

# A Model Centric Framework and Approach for Complex Systems Policy

Shamsnaz Virani Bhada  and Rahul Krishnan 

**Abstract**—Twenty-first century systems engineering is no longer document-centric; instead, it is model-centric. Model-centric systems engineering helps reduce ambiguity, increase clarity, and increase the analytics of the resulting complex systems. However, complex systems are governed by organizational policies that are still document-centric. Such policies are difficult to analyze, and gaps in policy can lead to major deficiencies in the resulting complex systems. This article introduces a framework for policy content modeling (PCM) and analysis. The framework represents the conceptual view and is supported by a step-by-step approach to achieve complete policy modeling and analysis. This approach was used with the intention of identifying and analyzing gaps in policy content and calculating policy toxicity, which negatively affects the resulting system. This framework and approach was also applied to Veterans Affairs (VA) and university policies. The VA PCM is conducted to discover toxicity in the policies and University policies modeling is done to graphically represent an undocumented policy and toxicity in the policy implementation.

**Index Terms**—Complex systems policy, modeling, model-based systems engineering (MBSE), model centric systems engineering, policy, policy analysis.

## I. INTRODUCTION

COMPLEX engineered systems are manifestations of requirements derived from customer wants and needs. In addition to satisfying customer requirements, engineered systems must also adhere to policies, regulations, or standards established by the government, organization, technology, or interfaces [1], [2]. Public or corporate policies are typically documents that use natural-language, such as English to explain different organizational structures, definitions, rules, and regulations of conduct that dictate the resulting system [3]. Gaps in these policies can influence the outcome of the resulting system [4], [5]. Natural-language documents are not completely digital, and thus, they cannot be automatically analyzed or updated [3]. Another issue is that policies are not static entities; rather they are, dynamic, complex interdependent mazes of natural-language documents designed to accomplish a specific mission [6]. Most public, academic, and healthcare policies tend to be nonmachine-readable natural-language documents, which

makes it difficult to dynamically change all the interdependent policies to achieve congruency in communication and implementation.

Although there have been attempts to use structured natural language in privacy policy development and implementation [6], [7], public, education, and healthcare policies largely remain unstructured, nonmachine readable, natural-language documents. The privacy policy modeling work helped organizations develop and maintain compliance, and hence, system success. However, most of the public sector has legacy policies that govern a heterogeneous population, not just electronic information; in such cases, it is essential to have some mechanism for natural-language processing (NLP) with the intention of making policies machine readable. NLP helps in processing natural language for use in a machine-readable and analyzable model [8]. Over the past decade, system engineers have been using the tenets of NLP and combining it with modeling representations, resulting in digital twins of documents and systems [9], [10]. The main principles used in modeling natural-language are ontologies, structured natural language, and traceability of interdependence. There are benefits to having structured natural-language, such as machine readability [11], [12], and perhaps future policies can be written using some structural language that is suited to automation and analysis. However, legacy and undocumented institutional policies remain obtuse. In this article, the systems engineering philosophy of moving from a document-centric to a model-centric system is used with the aim of capturing, analyzing, and modeling policy content.

Systems engineering has always been a document-centric process, necessitating documents such as requirements, concepts, architectures, and test documents. These documents are written using natural-language such as English, and are published electronically in Word or PDF formats. Although Word and PDF files are electronic, they have to be interpreted by a human and are, therefore, not digital or machine readable, and will, thus, face similar problems to those of policy documents. Over the past decade, systems engineering has made constructive efforts to move from document-centric approaches to being more model-centric. This has had positive effects when it comes to understanding interdependencies and complexity across systems [13]. In recent years, systems modeling language (SysML) has been the preferred modeling language for model-based systems engineering (MBSE); it not only models all aspects of complex systems (structure and behavior), but also provides analysis capabilities [14], [15]. SysML is based on object-oriented design principles that reduce complexity via

Manuscript received September 5, 2019; revised January 8, 2020 and April 29, 2020; accepted June 12, 2020. Date of publication; date of current version. This work was supported by the U.S. Department of Veterans Affairs under BOA VA777-16-G-0020. This paper was presented in part at the 15th Annual Conference on Systems Engineering Research, Los Angeles, CA, USA, March 23–25, 2017. (Corresponding author: Shamsnaz Virani Bhada.)

The authors are with the Worcester Polytechnic Institute, Worcester, MA 01609 USA (e-mail: ssvirani@wpi.edu; rkrishnan2@wpi.edu).

Digital Object Identifier 10.1109/JSYST.2020.3003204

## Why Systems Engineers Model

- **Reduce -**
  - Ambiguity,
  - Vagueness,
  - Complexity,
  - Omission,
  - Duplication,
  - Wordiness,
  - Inappropriateness
- **Increase -**
  - Clarity
  - Simplicity
  - Understanding
  - Communication
  - **Analytics**

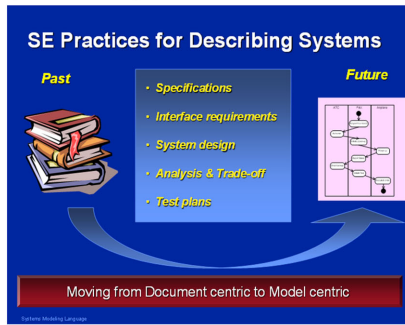


Fig. 1. Case for modeling policy adapted from Friedenthal *et al.* [18] and Mavin *et al.* [19].

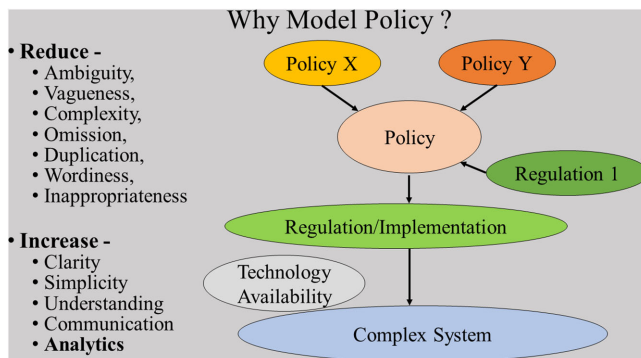


Fig. 2. Case for modeling policy.

inheritance, hierarchies, and polymorphism. The transformation from document-centric to model-centric systems not only demands the conversion of documents to models, but also guidelines for successful modeling to achieve an outcome. In some places, this involves a complete transition to model-centric or digital engineering systems, and in others, it is specific to understanding vulnerabilities in the system [16]. The move toward modeling or digitizing has always been outcome-driven; some have modeled documents for better semantics, others for traceability, and still others for analysis [9], [16], [17]. The International Council on Systems Engineering (INCOSE) introduced a model-based systems initiative (see Fig. 1), which highlighted the reasons why systems engineers should use modeling. This article makes a similar case for modeling policy content (see Fig. 2)

As can be seen in Fig. 2, several interdependent policies influenced by administrative philosophy typically result in a regulation that dictates a new or major change in a dynamic complex system that is constantly influenced by technology changes. To build a successful complex system, policy content modeling (PCM) is needed to

- 1) increase policy clarity with policy visualization and traceability;
- 2) add simplicity of organization, understanding, and communication;
- 3) automate policy gap analysis.

