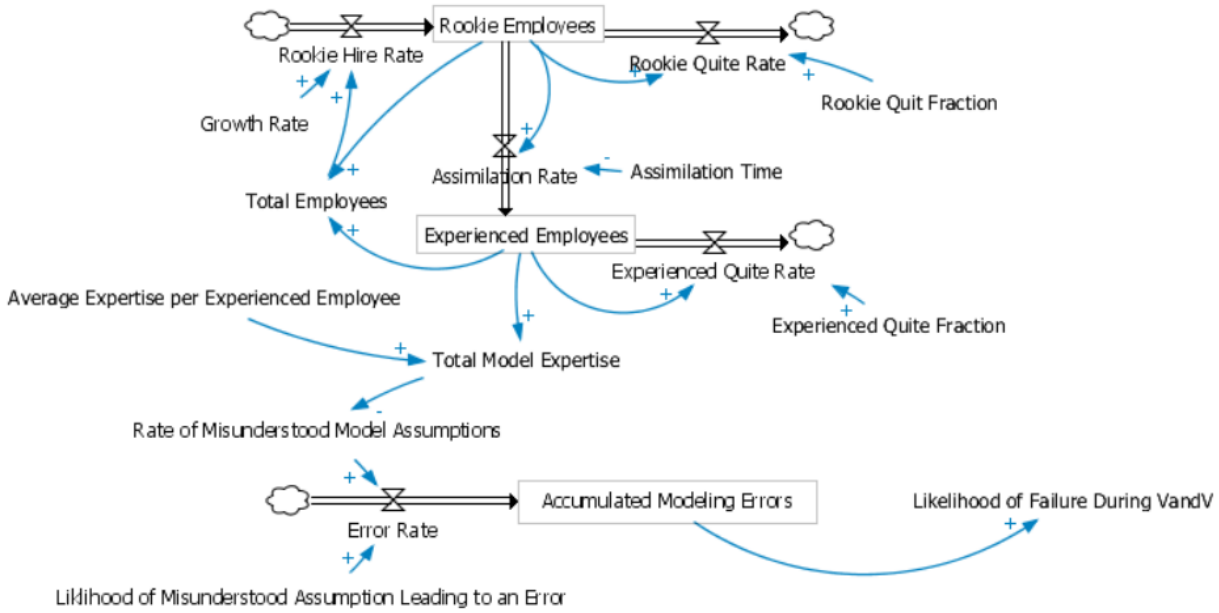


Figure 9. System Dynamics Model of Accumulated Modeling Errors

Various analytic approaches considered in the research include network analysis, graph theory techniques, and use of system dynamics. This investigation reinforced the potential benefit of augmenting the use of cause-effect mapping with other analytic techniques to enrich the analysis. As discussed in Reid (2018), there is a need to better determine where to place intervention points in intertwined vulnerability chains. Simple in-degree/out-degree analysis is helpful, but more sophisticated applications of graph theory and probabilistic modeling offer greater potential benefit. As more information on vulnerabilities in model-centric enterprises becomes available, the Reference CEM can be enhanced. For instance, if probabilities, likelihoods, or time scales of each event transition are known, techniques such as Markov Chain Modeling, Monte Carlo Analysis, and Bayesian Networks can be brought to bear, weighting each graph instead of treating them equally.

Interactive Reference CEM

The potential for an interactive Reference CEM was explored in the research, given the potential benefits. One of these would be that of increased accessibility. As

an example, Ross et al. (2016) state the interactive NIST Cybersecurity Assessment Tool makes the risk management framework more approachable to small manufacturers. Availability of a prototype for an Interactive Reference CEM could provide a platform for future usability testing of cause-effect mapping in model-centric enterprises. An interactive and executable Reference CEM would enable more effective use of dynamic methods. A web-based implementation could be a good approach to enable more effective collaboration in generating a standard reference map for vulnerability assessment of a program or enterprise. As discussed in Grogan et al. (2015), new methods for collaborative modeling are enabled with web and browser-based technologies.

An *Interactive Reference CEM* could provide options for a program leader to more effectively and completely explore vulnerabilities. An initial concept was explored in the research. It is envisioned that the user would click on one of the external triggers, that could then display the related vulnerability chains. Clicking on an intermediary event or on a link between events could be set up to display lessons learned and strategies for addressing the vulnerability chain through interventions. Clicking on a terminal event could display the vulnerability chains as working backward from that event.

Given the complexity of vulnerabilities in a model-centric program or enterprise, an Interactive Reference CEM could reduce the complexity of displayed information and/or underlying analytics. The ability to analyze the value and impact of interventions would be enabled through interactivity. A program leader could perform "what if" analysis, including varying weighting and probability of intermediary events. Impacts of interventions could be explored to better enable the decisions of where to invest in vulnerability interventions and where to assume the risk. This is very important given the realities that many more interventions could be made than would be possible given practical budget and time constraints. Interactivity could also facilitate the development of a standardized Reference CEM for a given enterprise, if designated individuals were able to easily update the Reference CEM using knowledge gained through experience with digital engineering practice and model-centric environments.

Future Research Directions

This section discusses recommendations for future research and limitations of the research.