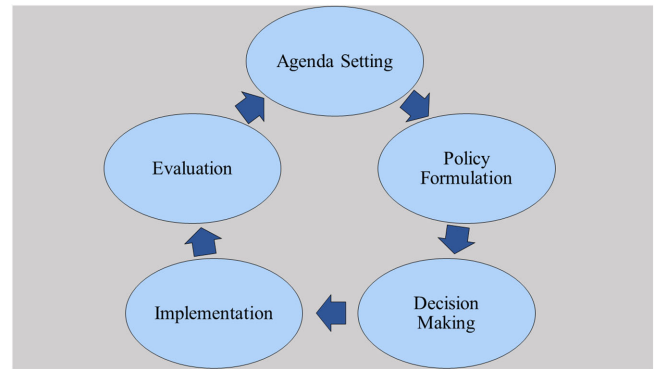


Fig. 3. Policy cycle adapted from [20] and [21].

A complete machine-readable digitized policy model can be achieved by means of a framework and a process for modeling policy.

In this article, a framework for PCM and analysis is proposed. The framework represents a conceptual view of the following:

- 1) converting policy content (implicit or natural language) to a digital entity;
- 2) analyzing the digital model for gaps;
- 3) capturing the negative effect of policy gaps (henceforth referred to as “policy toxicity”) in a metric.

This conceptual framework is supported by a step-by-step process to achieve complete policy modeling and analysis. The PCM process is used to identify and analyze gaps in policy content in order to calculate policy toxicity as it negatively affects the resulting system. The framework and approach are applied to the Veterans Affairs (VA) and Worcester Polytechnic Institute (WPI) policies. The VA PCM is conducted to discover toxicity in the policies and WPI policies modeling is done to graphically represent an undocumented policy and toxicity in the policy implementation.

## II. POLICY DEVELOPMENT AND IMPLEMENTATION

Depending on the domain, i.e., public or private policy development, analysis, and implementation is done by decision-makers, lawmakers, economists, social scientists, and political scientists. Most political scientists use some version of the policy cycle shown in Fig. 3 as a framework for policy development [20], [21].

In recent years, there has been an emphasis on more systemic approaches to policy modeling, evaluation, and implementation, such as Overseas Development Institute’s Research and Policy in Development Program (RAPID) [22]. Some agencies, such as the Centers for Disease Control and Prevention (CDC), have a formalized policy cycle framework that institutes policy evaluation as content analysis to check the logic of the policy [23]. According to the CDC, for policy content evaluation, it is necessary to ask: “Does the content clearly articulate the goals of the policy, its implementation and the underlying logic for why the policy will produce intended change? Evaluating the development of a policy helps to understand the context, content, and implementation” [22]. This article develops policy content evaluations and the estimated effects of these evaluations on

the system or program. The goal is to capture policy issues before implementation so the policy is clear and consistent. Policy content analysis has been performed in several social science methodologies [24]–[28], which represent efforts to apply qualitative research techniques to understand language using a coding scheme. Such efforts are essential when it comes to understanding natural-language and analyzing documents for word occurrences or other language patterns, but they fail to visualize and analyze gaps in the content.

This article uses the premise of content analysis and codes content based on the SysML coding rules to achieve machine readability and investigate policy gaps. Although all public and corporate policy dimensions in development, implementation, and evaluation are relevant for a complete policy analysis, this article focuses on two extreme cases: a policy that is already developed and written and a policy that has never been written but resides as institutional knowledge. For the purpose of this article, a policy is a “statement of intent [...] implemented as a procedure or protocol” [29]. The goal of this article is to demonstrate the application of SysML for PCM with the following aims:

- 1) convert policy content (implicit or natural language) to a digital entity;
- 2) analyze the digital model for gaps;
- 3) capture the negative effect of policy gaps (called policy toxicity by this article) a metric.

### III. CONVERTING NATURAL-LANGUAGE TO SYSML

Converting natural-language to SysML models is the first challenge in the endeavor of PCM. MBSE approaches have had some successes in converting requirements documents to requirements models using restricted natural language [12], [19], [30]. Policies are often written by diverse sets of subject matter experts (SMEs), such as government officials in public policy or engineers, as in the case of the Institute of Electrical and Electronics Engineers (IEEE) standards. The language used can be heavily domain dependent, disconnected, and ambiguous. In fact, ambiguity, vagueness, complexity, omission, duplication, wordiness, inappropriate implementation, and lack of testability are problems associated with natural-language requirements documents [19]. Unlike requirements documents, policy documents can be highly diverse in structure and behavior. For example, they may involve “if/then” processes, roles and responsibilities, goals, procedures, coverage criteria or eligible systems, and people or geographical/technical areas. They can also contain frameworks and guidelines without a clear step-by-step description of the process. The requirements document is built with the sole purpose of stating the requirements for the system being built. Policy documents are not created for a single purpose: Some are created as guidelines, some as statements of intent, and others as decision-making guides. Thus, in some way, the nine problems suggested by Mavin *et al.* [19] are magnified when modeling policy. Another major policy-modeling problem is the lack of a boundary between the policy and the system. For example, if it is assumed that a healthcare policy is a system, and it is modeled that way, the result will be structural and behavioral diagrams of that are stated in the healthcare policy, but rather

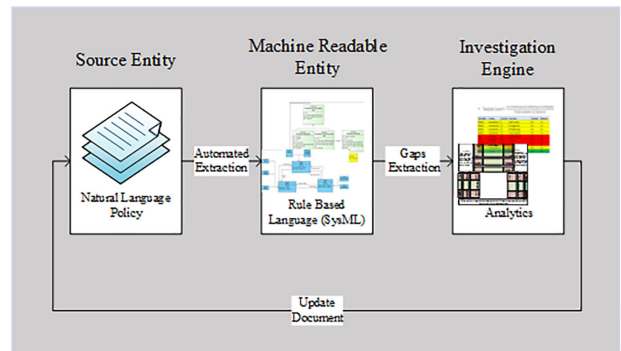


Fig. 4. PCM framework.

than being a representation of the actual healthcare system, they may illustrate what the policy dictates. There can be a major gap between the policy description of the system and the real system. If MBSE is accurately done, using traceability can help find the policy gaps affecting the system. This article proposes a framework for modeling policy and addressing some of the concerns discussed in this section.

### IV. PROPOSED PCM FRAMEWORK AND PROCESS

The PCM framework and process goals are convert policy content, i.e., implicit or natural-language to a digital entity, analyze the digital model for gaps, and capture the negative effects of policy gaps referred to in this article as policy toxicity as a metric. The PCM framework is the conceptual model, and the PCM process is a step-by-step iterative modeling process. Fig. 4 represents the conceptual framework.

The proposed framework is built with three main entities, which are as follows: a source entity, which may be a document or institutional knowledge (source); a digital entity (conversion/transformation); and an investigation engine. In this case, the natural-language source is a policy document or institutional knowledge source. The source entity serves as the system under investigation, and the extraction can be customized to the constructs under examination. For example, if it is necessary to examine the policy structure, one can customize the extraction of only the structure from the natural-language. The digital entity then essentially converts the natural language document to a machine-readable entity that is representative of the source under examination. The investigation engine extracts the necessary information from the digital entity to deduce the location of gaps and changes needed, then issues and updates the natural language document. Once the updates are received by the natural document, it can update and resolve the existing flaws or gaps. This article also defines a process for implementing the PCM framework. Fig. 5 shows the four main steps in the process, which are source selection, modeling strategy, investigation, and verification. Table I shows the relationship between the framework and the process; it also outlines all of the major decisions and outcomes that can be accomplished in each step of the PCM process.

As can be seen from Table I and Fig. 5, source selection involves defining the basic parameters and team, the modeling

TABLE I  
RELATIONSHIP BETWEEN THE PCM FRAMEWORK AND PCM PROCESS

Framework: Source Entity			
PCM Process:	Decisions	Outcomes	Comment
Source Selection	<ol style="list-style-type: none"> <li>1) What policy is being selected?</li> <li>2) What is the investigation question?</li> <li>3) Who are the team members?</li> <li>4) Who are the domain or subject matter experts?</li> </ol>	<ol style="list-style-type: none"> <li>1) Policy to be modeled is selected</li> <li>2) Investigation question is stated and agreed-on by all members</li> <li>3) Team members roles and responsibilities are decided</li> </ol>	Policies can be implicit, explicit, written in natural-language, or under development. Before modelers start modeling the policy, it is imperative to decide which policy and what in that policy is to be investigated. Explicit team member roles and responsibilities help divide the work of modeling and model verification.
Framework: Machine Readable Entity			
PCM Process:	Decisions	Outcomes	Comment
Modeling Strategy	<ol style="list-style-type: none"> <li>1) How much and when the automation is used?</li> <li>2) What modeling process is employed?</li> <li>3) What modeling technique is used?</li> <li>4) What modeling construct is used for evaluation?</li> </ol>	<ol style="list-style-type: none"> <li>1) Automation is employed for large policies and added with some level of verification.</li> <li>2) Iterative process is usually employed, but Big Bang helps when a policy involves basic definitions of processes</li> <li>3) SysML is preferable for descriptive-modeling of policy</li> <li>4) Structural and behavioral representations are selected as modeling constructs</li> </ol>	Machine-readable entity development can be done in many ways. This paper highlights modeling policy using SysML, especially the structural and behavioral aspects of the policy.
Framework: Investigation Engine			
PCM Process:	Decisions	Outcomes	Comment
Investigation	<ol style="list-style-type: none"> <li>1) How are the gaps classified?</li> <li>2) How are they identified or detected in the model?</li> <li>3) What analysis will help answer the investigation question?</li> </ol>	<ol style="list-style-type: none"> <li>1) Build a table with the domain experts that defines explicit gaps that exist in the policy and their severity or priority</li> <li>2) All modelers should follow a protocol of tagging the gaps in the policy dictated by the classification table</li> <li>3) Assign gap weights based on their severity and identify normalization factor for policy</li> <li>4) Employ policy toxicity formula to calculate gap density</li> </ol>	Investigation heavily relies on the investigation question decided in the source selection step. The goal here is to classify, identify, trace, and quantify impact of gaps on the policy.
PCM Process:	Decisions	Outcomes	Comment
Verification	<ol style="list-style-type: none"> <li>1) How many times the model is reviewed and by whom?</li> <li>2) How to keep models unbiased?</li> </ol>	<ol style="list-style-type: none"> <li>1) The models will be peer-reviewed by other modelers and final review with SME</li> <li>2) SME review will focus on models being unbiased by the underlying policy</li> </ol>	This is an essential step for model verification and is usually undertaken by the entire team.

strategy helps shape the building of the digital entity, investigation focuses on collecting relevant data and analyzing them, and verification should focus on how true to the source the digital entity is. Although the four steps are represented and described sequentially, the activities of the last three are often done iteratively. The following sections elaborate further on each step.

Depending on the domain, source selection can be easy or the most-difficult aspect of PCM. Subject to the domain, policy can be a natural-language document or implicit information that can be difficult to capture. Therefore, it is important to develop some basis for the factors mentioned as follows.

#### A. Selection Parameters

The selection parameters include the following:

- 1) a new policy document under development or an old policy document needing review;

- 2) size, complexity, risk, eligible population, and domain;
- 3) implicit department-wide policy or government-regulated standards;
- 4) the following investigation question: What are we trying to answer with this effort?

#### B. Key Team Member

The key team member considerations are as follows:

- 1) number of modelers;
- 2) sponsor or champion.

#### C. Subject Matter Experts

The following SMEs are most helpful:

- 1) if possible, the policy author;
- 2) the policy authority.

Once the policy and key team members are selected, define the modeling strategy. The main decision in this strategy is to

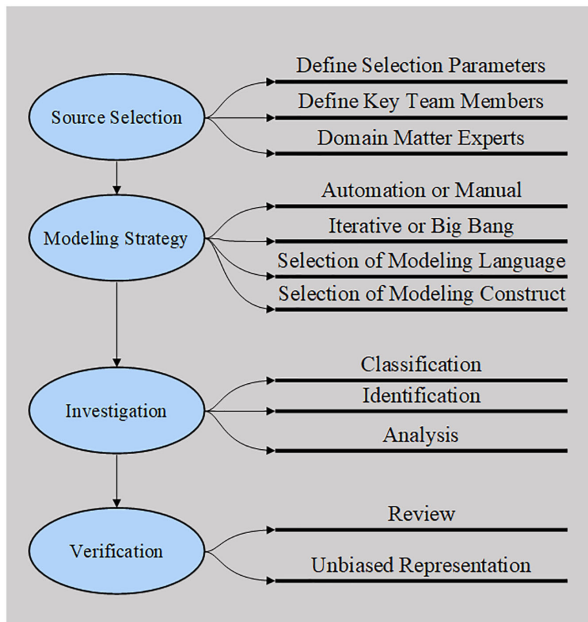


Fig. 5. PCM process.

identify whether an NLP engine can automate some modeling. This decision is based on the size and complexity of the policy. If there is one small policy (i.e., 1–10 pages, standard A4 size, written in English), a manual modeling process will be sufficient; if there are more than ten pages, it may require some automation to generate some initial models. Other important decisions include selecting a modeling process, i.e., iterative model development with several reviews or one final model without an iterative review, and selecting a modeling technique, i.e., SysML, discrete event, etc., that the team finds most relevant for the selected policy. At this stage of the modeling strategy, it is necessary to define the modeling construct that will help answer the investigation question. In the source-selection step, the modeling construct is defined based on this question.

The investigation step involves the design team and SMEs, who decompose the investigation question into gaps, failures, and violations that may be present in the source. These are usually commonly observed gaps, failures, or violations that might be either pre-existing or known issues, such as ambiguity in a natural-language document. The gaps are then classified in some sort of ranking or priority scheme to demonstrate effects of the gaps on the underlying population. The modeling team examines the model, identifies each gap, and tags the modeling elements with the type of gap that is present. In order to understand the collective effect of the gaps on the policy, a policy toxicity calculation can be accomplished by employing (1) or any other form of analysis technique that will help answer the investigation question. The policy toxicity formula introduced in this article is a simplistic calculation of gap impact on the policy and its underlying population. The formula should be modified to serve the investigation question and the source entity

$$\text{Policy Toxicity} = \sum_{i=1}^T \frac{\alpha_i n_i}{N} \quad (1)$$

where  $T$  is the number of weight-levels for gaps,  $\alpha$  is the weighting factor,  $n$  is the number of gaps of a given weight-level, and  $N$  is the normalization factor. The authors understand that the normalization factor  $N$  is subjective and can be manipulated by the investigators, resulting in irregular policy toxicity. Since policies do not follow a standard representation (i.e., language, font, etc.), the selection and implementation of the normalization factor can vary. This article introduces a policy toxicity formula, and future research will be dedicated to formula validation.