Limitations

While a fully-developed generalized CEM Reference Map could provide overall benefit to digital engineering programs, the fact that programs and enterprises are unique makes it difficult to accomplish this without much more extensive application and study. Secondly, digital engineering practice and supporting environments are still evolving, so limited knowledge exists at present. Nonetheless, programs and enterprises may derive significant benefit by the activity of constructing a Reference CEM for their unique situation. The Reference CEM generated in this research can serve as a baseline for constructing a unique map. The process of generating the map invokes thoughtful discussion and anticipating potential hazards that may have been introduced as a result of the digital transformation. The approach of considering vulnerabilities as casual chains yields rich discussion, regardless of whether an overall map is developed.

Recommendations for Future Research

Knowledge gathered in this research indicates that program leaders do not formally grapple with vulnerabilities within the program and overall enterprise to the extent they do with vulnerabilities related to the end-system. Cause-Effect Mapping with re-conceptualizing vulnerabilities as causal chains enables program and enterprise leaders to identify connections, categories, and potential interventions in the vulnerability chains. The research indicates identifying external triggers and representing vulnerabilities as chains enables a more detailed assessment of how interim cascading events can result in significant terminal outcomes. Use of the CEM approach assists in understanding these causal chains, and decomposes a vulnerability in a manner that encourages finding multiple options for mitigation. Particular choices for disrupting a harmful causal chain are useful for considering

where and when to place interventions based on the specific nature of the situation. Several suggested future research directions follow.

Continued Knowledge Gathering

Additional study is needed on leading indicators of vulnerability in digital engineering enterprises, along with potential mitigation strategies. Specific approaches to quantification of interventions in breaking vulnerability causal chains is desired as related to cost, benefit, importance, frequency, etc. Additional investigation is needed to continue to explore the latest research related to digital engineering, for example, Wach & Salado (2018) describe a plan to discover patterns of unknown vulnerabilities associated with SysML.

Additional research on dynamic simulation with CEM is a promising area to explore given the complexities that will be inherent in a fully populated reference CEM (further discussion is found in Reid, 2018). Model-based implementation of the interaction method to perform vulnerability assessment using a Reference CEM is a future area of inquiry. And, further collaborative research with government and industry is desired to gain knowledge on scaling of the approach, and transitioning it to practice.

Measures of Enterprise Capabilities for Cybersecurity

A study report by the NAS (2017) discusses the need for research on cybersecurity metrics. It states, “There are properties of systems and of organizations that can be measured and can serve as a useful proxy against known classes of attacks”. As a result a future area of research would be to investigate vulnerability metrics that could measure the capabilities that the enterprise possesses for vulnerability assessment and mitigation. The NAS study cites examples of indicators that could be measureable:

> Does the organization use publicly available indicators of compromise? For what fraction of its systems and network? Does the organization collect data flows on its internal network and external connections? Another aspect that might be subject to measurement is commitment to continuous improvement. Does the

organization learn from its own (and others') mistakes and adapt? Over what kind of time frame?

Vulnerabilities under Varying Contexts

Given the limitation of resources within enterprises, National Academies of Sciences (2017) suggests that researchers have an opportunity to investigate the degree to which certain practices (or interventions) are successful in a given context. A Reference CEM could provide a common basis for capture of practices and interventions that could then be studied under differing program and enterprise contexts. As a result, program leaders would be able to understand the vulnerabilities respective to their specific program and context situation, which would inform the most effective use of resources for mitigation and intervention. This would also provide an opportunity to do some sensitivity analysis around intervention approaches.

THIS PAGE LEFT INTENTIONALLY BLANK

Conclusion

Digital engineering transformation introduces new vulnerabilities within model-centric acquisition programs and enterprises. Causal chains provide a useful way to understand how external triggers lead to cascading intermediary events that result in specific outcomes. Understanding a vulnerability chain provides program leaders with increased knowledge and options for inserting interventions to avoid undesired vulnerability outcomes. With more experience and knowledge of vulnerabilities inherent in digital engineering practice and infrastructure, the systems community may find it valuable to establish a standardized Reference CEM that can guide future programs and enterprises to assess and manage vulnerabilities, leading to more successful program outcomes. Related research on model curation views a Reference CEM as an enabling tool (Rhodes, 2019) for enterprise leaders.

Outcomes achieved in the two phases of research include empirically-grounded vulnerabilities of model-centric programs and enterprises, and a cause-effect mapping reference model (Reference CEM) for identifying vulnerabilities and interventions. Investigation of cybersecurity vulnerabilities within model-centric enterprises uncovered vulnerability chains, and revealed this as an important area for further study. The research confirms the need to perform formal vulnerability assessment of model-centric acquisition enterprise practices and environments. Failing to uncover such vulnerabilities could ultimately jeopardize program success and lead to end-system failures. Additionally, the research confirms the need for further investigation, including dynamic approaches and interactive reference model, context-specific vulnerabilities, and measures of enterprise vulnerability assessment capability.

THIS PAGE LEFT INTENTIONALLY BLANK

References

Braun, S. (2014, September 10). OPM plans to terminate contracts with USIS. Federal News Radio. Retrieved from https://federalnewsradio.com/management/2014/09/opm-plans-to-terminate-contracts-with-usis/

Burnson, P. (2017). New Deloitte study identifies cyber vulnerabilities in manufacturing supply chains. Supply Chain Management Review, https://www.scmr.com/article/new_deloitte_study_identifies_cyber_vulnerabilities_in_manufacturing_supply. Accessed 26 Mar 2019.