Verification focuses on closeness of the machine-readable entity with the source. Here, the team and SMEs review the model to ensure closeness to the source. The modeling team’s goal is to model the source, not the underlying system, people, and process. To this end, the modeling team and SME should be vigilant in terms of being true to the source and letting the model be an unbiased representation of the policy. The proposed framework and process are applied to two distinct areas: a review of VA policies and modeling implicit institutional processes.

## V. POLICY MODELING APPLICATION

Policies can be explicit or implicit. Explicit policies are natural-language documents, and implicit policies are undocumented institutional knowledge. The framework and process are applied to both policy types. In the VA application, this article models four policy documents in order to analyze and build a metric that represents overall gaps in the policy; this metric is called “policy toxicity.” In the WPI application, this article graphically represents an undocumented policy and calculates the toxicity in policy implementation.

### A. Veterans Affairs

As shown in Fig. 5, policy selection is the first step in the modeling process. The Veterans Engineering Resource Center (VERC) needed some insight into Government Accountability Office (GAO) [4] reporting, which repeatedly identified ambiguous policies as a major issue with the VA. They identified three policies in the Veterans Health Affairs (VHA) Procurement and Logistics Office (P&LO) to investigate. The key members of the modeling process were the research team at WPI and members from the VERC (who were also the SMEs). The VA policy dealt with VHA inventory management, logistics management, and standardization of equipment. These handbooks and directives provide operating procedures and requirements for personnel in their respective offices and departments. Handbook 1761.1 [31] dealt with Standardization of Supplies and Equipment, Handbook 1761.02 [32] was on VHA Inventory Management, and Directive 7002 [33], and Handbook 7002 [34] were related to Logistics Management Policy. Thus, the investigation question is as follows: “How can we trace and calculate qualitative issues, such as ambiguity, inconsistency, and incompleteness, in the given set of written policies?” This generic investigation question is further broken down, as shown in Table II, by defining the gap weight and other classifiers. The process is highly iterative, and Table II is the final version of classifying the gaps. The investigation question led to allocating gaps as structural and/or

TABLE II  
CLASSIFICATION TABLE FOR GAPS

Gap Code	Gap Name	Gap Weight	Property	S	B
1	Role with no function	Medium	Incompleteness	Yes	Yes
2.1	Response required without metric	Medium	Incompleteness	Yes	Yes
2.2	Response required without verification	Medium	Incompleteness	Yes	Yes
3.1	No trigger event	Medium	Incompleteness	No	Yes
3.2	Misplaced trigger	Medium	Incompleteness	No	Yes
3.3	Gaps in sequence	Medium	Incompleteness	No	Yes
3.4.1	Conflicting time target	High	Inconsistent	Yes	Yes
3.4.2	Conflicting role target	High	Inconsistent	Yes	Yes
3.4.3	Requirement	Low	Inconsistent	Yes	Yes
4	Function with no role	Medium	Incompleteness	No	Yes
5	Unnecessary words	Low	Verbosity	Yes	No
6	Vague language	Low	Ambiguity	Yes	No
7	Undefined term	Medium	Ambiguity	Yes	No
8	Ref. other document	Low	Reference	Yes	No
9	Redundancy	Low	Verbosity	Yes	No
10	Misplaced information	Low	Inconsistent	Yes	No
11	Role not listed	Medium	Inconsistent	Yes	No

S - Structure, B - Behavior.

behavioral, which resulted in selecting SysML as the modeling language for this project.

This article used SysML as the modeling language and a mix of manual and automation for the modeling process. In this article, “manual process” means that the modeling team had to manually make the resulting models, whereas automation means that the team used algorithms to read a natural-language document and output some SysML diagrams. In the manual process, the team reads a section of the policy, identifies the diagram that best describes it, and populates the diagram with its corresponding elements (SysML blocks: requirement, activity, role, etc.). This approach worked well for the smaller policies, that is, 1761.01 and 1761.02, which totaled 20 pages each. However, the 7002 Handbook was a 150-page policy, and adopting a manual process would have been time and resource intensive; hence, a semiautomated process was adopted for this policy. An automated process was favorable because the handbook only contained requirements, and VERC staff had already classified sentences in the policy as requirements. Therefore, developing an algorithm to convert text to requirement blocks helped automate the process and reduce the lead time for modeling.

With the SysML diagrams built, model the structural and behavioral aspects of the policy. To develop the organizational structure, the policy was divided into different sections following the same structure as the Table of Contents to ensure an unbiased representation of the policy. A package diagram is ideal for developing organizational structure, where a package represents a section of the policy; each package contains one or more diagrams that further model the section. Fig. 6 illustrates the organizational structure of policy 1761.02.

To identify the different actors in the policy and their associated responsibilities, a roles and responsibilities diagram was created. A SysML “block” was used to model each role, with the actors’ responsibilities listed in the block. Linking each responsibility to a requirement (in a requirements diagram) or an action (in an activity diagram) in the section or packages in

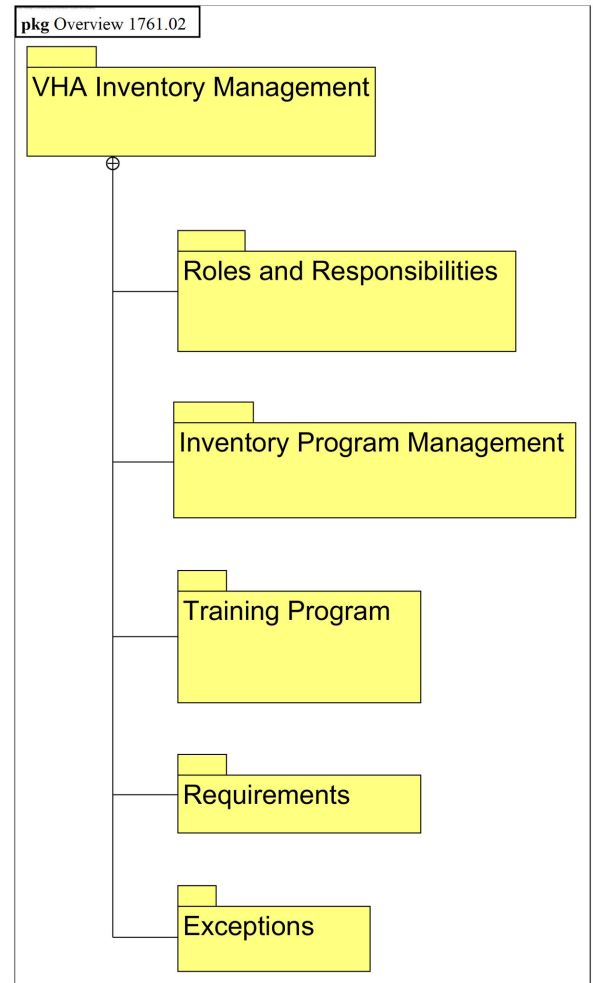


Fig. 6. Organizational structure of policy 1761.02 using the package diagram.

which it exists provided traceability to the model. Any transfer of information or documents between roles was shown using directional arrows, with a brief description of what was being transferred. Fig. 7 shows a roles and responsibilities diagram for Policy 1761.02. Finally, each block was stereotyped as executive, management, or office, and color coded to provide a visual aid. Fig. 8 shows the details of the roles and responsibilities as described in Policy 1761.02.