Conigliaro, R. A., Kerzhner, A. A., & Paredis, C. J. J. (2009). Model-Based Optimization of a Hydraulic Backhoe using Multi-Attribute Utility Theory. *SAE International Journal O Materials and Manufacturing*, *2*(1), 298–309. Retrieved from http://www.sae.org

Department of Defense. (2018, June). Department of Defense Systems Engineering Strategy. https://www.acq.osd.mil/se/docs/2018-DES.pdf

DeSmit, Z., Elhabashy, A., Wells, L. & Camelio, J. (2016). 44th Proceedings of the North American Manufacturing Research Institution of SME. Procedia Manufacturing. 5: 1060-1074.

German, E.S. & Rhodes, D.H. (2017). Model-Centric Decision-Making: Exploring Decision-Maker Trust and Perception of Models. 15th Conf. on Systems Engineering Research. Los Angeles, CA.

Gressin, S. (2017). The Equifax Data Breach: What to Do. Retrieved March 27, 2018, from https://www.consumer.ftc.gov/blog/2017/09/equifax-data-breach-what-do

Grogan, P., de Weck, O., Ross, A., & Rhodes, D.H. (2015). Interactive Models as a System Design Tool: Applications to System Project Management, Procedia Computer Science, Vol 44, p 285-294

Hanna, J., Smythe, C., & Martin, C. (2018, January 24). China's Sinovel Convicted in U.S. of Stealing Trade Secrets. Bloomberg. Retrieved from https://www.bloomberg.com/news/articles/2018-01-24/chinese-firm-sinovel-convicted-in-u-s-of-trade-secret-theft

Hanselman, S. (2012). Everything's broken and nobody's upset. Retrieved March 27, 2018, from https://www.hanselman.com/blog/EverythingsBrokenAndNobodysUpset.aspx

Huff, J., Medal, H. & Griendling, K. (2019). A model-based systems engineering approach to critical infrstructure vulnerability assessment and decision analysis. *Systems Engineering*. 22: 114– 133

Kamali, M., Dennis, L. A., McAree, O., Fisher, M., & Veres, S. M. (2016). Formal verification of autonomous vehicle platooning. Science of Computer Programming, 1, 1–19. https://doi.org/10.1016/j.scico.2017.05.006

Kern, C. & Greenstreet, M. R. (1999). Greenstreet, Formal Verification in Hardware Design: a survey. ACM Transactions on Design Automation of Electronic Systems, vol. 4, no. 2, pp. 123–193.

LeSaint, J., Reed, M., & Popick, P. (2015). System Security Engineering Vulnerability Assessments for Mission-critical Systems and Functions. In 2015 Annual IEEE Systems Conference (SysCon) Proceedings. 608-613.

Maley, J., & Long, J. (2005). *A Natural Approach to DoDAF*. Blacksburg, VA.

Martz, M., & Neu, W. L. (2008). Multi-Objective Optimization of an Autonomous Underwater Vehicle. *Oceans 2008, Vols 1-4*, 1042–1050\r2248.

Meadows, C. A. (1994). Formal Verification of Cryptographic Protocols: A survey. In International Conference on the Theory and Application of Cryptology (pp. 133–150). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0000430

Mekdeci, B., Ross, A. M., Rhodes, D. H., & Hastings, D. E. (2012). A taxonomy of perturbations: Determining the ways that systems lose value. In *2012 IEEE International Systems Conference, Proceedings* (pp. 507–512). Vancouver: IEEE. https://doi.org/10.1109/SysCon.2012.6189487

Morimoto, S. (2008). A Survey of Formal Verification for Business Process Modeling (pp. 514–522). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69387-1_58

NASA Office of the Chief Engineer. (1994). NASA Public Lessons Learned System. Retrieved July 13, 2017, from https://llis.nasa.gov/

NAS, National Academy of Sciences. Foundational Cybersecurity Research: Improving Science, Engineering, and Institutions.National Academies of Sciences, Engineering, and Medicine. 2017. Washington, DC: The National Academies Press. https://doi.org/10.17226/24676.

Overly, S. (2017, October). IRS temporarily suspends contract with Equifax. Politico. Retrieved from https://www.politico.com/story/2017/10/12/irs-equifax-contract-suspended-243732

Pope, G. (2017). A Hazard Analysis Technique for the Internet of Things (IoT) and Mobile. In STAMP Workshop. Cambridge, MA.

Pope, G. (2018). Combining STPA with Compiler Technology to Identify Vulnerabilities and Hazards in Software-Controlled Systems. In STAMP Workshop. Cambridge, MA.

Pope, G., & Yampolskiy, M. (2016). A Hazard Analysis Technique for Additive Manufacturing. In Better Software East Conference. Orlando, FL. Retrieved from https://arxiv.org/ftp/arxiv/papers/1706/1706.00497.pdf

Raymond, N. (2017, August 31). U.S. charges Chinese-Canadian citizen with trade secret theft. Reuters2. Retrieved from https://ca.reuters.com/article/topNews/idCAKCN1BB2K8-OCATP

Reid, J. B., & Rhodes, D. H. (2016). Digital System Models : An investigation of the non-technical challenges and research needs. In *Conference on Systems Engineering Research*. Huntsville, AL.

Reid, J. B., & Rhodes, D. H. (2018a). Assessing Vulnerabilities in Model-Centric Acquisition Programs Using Cause-Effect Mapping. In 15th Annual Acquisition Research Symposium. Monterey, CA: Naval Postgraduate School.

Reid, J. B., & Rhodes, D. H. (2018b). Applying Cause-Effect Mapping to Assess Cybersecurity Vulnerabilities in Model-Centric Acquisition Program Environments In 15th Annual Acquisition Research Symposium. Monterey, CA: Naval Postgraduate School.

Reid, J.B. (2018). Assessing and Mitigating Vulnerability Chains In Model-Centric Acquisition Programs, MIT Master's Thesis.

Rhodes, D.H. & Reid, J.B. (2018a). Assessing Vulnerabilities in Model-Centric Acquisition Programs Using Cause-Effect Mapping. NPS Acquisition Research Program Sponsored Report Series, 3 August 2018.

Rhodes, D.H. & Reid, J.B. (2018b). Uncovering Cascading Vulnerabilities in Model-Centric Programs and Enterprises. NDIA Systems Engineering Conference, Springfield, VA.

Rhodes, D..H. & Reid, J.B. (2019). Uncovering Cascading Vulnerabilities in Model-Centric Acquisition Programs and Enterprises, In 15th Annual Acquisition Research Symposium. Monterey, CA: Naval Postgraduate School.

Rhodes, D.H. (2019). Model Curation: Requisite Leadership and Practice in Digital Engineering Enterprises. 17th Conference on Systems Engineering Research, Washington, DC.

Ross, R., Dempsey, K., Pillitteri, V. Y., Jacobs, J., & Goren, N. (2016). Risk Management. Retrieved March 29, 2018, from https://csrc.nist.gov/projects/risk-management/risk-management-framework-(RMF)-Overview

Rouwette, E., & Ghaffarzadegan, N. (2013). The system dynamics case repository project. *System Dynamics Review*, *29*(1). https://doi.org/10.1002/sdr.1491

Rovito, S. M., & Rhodes, D. H. (2016). Enabling Better Supply Chain Decisions Through a Generic Model Utilizing Cause-Effect Mapping. In *2016 Annual IEEE Sytems Conference, Proceedings*. Orlando: IEEE.

Schwarz, E. (2018). Automating Vulnerability Discovery in Critical Applications. Retrieved March 27, 2018, from https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6487

Software Engineering Institute. (2007). Acquisition Archetypes: Firefighting. Pittsburgh, PA.

Statt, N. (2018, March). Boeing production plant hit with WannaCry ransomware attack. The Verge. Retrieved from https://www.theverge.com/2018/3/28/17174540/boeing-wannacry-ransomware-attack-production-plant-charleston-south-carolina

Sterman, J. D. (2000). Coflows and Aging Chains. In *Business Dynamics: Systems Thinking and Modeling for a Complex World* (pp. 469–512). Boston, MA: Irwin McGraw-Hill.

The MITRE Corporation. (2015). Terminology. Retrieved February 20, 2018, from https://cve.mitre.org/about/terminology.html

The MITRE Corporation. (2017). CVE-2017-5753. Retrieved February 20, 2018, from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5753

Wach, P. & Salado, A. (2018). A Research Plan to Discover Patterns of Unknown Vulnerabilities Associated with Adopting SysML. *16th Conf. on Systems Engineering Research.*

Young, W. E. (2013). A System Safety Approach to Assuring Air Operations Against Cyber Disruptions. In *STAMP Workshop*. Cambridge, MA.

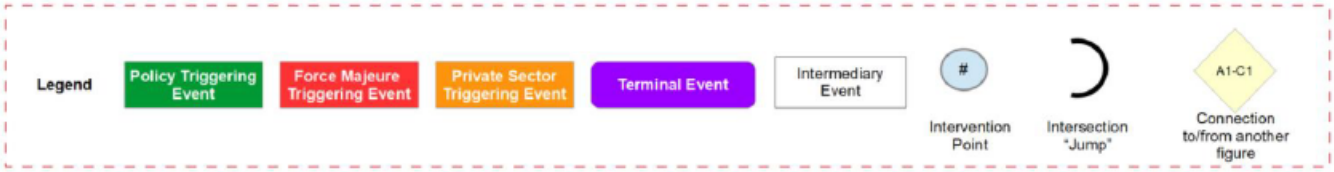
Young, W. E., & Porada, R. (2017). System-Theoretic Process Analysis for Security (STPA-SEC): Cyber Security and STPA. In *STAMP Workshop*. Cambridge, MA.

Appendix A (Reference CEM)

This appendix contains detailed information about the Reference CEM.

Four figures are provided for the Reference CEM. This includes one figure showing the legend and intervention point description. Three figures are the portions of Reference CEM that are provided to enable better readability of the results.

Five tables are provided to describe the external triggers, intermediary events, and terminal events, as correspond to the Reference CEM.