Since a process is a behavior, this article modeled the natural-language of processes using activity diagrams. A process was further broken down into a series of actions and decisions, with each action linked to the actor executing it. These diagrams helped elucidate the flow of events in the process and made it easier to identify any gaps in sequences or missing actors for actions. Fig. 9 illustrates an activity diagram for Policy 1761.02.

After completing a diagram, i.e., requirement, activity, etc., and reviewing it with the SMEs, this article rescanned it element-wise to find and tag gaps, if present. This was part of the investigation aspect of the methodology, and it is explained as follows.

The investigation step involves classification, identification, and analysis of gaps present in the policy. Classification involves

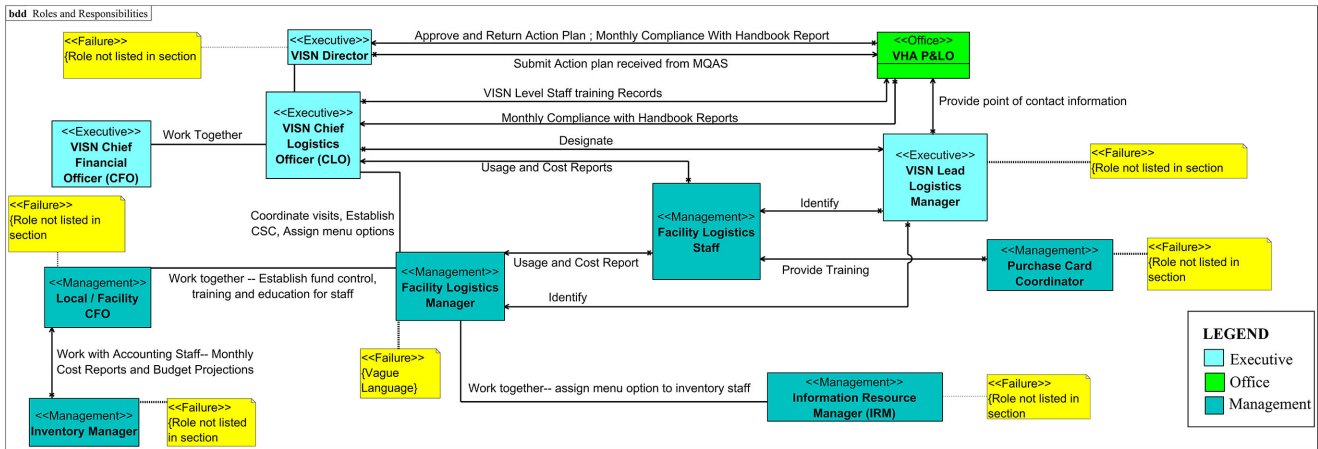


Fig. 7. Roles and responsibilities diagram.

<<Executive>> <b>VISN Chief Logistics Officer (CLO)</b>	
<<Explicitly Mentioned>>	+Represent VISN Director on logistics related matter
<<Explicitly mentioned>>	+Manage logistics data()
<<Explicitly mentioned>>	+Facilitate sound business practices()
<<Explicitly mentioned>>	+Provide guidance to local logistics managers()
<<Explicitly mentioned>>	+Assess inventory management programs(At : Eac...
<<Explicitly mentioned>>	+Improve understanding of logistics goals(Work wit...
<<Explicitly mentioned>>	+Ensure compliance(With : VA Directives, policies, ...
<<Explicitly mentioned>>	+Developing and Implementing VISN strategies to i...
<<Explicitly mentioned>>	+Improve cost control and and Reporting efforts(Wit...
<<Explicitly mentioned>>	+Facilitate communication between(Field Organizati...
<<Explicitly mentioned>>	+Fix facility visits(Coordinate with : Facility Logistics Manager, Parameter134 ...
<<Explicitly mentioned>>	+Review(Usage & Cost Report, From : Facility Logistics Staff, Parameter135, ...
<<Explicitly mentioned>>	+Establish Commodity Standards Committee(At : VISN Level, Parameter181 ...
<<Explicitly mentioned>>	+Report VISN level Staff Training records(To : VHA P &LO, Parameter182 : V...
<<Explicitly mentioned>>	+Supervise designation(Of : VISN Lead Logistics Manager, Parameter183 : ...
<<Explicitly mentioned>>	+Review(Monthly Compliance with Handbook report, From : VHA P &LO, Par...
<<Explicitly mentioned>>	+Evaluate GIP Program()
<<Explicitly mentioned>>	+Plan staffing needs()
<<Explicitly mentioned>>	+Designate(VISN Lead Logistics Manager, Parameter186)
<<Explicitly mentioned>>	+Assisting with formulation of VHA logistics policies...
<<Explicitly mentioned>>	+Help assign menu options to inventory staff(With : Facility Logistics Manager...
<<Explicitly mentioned>>	+Work with clinical groups to improve understandin...

<<Executive>> <b>VISN Lead Logistics Manager</b>	
<<Explicitly mentioned>>	+Point of contact for Network Inventory management and Standardization issues
<<Explicitly mentioned>>	+Provide point of contact information(To : VHA P &LO, Parameter202 : VHA P &LO)
<<Explicitly mentioned>>	+Identify logistics manager at each facility(Within network, Parameter203)
<<Explicitly mentioned>>	+Identify all employees at each facility(Inventory management staff, Parameter204)
<<Explicitly mentioned>>	+Create VISN wide Outlook email groups()
<<Explicitly mentioned>>	+Schedule monthly network meetings()

Fig. 8. Details of roles and responsibilities.

creating a comprehensive list of gaps applicable to the policy under consideration. Virani and Rust [35] developed a set of gaps for policies related to healthcare. While those gaps are not exhaustive, they can be applied to most policies. Using them as a base, Krishnan *et al.* [1] introduced gaps similar to the failures identified in the GAO report, which was used by the research team and SMEs to create a list of 11 gaps (see Table II). Each gap was classified into one of five types: incompleteness, inconsistency, ambiguity, verbosity, or reference. Each of the gap types was then classified by importance or gap weight and the gap category (structural or behavioral). Table II shows the list

TABLE III  
GAP TRACE

Gap Tagged Value	Description
Gap Code	3.4.2
Description	Two goals for the same target requirement
Source	Frequency of meetings
Source Type	Requirement
Text Source	Ch. 6b, p. 5
Gap weight	3
Policy	1761.1

of gaps used when modeling each of the four policies. The first column in Table II represents the “gap weight,” which classifies policy gaps that have a greater negative influence on the system from those that have a lesser effect.