Intervention Points

1) Initiate Internal Assessment and a PR Strategy
2) Initiate various non-monetary benefits (such as allowing pets at work or a 9/80 schedule) to encourage employees to stay
3) Seek to share resources and employees with other programs
4) Hire employees with prior experience with the new software
5) Compartmentalize sensitive information
6) Obfuscate sensitive data with false or misleading information
7) Create documentation and curation processes within the program
8) Institute handover periods to benefit from contractor expertise
9) Reevaluate training regime and needed fields of expertise
10) Increase the amount of testing conducted
11) Increase use of contractors/consultants (including previous employees) to maintain expertise level
12) Reevaluate requirements with the client and other stakeholder
13) Design for modularity to minimize impact on system
14) Negotiate with client to see if they are able to pay for the software
15) Isolated but readily accessible back-ups of data
16) Reviews/Comparisons of models between lifecycle stages
17) Multiple, independent simulations or component checkers
18) Isolated, independent backup equipment that can be switched to while primary equipment is being evaluated
19) Conduct regular "red-team" / penetration test exercises

Figure A. 1 Legend and Intervention Points for Reference CEM

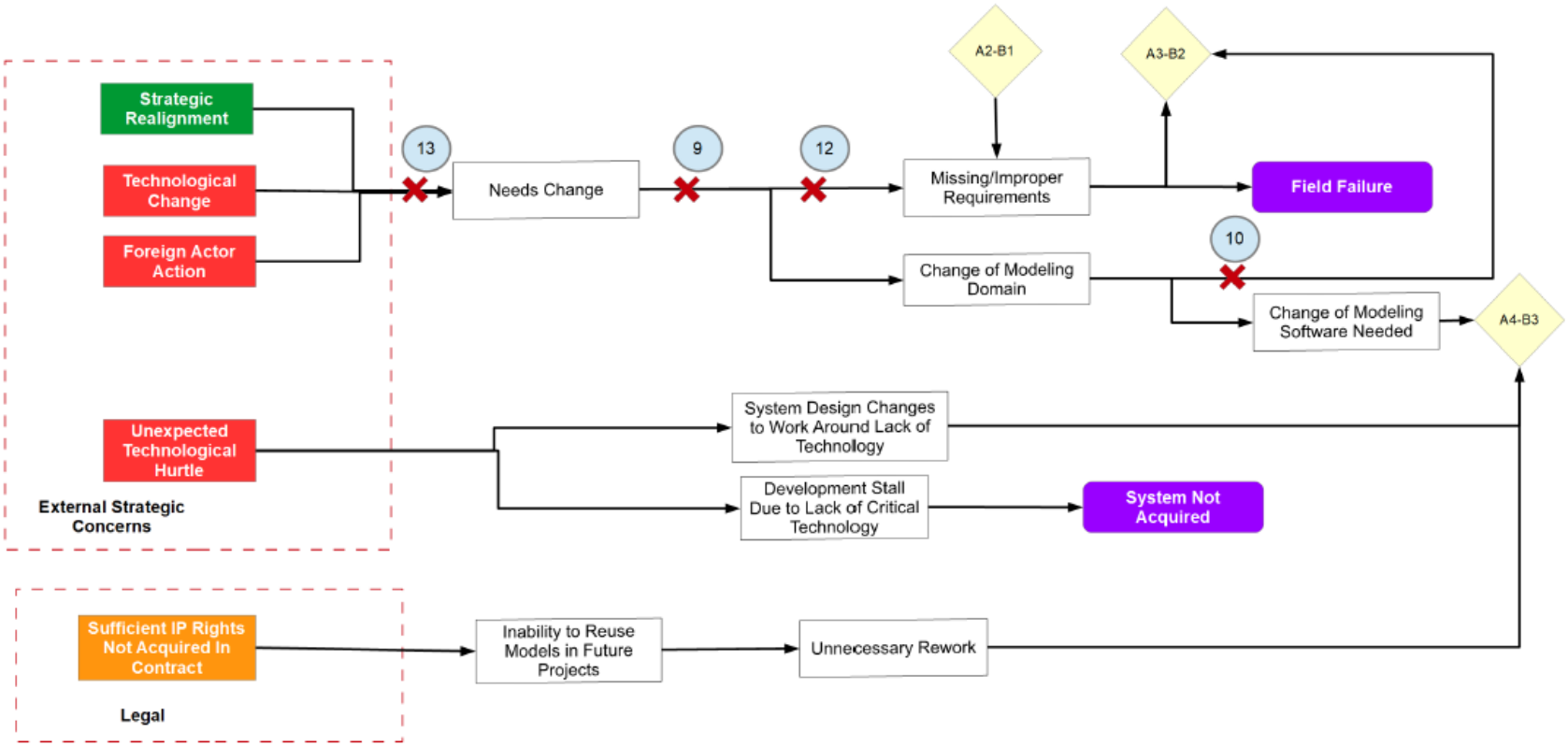


Figure A. 2 Part A of the Reference CEM

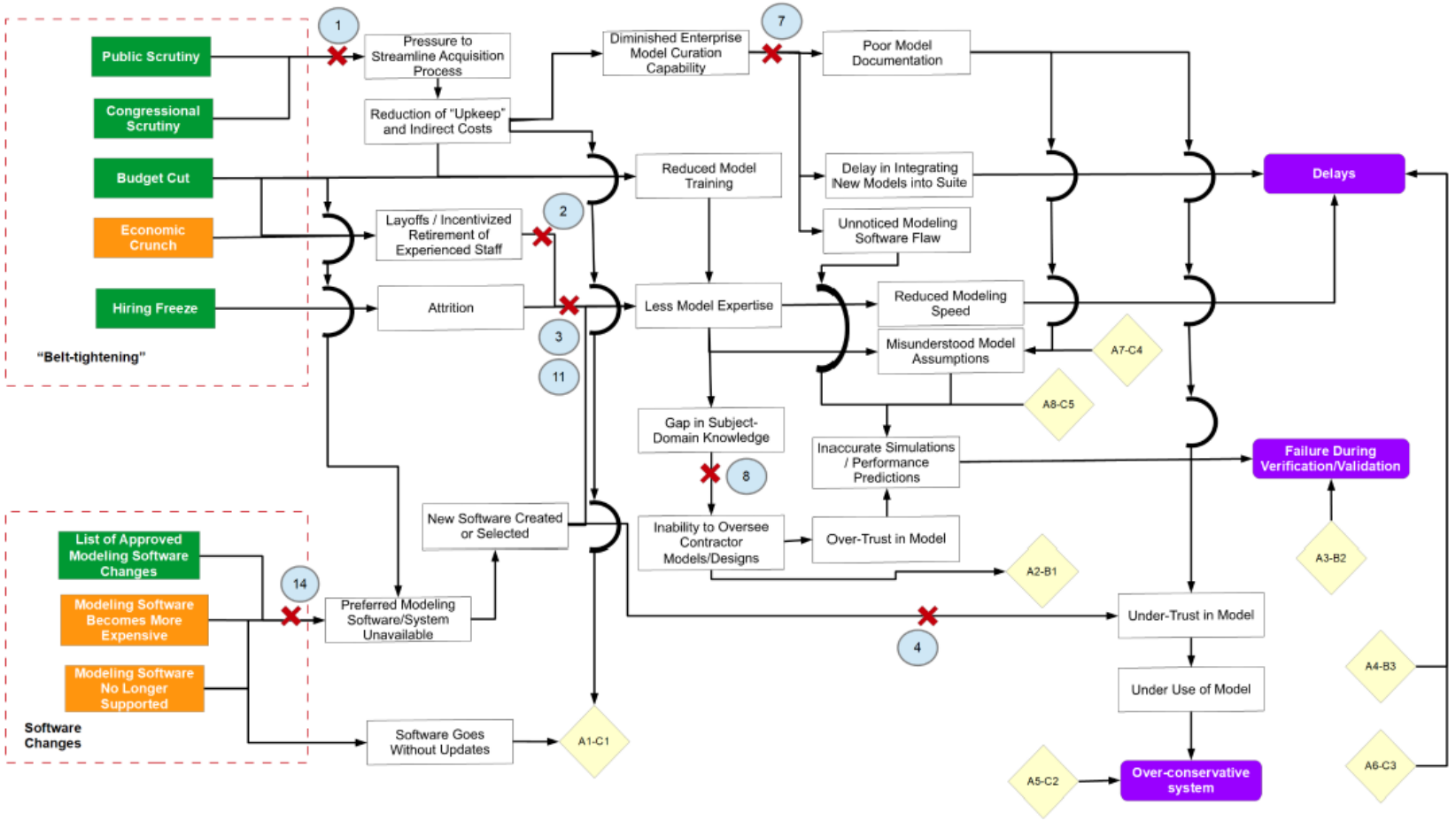


Figure A. 3 Part B of the Reference CEM

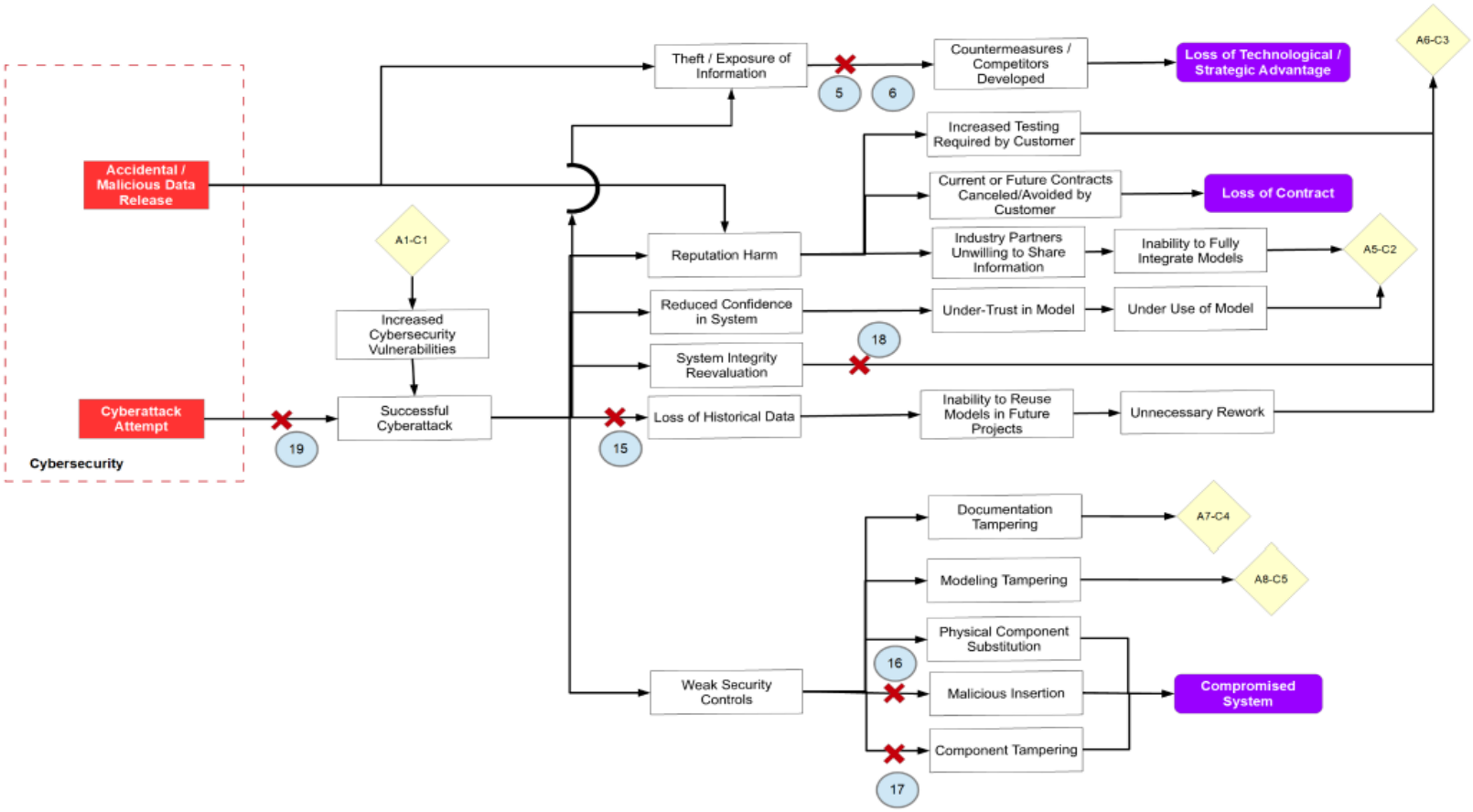


Figure A. 4 Part C of the Reference CEM

Table A-1. Reference CEM Events and Descriptions - External Triggers (Part 1 of 5)

Event Name	Event Type	Event Description
Accidental/Malicious Data Release	External Trigger	Either accidentally or intentionally, some amount of sensitive data involving the program has been released to individuals or groups not cleared to access such information
Budget Cut	External Trigger	Current or projected funding for this program or for the enterprise as a whole is being reduced
Congressional Scrutiny	External Trigger	Congress has become increasingly concerned with the management or status of this program or of defense programs in general
Cyberattack Attempt	External Trigger	Some individual or group attempts to disrupt or surveil the program using a cyberattack
Economic Crunch	External Trigger	Reduced consumption, reduced willingness to lend, raising unemployment or other forms of economic recession or depression are occurring
Foreign Actor Action	External Trigger	A foreign actor has taken some significant action that impacts the intended use of system
Hiring Freeze	External Trigger	The program or enterprise has ceased all new hiring for some period of time
List of Approved Modeling Software Changes	External Trigger	The organization maintains a list of software approved for use in programs and has changed (added/removed) certain software from this list
Modeling Software Becomes More Expensive	External Trigger	Some modeling software used by the program which is purchased or licensed from an external provider has become more expensive in the upcoming version/renewal
Modeling Software No Longer Supported	External Trigger	Maintainer/developer of in-use modeling software has ceased issuing updates and/or new versions
Public Scrutiny	External Trigger	The public and/or news media has become increasingly concerned with management of program or of defense programs in general
Strategic Realignment	External Trigger	The strategic interests of the client or other stakeholders in the program have changed
Sufficient IP Rights Not Acquired	External Trigger	Sufficient intellectual property rights not acquired for system components, product, data.
Technological Change	External Trigger	A significant new technology has been developed or put into use that impacts program
Unexpected Technological Hurtle	External Trigger	During the program, some desired technology either is unexpectedly unavailable or is taking an unexpectedly long time to develop

Table A-2. Reference CEM Events and Descriptions – Intermediary Events (Part 2 of 5)

Event Name	Event Type	Event Description
Attrition	Intermediary Event	Reduction in experienced staff and/or total staff available to the program
Change of Modeling Domain	Intermediary Event	The program must now model the system, subsystem, or component in a different environment or in a different manner than was previously done
Change of Modeling Software Needed	Intermediary Event	The program must now change the modeling software used
Component Tampering	Intermediary Event	A digital or physical component of the system is maliciously altered to harm the program
Countermeasures/Competitors Developed	Intermediary Event	Other individuals/orgs able to either develop countermeasures to system being acquired or developing competing systems
Current or Future Contracts Canceled/Avoided by Customer	Intermediary Event	The program is either cancelled by the client or future programs are never initiated