The next step is identification, which involves tracing each gap as it exists in the SysML models. As shown in Table III, each gap has a “code,” which makes it easier to highlight gaps in the model. Hence, when an element in the model contains a gap, it is tagged with the code corresponding to that gap. The gap weight and gap code are both crucial to the tagging process, and they are useful when running a qualitative analysis on the finished models. For example, in Table III, a requirement has a “function with no role” gap; that element is tagged in a yellow box with the <<Failure>> stereotype. The tag states the gap name and why the modeler thinks it is a gap. To improve visual identification of the gap in the model, a yellow box with a description of the gap is linked to the element. The modeler can add further text to help future modelers understand the reason for the tagging. The advantage of such tagging is that most SysML software programs have the functionality to run Unified Modeling Language (UML) scripts on the models; using these, this article can obtain a wide range of data on the gaps and associated elements. The trace and aggregate of the gaps can then be used to answer the investigation question, “How can we trace and calculate qualitative issues, such as ambiguity, inconsistency, and incompleteness, in the given set of written policies?” The trace is accomplished with the gap-name tags associated with each policy. The description section of the tags

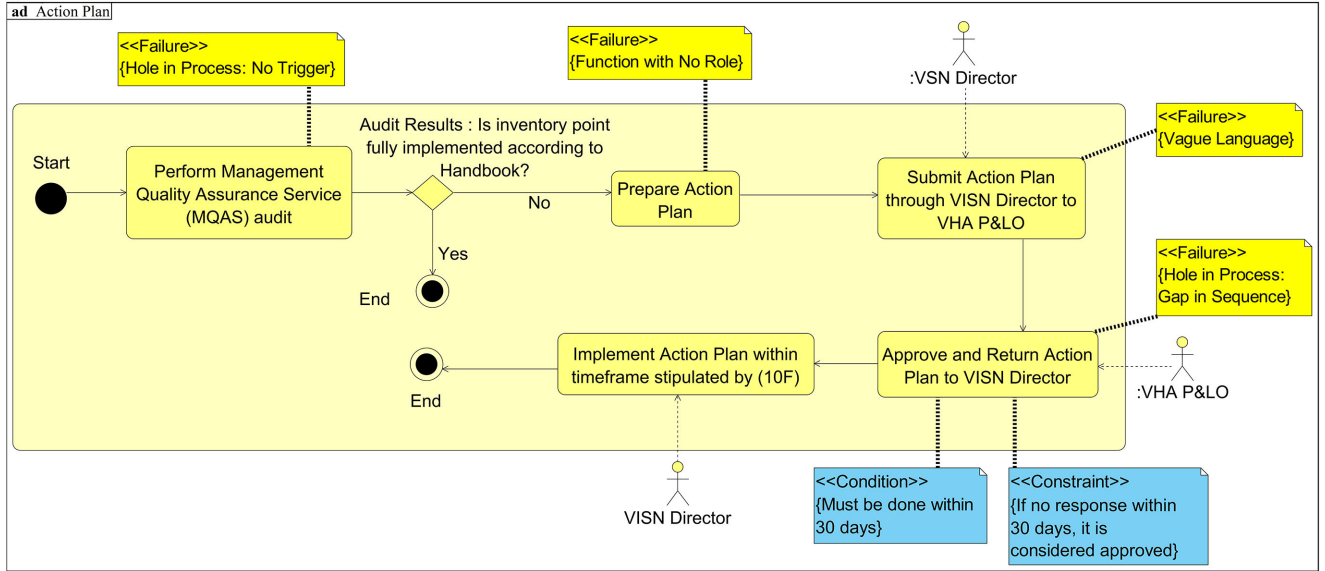


Fig. 9. Activity diagram.

contains the page number, paragraph number, and reason for the gap. All this helps trace the gap to the text location and change the paragraph language. For example, Table II shows the gap code, description of the gap, and tracing of the gap to a specific frequency of the meeting section in the document, which is modeled as a requirement in the SysML model and written about in Chapter 6, Section B, page 5 of the 1761.1 policy document.

Tracing gaps provide evidence to change language, structural, and behavioral gaps in a natural-language document. The collective influence of these policy gaps is calculated by policy toxicity; this term because the collective influence of these gaps results in healthcare hazards, such as the 2016 VA scandal related to massive wait times in their healthcare system [36]. Just as the density of toxins in food or water is reported in toxicology reports, policy toxicity represents the per-page density of gaps in the policy document. Policy toxicity in the VA policies helped to determine the per-page density of high, medium, and low gaps in each policy. The total number of pages is used as a denominator in the policy toxicity formula, because the formatting of pages between policies is consistent in the VA, which defines the policy domain. In other applications of PCM, the denominator could be something else. The toxicity formula can change based on the policy domain, investigation question, and modeling construct. In the VA application of the PCM, the policy toxicity is calculated using (1), as follows:

$$\text{Policy Toxicity} = \frac{\alpha_1 \times n_1 + \alpha_2 \times n_2 + \alpha_3 \times n_3}{\text{Total number of pages}} \quad (2)$$

where  $T(=3)$  is the gaps classified as high, medium, and low, as shown in Table II,  $\alpha$  is the gap weight-level factor, for the VA policies,  $\alpha_1 = 0.2$ ,  $\alpha_2 = 0.3$ , and  $\alpha_3 = 0.5$ ;  $n$  is the number of gaps of a given weight-level and  $N$  is the normalization factor, for the VA policies, this is the total number of pages for the policy.

TABLE IV  
POLICY TOXICITY ANALYSIS FOR VA APPLICATION

Policy	1761.01	1761.02	7002	Directive 7002
Total number of pages (N)	21	30	132	20
High gaps ( $n_3$ )	8	18	13	7
Medium gaps ( $n_2$ )	21	46	14	20
Low gaps ( $n_1$ )	43	37	34	42
Policy Toxicity	0.9000	1.0067	0.1326	0.8950

For the four VA policies, the tags labeled “failure” from the SysML diagrams were exported and used the abovementioned formula to calculate policy toxicity (see Table IV).

In the verification phase, SMEs from VERC verified the finished model. They checked whether the elements used for modeling each section of the policy were appropriate (i.e., if the sentence in the policy was in fact a requirement or process) and the validity of the gap identified for an element, in order to rectify any potential misinterpretation on the modelers’ behalf. Based on this analysis, VERC recommended that the P&LO office at the VA should change the policy language. The other application of policy modeling was for an undocumented or implicit policy at the WPI.

### B. WPI Application

The policy modeling approach described in the previous section was applied to a university-wide educational project process at WPI. As mentioned above, there was no official policy or documentation for guiding this process, which was run by experienced users employing implicit institutional knowledge alone. For this article, a university-wide student registration process was selected to demonstrate application of this methodology. The process dealt with registering a student or group of students for a course in which they will work on a project that ran for the duration of one or two quarters. The process began