Delay in Integrating New Models into Suite	Intermediary Event	Increased time is required to integrate a new modeling software or model into the MCE environment
Development Stall Due to Lack of Critical Technology	Intermediary Event	A critical technology for the program is not available and there is not means of altering the design to circumvent the need for it
Diminished Enterprise Model Curation Capability	Intermediary Event	Reduced ability of the enterprise or program to track, integrate, document, and improve the modeling environment
Documentation Tampering	Intermediary Event	Documentation of assumptions or other important aspects of models is maliciously altered to harm the program
Gap in Subject-Domain Knowledge	Intermediary Event	Among program staff there is a lack of knowledge on a particular subject area of relevance to the program
Inability to Fully Integrate Models	Intermediary Event	One or more models in use by the program cannot be integrated into the MCE environment and thus remain "stove-piped"
Inability to Oversee Contractor Models/Designs	Intermediary Event	Program staff no longer have the ability to effectively understand and vet the models and/or designs provided by the contractor
Inability to Reuse Models in Future Projects	Intermediary Event	Models from previous or current programs are not available to be used in future programs
Inaccurate Simulations / Performance Predictions	Intermediary Event	Models and simulations do not accurately reflect real-world behavior or performance
Increased Cybersecurity Vulnerabilities	Intermediary Event	The program and/or MCE environment has become more susceptible to cyberattacks
Increased Testing Required by Customer	Intermediary Event	The client increases the amount of testing required during verification to demonstrate system readiness
Industry Partners Unwilling to Share Information	Intermediary Event	Subcontractors, suppliers, or other members of industry are unwilling to share information with the program due to concerns that it will be misused or inappropriately shared
Layoffs/Incentivized Retirement of Experienced Staff	Intermediary Event	Staff experienced with the MCE environment or having general relevant experience are laid off or incentivized to retire
Less Model Expertise	Intermediary Event	Among program staff there is either a reduced total amount of experience or reduced average amount of experience with the MCE environment