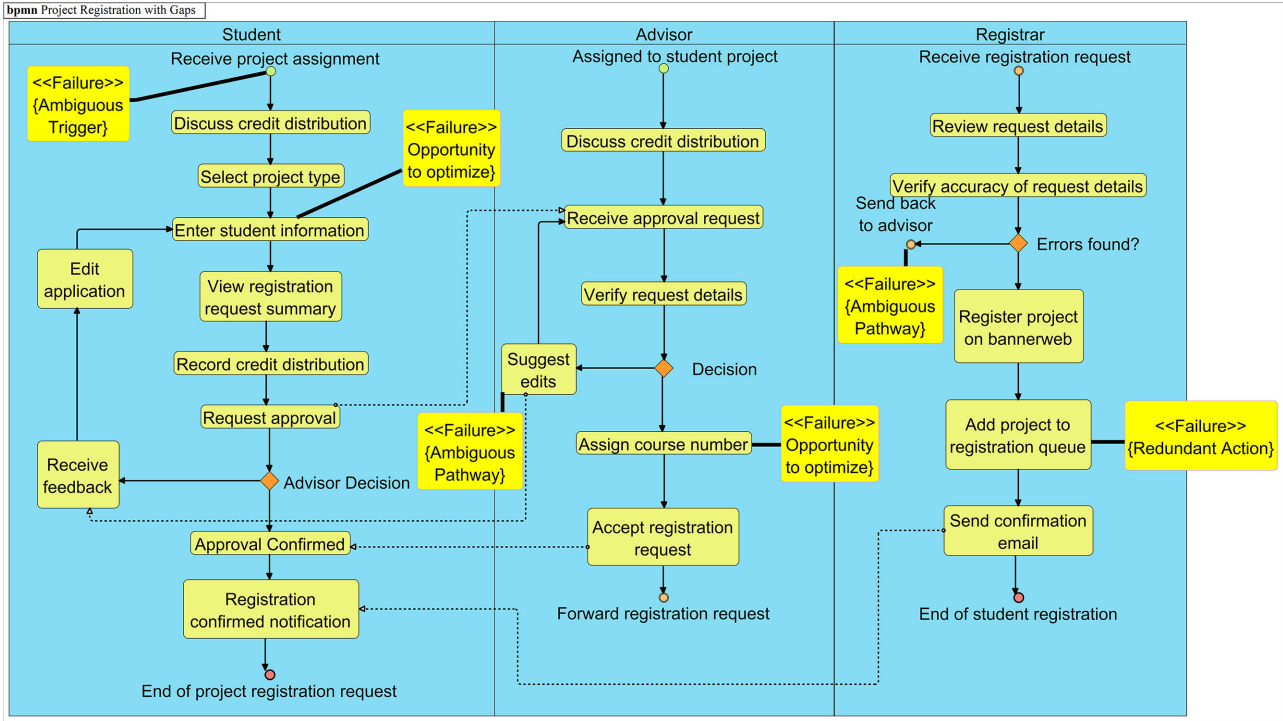


Fig. 10. Business process diagram after gap investigation, highlighting the gaps identified.

with the student receiving the project assignment. The next step for the student was to log in to an online portal and enter the project details along, with the number of credits. These details were submitted to the project advisor, who forwarded it to the university registrar to verify and review. Once all the details were verified, the registrar officially registered the student in the course. The expected outcome of policy modeling here was to highlight any inconsistencies in the process, identify sources of ambiguity, and digitally document the process in the form of a model. The following section describes the application of the methodology to this process in further detail.

1) *Policy Selection:* As explained earlier, the process being modeled was the student-project registration process. The key members were the stakeholders of the process: the authors as the modeling team, a student with prior experience with the process, a faculty member who served as a student advisor for such projects, and staff from the university registrar’s office. The staff from the registrar’s office also served as the SMEs, because these staff members were the most knowledgeable about the online portal being used and the steps in the process. Since the process was not documented, an extra step was added to the methodology to gather information about the process. This involved using some elicitation techniques: scheduling interviews with the various stakeholders and running a mock registration using a dummy project and student. The process was modeled and documented based on this information.

2) *Modeling Strategy:* To utilize the strengths of modeling process, business process model and notation was selected as the modeling language. A manual and final model without any iterative feedback process was chosen due to the small size

TABLE V  
GAP CLASSIFICATION FOR THE PROJECT REGISTRATION PROCESS

Gap Code	Gap Name	Gap Weight	Property	S	B
1.1	Conflicting Target - Time	High	Inconsistent	No	Yes
1.2	Conflicting Target - Role	High	Inconsistent	No	Yes
2	Ambiguous Trigger	High	Ambiguity	No	Yes
3	Redundant Action	Low	Upgrade	No	Yes
4	Ambiguous Pathway	High	Ambiguity	No	Yes
5	Opportunity to Optimize	Medium	Upgrade	No	Yes

S - Structure, B - Behavior.

of the model and relatively short turnaround time to build it. By developing business process diagrams, only the behavioral aspect of the policy was modeled. Fig. 10 shows the business process diagram that models the student registration process.

3) *Investigation:* This step in the PCM process dealt with classifying, identifying, and analyzing the gaps in the model. It is important to note that the procedure for gap classification and gap identification for this article was identical to that used in the previous application.

4) *Classification:* After consultation with the SMEs, a list of gaps that were applicable to the project registration process was identified. The gaps aid in capturing any inconsistency, ambiguity, or opportunity to optimize the process. Table V shows the list of gaps used for this case study. To be consistent, the format was identical to the classification used in Table II.

5) *Gap Identification:* Once the initial model was built, the modelers reviewed it to look for any gaps classified in the previous section. If a gap was present, the element was tagged with the corresponding gap code. Beyond the advantage of

TABLE VI  
POLICY TOXICITY ANALYSIS FOR WPI APPLICATION

Policy	Student-Project Registration
Total number of Actions (N)	22
High gaps ( $n_3$ )	3
Medium Gaps ( $n_2$ )	2
Low Gaps ( $n_1$ )	1
Policy Toxicity	0.1045

visually representing the presence of a gap, tagging also helps to automate the analysis of gaps in a model or over multiple models. Fig. 10 shows a process model with the gaps identified.

6) *Toxicity Analysis*: With the gaps in the model identified, the final step was to calculate the policy toxicity. This is an indicator of poor process construction, wherein higher toxicity corresponds to greater negative consequences in policy implementation. For the model shown in Fig. 10, the policy toxicity is calculated based on (1), as follows:

$$\text{Policy Toxicity} = \frac{\alpha_1 \times n_1 + \alpha_2 \times n_2 + \alpha_3 \times n_3}{\text{Total Number of Actions}} \quad (3)$$

where  $T(=3)$  is the gaps classified as high, medium, and low, as shown in Table IV;  $\alpha$  is the gap weight-level factor, for the WPI policies,  $\alpha_1 = 0.2$ ,  $\alpha_2 = 0.3$ , and  $\alpha_3 = 0.5$ ;  $n$  is the number of gaps of a given weight-level, and  $N$  is the normalization factor, for WPI policy, this is the total number of actions, as seen in Table VI.

7) *Verification*: The final step in the methodology is reviewing the model with the SMEs; this was done by reviewing the model with the staff from the registrar's office. Verification helps with model accuracy, unbiased representation of the process, and to confirm the validity of identified gaps. In this case, an evidence-based discussion on resolving gaps was conducted with the staff of the registrar's office and the information technology team to automate the implicit institutional process.

## VI. CONCLUSION

This article proposed and demonstrated a framework for policy modeling and analysis from the systems engineering perspective. The framework and approach were both based on a model-centered engineering philosophy of eliminating ambiguity from the critical artifacts of a complex system. To that end, the proposed framework and approach were applied to specific VA and university policies. The VA application resolved the issue of ambiguity by modeling the structure and behavior of policy content, then tracing and analyzing gaps to calculate the gap density per page, which in this article was called policy toxicity. In the case of implicit or undocumented university processes, this article sought to identify the implementation gaps and build a toxicity index. In this case, the toxicity index informed the SMEs the location and severity of the implementation issue, and provided a recommended resolution. The focus was on building visibility, traceability, and analyzability into entities that are critical to the success of a complex system. The VA application revolved around policy as is, while the university process application focused on making the implicit explicit. Both applications showed that the PCM framework and approach

result in evidence-based analysis that can then be used to make informed decisions. These decisions are further facilitated by tracing them back to their exact location in the artifact to reduce overall ambiguity, thereby facilitating better management of the policy or process. The focus of future research will be on implementing this framework for other policies and building a battery of approaches, metrics, and analyses for different complex system critical artifacts.

## ACKNOWLEDGMENT

The authors would like to thank Dr. T. Rust at the New England VERC and R. Brown at WPI Information Technology.