Table A-3. Reference CEM Events and Descriptions - Intermediary Events (Part 3 of 5)

Event Name	Event Type	Event Description
Loss of Historical Data	Intermediary Event	Data from current or previous programs has been lost or corrupted such that it is unavailable for future reference or reuse
Malicious Insertion	Intermediary Event	A new digital or physical object is inserted into the model or system with intent to harm system integrity
Missing/Improper Requirements	Intermediary Event	The current set of requirements do not sufficiently match stakeholder needs, either due to a lack of relevant requirements or due to a conflict between the requirements and needs
Misunderstood Model Assumptions	Intermediary Event	Some member of the program staff misunderstands or fails to consider one or more of the assumptions underlying the model in such a way as to potentially impact the program
Modeling Tampering	Intermediary Event	The models themselves are maliciously altered to harm the program and/or enterprise
Needs Change	Intermediary Event	The needs of the client stakeholder(s) are changing in a way that impacts the program
New Software Created or Selected	Intermediary Event	A new modeling or simulation software must be created or selected (if COTS) and integrated into the MCE environment
Over-Trust in Model	Intermediary Event	One or more members of the program staff trust the current state of the model or results of simulation when these do not represent reality in some significant manner
Physical Component Substitution	Intermediary Event	A physical component in the system being acquired is substituted, threatening system integrity
Poor Model Documentation	Intermediary Event	The model documentation is missing relevant information or presented in an inaccessible manner
Preferred Modelling Software/System Unavailable	Intermediary Event	The preferred modeling software or system, either due to prior experience or particular relevance to the program, is unavailable for use in this program
Pressure to Streamline Acquisition Process	Intermediary Event	External pressure, either from the organization, client, or other stakeholder, is exerted on the program manager to minimize costs, timing, and "bloat."
Reduced Confidence in System	Intermediary Event	Members of the program, organization, or clients have reduced confidence in the ability of the MCE environment to be able to operate effectively and securely
Reduced Model Training	Intermediary Event	Reduction in training in the use of the MCE environment is available to program personnel
Reduced Modeling Speed	Intermediary Event	Time required to effectively use the modeling software is increased
Reduction of "Upkeep" and Indirect Costs	Intermediary Event	Reduction or elimination of "unessential" procedures, tests, and personnel. Particularly likely targets include any procedures aimed at improving reuse or other programs
Reputation Harm	Intermediary Event	The professional, political, or public reputation of the program, organization, or client has become significantly harmed
Software Goes Without Updates	Intermediary Event	Software important to the MCE environment goes without updates or patches for longer than ideal, resulting in lack of capability or security concerns
Successful Cyberattack	Intermediary Event	A cyberattack was not repelled or stopped until after it significantly impacted the program

Table A-4 Reference CEM Events and Descriptions – Intermediary Events (Part 4 of 5)

Event Name	Event Type	Event Description
System Design Changes to Work Around Lack of Technology	Intermediary Event	A critical technology for the program is not available and thus the design of the system must be altered in order to circumvent the need for the technology
System Integrity Reevaluation	Intermediary Event	The program or organization must reevaluate the integrity of the MCE environment and/or the system being designed to ensure that it has not been compromised
Theft/Exposure of Information	Intermediary Event	Sensitive or otherwise non-public information is either intentionally stolen or otherwise made available to those uncleared to possess it
Under-Trust in Model	Intermediary Event	One or more members of the program staff do not place do not believe the current state of the model or results of simulation when these do represent reality to a sufficient extent
Under-Use of Model	Intermediary Event	The program does not make full effective use of the MCE environment available
Unnecessary Rework	Intermediary Event	Work previously accomplished in the current program or a previous one must be done again when it should not have to be
Unnoticed Modeling Software Flaw	Intermediary Event	Some modeling bug, error, or other form of inaccuracy develops and goes undetected by the model curation staff and/or the program
Weak Security Controls	Intermediary Event	No or few security controls are in place to prevent physical or virtual security compromises

Table A-5 Reference CEM Events and Descriptions – Terminal Events (Part 5 of 5)

Event Name	Event Type	Event Description
Compromised System	Terminal Event	The system is put into operation, but suffers from lack of integrity
Delays	Terminal Event	Acquisition of the program is delayed and/or increased costs incurred
Failure During Verification/Validation	Terminal Event	The system fails during verification and/or validation, prompting redesign to occur or cancellation of the program
Field Failure	Terminal Event	The system passes verification or validation but fails during operation
Loss of Contract	Terminal Event	The program is cancelled
Loss of Technological/Strategic Advantage	Terminal Event	The system does not provide the advantage or superiority that it was intended to
Over-conservative System	Terminal Event	The system is "over-engineered" and thus more costly and/or has reduced performance that should be possible
System Not Acquired	Terminal Event	The program is cancelled and the system is not acquired

Acquisition Research Program
Graduate School of Business & Public Policy
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net