## REFERENCES

- [1] R. Krishnan, S. Virani, and R. Gasoto, "Discovering toxic policies using MBSE constructs," in *Disciplinary Convergence in Systems Engineering Research*. Berlin, Germany: Springer, Nov. 2017, pp. 341–350.
- [2] O. L. De Weck, D. Roos, and C. L. Magee, *Engineering Systems: Meeting Human Needs in a Complex Technological World*. Cambridge, MA, USA: MIT Press, 2011.
- [3] J. Hemerly, "Public policy considerations for data-driven innovation," *Computer*, vol. 46, no. 6, pp. 25–31, 2013.
- [4] D. A. Draper, "Veterans affairs health care: Addition to GAO's high risk list and actions needed for removal," Government Accountability Office, Washington, DC, USA, Tech. Rep. GAO-15-580T, 2015.
- [5] P. C. Light, "A cascade of failures: Why government fails, and how to stop it," *Center for Effective Public Management*, Washington, DC, USA: Brookings Institution, 2014.
- [6] C. A. Brodie, C.-M. Karat, and J. Karat, "An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench," in *Proc. 2nd Symp. Usable Privacy Secur.*, 2006, pp. 8–19.
- [7] C. Brodie *et al.*, "The coalition policy management portal for policy authoring, verification, and deployment," in *Proc. IEEE Workshop Policies Distrib. Syst. Netw.*, 2008, pp. 247–249.
- [8] R. K. Taira, V. Bashyam, and H. Kangarloo, "A field theoretical approach to medical natural language processing," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 4, pp. 364–375, Jul. 2007.
- [9] A. M. Madni and M. Sievers, "Model based systems engineering: Motivation, current status, and research opportunities," *Syst. Eng.*, vol. 21, no. 3, pp. 172–190, 2018.
- [10] T. D. West and A. Pyster, "Untangling the digital thread: The challenge and promise of model-based engineering in defense acquisition," *Insight*, vol. 18, no. 2, pp. 45–55, 2015.
- [11] J. T. Adams, K. Bauer, A. Hardcastle, D. Grunwald, and D. Sicker, "Automated tracking of online service policies," in *Proc. Telecommun. Policy Res. Conf.*, 2010. [Online]. Available: <https://ssrn.com/abstract=1989112>, Accessed on: Aug. 15, 2010.
- [12] G. Zhang, T. Yue, J. Wu, and S. Ali, "Zen-RUCM: A tool for supporting a comprehensive and extensible use case modeling framework," in *Demos/Posters/StudentResearch@ MoDELS*, 2013, pp. 41–45.
- [13] D. Harvey, M. Waite, P. Logan, and T. Liddy, "Document the model, don't model the document," in *Proc. Syst. Eng./Test Eval. Conf. 6th Asia Pac. Conf. Syst. Eng.*, 2012.
- [14] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML: The Systems Modeling Language*. San Mateo, CA, USA: Morgan Kaufmann, 2014.
- [15] "OMG Systems Modeling Language (SysML)," 2018. [Online]. Available: <https://sysml.org/>
- [16] J. B. Reid, "Assessing and mitigating vulnerability chains in model-centric acquisition programs," S.M. thesis, Dept. Aeronaut. Astronautics, Massachusetts Inst. Technol., Cambridge, MA, USA, 2018.
- [17] M. Jackson and M. Wilkerson, "MBSE-driven visualization of requirements allocation and traceability," in *Proc. IEEE Aerosp. Conf.*, 2016, pp. 1–17.
- [18] S. Friedenthal, R. Griego, and M. Sampson, "INCOSE model based systems engineering (MBSE) initiative," in *Proc. Int. Council Syst. Eng. Symp.*, 2007, vol. 11.
- [19] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (EARS)," in *Proc. 17th IEEE Int. Requirements Eng. Conf.*, 2009, pp. 317–322.

- [20] J. E. Anderson, *Public Policymaking: An Introduction*. Boston, MA, USA: Wadsworth/Cengage Learning, 2011.
- [21] P. Bridgman and G. Davis, *The Australian Policy Handbook*. London, U.K.: G. Allen, 2004.
- [22] J. Young and E. Mendizabel, *Helping Researchers Become Policy Entrepreneurs*. Overseas Development Institute London, London, U.K., 2009.
- [23] *Brief 1: Overview of Policy Evaluation—Step by Step Evaluating Violence and Injury Prevention Policies*, Centers for Disease Control and Prevention, Atlanta, GA, USA, 2016.
- [24] H.-F. Hsieh and S. E. Shannon, “Three approaches to qualitative content analysis,” *Qualitative Health Res.*, vol. 15, no. 9, pp. 1277–1288, 2005.
- [25] “Content analysis methodology & prominent scholars—Terry college of business,” [Online]. Available: <https://www.terry.uga.edu/contentanalysis/research/>, Accessed on: Jun. 24, 2020.
- [26] N. L. Kondracki, N. S. Wellman, and D. R. Amundson, “Content analysis: Review of methods and their applications in nutrition education,” *J. Nutrition Educ. Behav.*, vol. 34, no. 4, pp. 224–230, 2002.
- [27] J. Lemiengre, B. D. de Casterlé, Y. Denier, P. Schotsmans, and C. Gastmans, “How do hospitals deal with euthanasia requests in Flanders (Belgium)? A content analysis of policy documents,” *Patient Educ. Counseling*, vol. 71, no. 2, pp. 293–301, 2008.
- [28] S. Stemler, “An overview of content analysis,” *Practical Assessment, Res., Eval.*, vol. 7, no. 1, 2000, Art. no. 17.
- [29] K. Brennan, *A Guide to the Business Analysis Body of Knowledge*. International Institute of Business Analysis (IIBA), Toronto, ON, Canada, 2009.
- [30] M. Fockel and J. Holtmann, “A requirements engineering methodology combining models and controlled natural language,” in *Proc. IEEE 4th Int. Model-Driven Requirements Eng. Workshop*, 2014, pp. 67–76.
- [31] *VHA Handbook 1761.1*, Department of Veterans Affairs, VHA Office of Clinical Logistics, Washington, DC, USA, 2003.
- [32] *VHA Directive 1761.02*, Department of Veterans Affairs, VHA Procurement & Logistics, Washington, DC, USA, 2009.
- [33] *VA Directive 7002*, Department of Veterans Affairs, Logistics Policy Supply Chain Management Service and Office of the Executive Director for Acquisition and Logistics, Washington, DC, USA, 2009.
- [34] *VA Handbook 7002*, Department of Veterans Affairs, Logistics Policy Supply Chain Management Service and Office of the Executive Director for Acquisition and Logistics, Washington, DC, USA, 2009.
- [35] S. Virani and T. Rust, “Using model based systems engineering in policy development: A thought paper,” in *Proc. Conf. Syst. Eng. Res.*, 2016, pp. 1–5.
- [36] C. Kalin, “The VA scandal refuses to end,” 2018. [Online]. Available: <https://www.cagw.org/thewastewatcher/va-scandal-refuses-end>



**Shamsnaz Virani Bhada** received the Ph.D. degree in systems engineering from Industrial and Systems Engineering University, Alabama, Huntsville, in 2008.

She is currently an Assistant Professor with the Systems Engineering Program, Worcester Polytechnic Institute, Worcester, USA. Her research interests include applying model based engineering philosophy to safety analysis as well as policy modeling and digitization.



**Rahul Krishnan** received the M.S. degree in robotics engineering in 2017, with Worcester Polytechnic Institute, Worcester, USA, where is currently working toward the Ph.D. degree in systems engineering with the Systems Engineering Program.

His research focuses on model-based systems engineering and safety analysis.