CMU-AM-20-010

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Risk Management and Information Assurance Decision Support

November 14, 2019

Hanan Hibshi

Dr. Travis Breaux

Institute for Software Research
Carnegie Mellon University

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. HQ0034-18-1-0014. 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.

Approved for public release; distribution is unlimited.

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

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact the Acquisition Research Program (ARP) via email, arp@nps.edu or at 831-656-3793.

Abstract

The DoD often requires a high degree of information assurance and risk management. The DoD IT acquisition process remains controlled by complex information assurance (IA) certification processes. In March 2014, the DoDD 8500.1 was reissued to require a multi-tier, risk management process as embodied in the CNSSP No. 22 and NIST Special Publication 800-39, which promotes alignment with NIST IA control sets to mitigate security risk. This strategy was in use as early as 2006 by some stakeholders, including the Department of Navy Chief Information Officer (DONCIO). Despite these improvements, those responsible for accreditation will continue to struggle with assessing security risk in dynamically reconfigurable systems that change at runtime. The combination of changing users, changing applications, and changing locations is characteristic of modern IT and, consequently, requires a modern solution.

Like any organization, the DoD relies on security analysts who can assure that security requirements are satisfied. Relying on one expert’s opinion can be risky, because the degree of uncertainty involved in a single person’s decision could increase with time, memory failure or inexperience. In this technical report, we show to automate scenario generation where less experienced IT personnel can create scenarios that correspond to their own system architecture using our tool. The automation allows to crowdsource security assessments from experts. The tool will collect and analyze the expert ratings and return the results to the original requestor. In this paper, we propose our designed prototype for the tool, and we share the results of evaluating the prototype on 30 students who are completing a master’s degree in cybersecurity at a US institution. Based on the qualitative and usability analysis of responses, our proposed method is shown effective in systematic scenario elicitation. Participants had a 100% task completion rate with 57% of participants achieving complete task-success, and the remaining 43% of participants achieving partial task-success. Finally, we discuss our findings and future directions for this research in systematic scenario elicitation.

This research will yield important public benefits to private sector companies who supply and consume the dual-purpose information technology (IT) used by the DoD and who are frequently subject to security threats from organized crime, foreign governments and stateless hackers.

About the Authors

Hanan Hibshi - is a Research and Teaching Scientist at the Information Networking Institute at Carnegie Mellon University. Dr. Hibshi's research area includes usable security, security requirements and expert's decision-making. Dr. Hibshi's research involves using grounded theory and mixed-methods user experiments to extract rules for use in intelligent systems. Dr. Hibshi received a PhD in Societal Computing from Carnegie Mellon University, an MS in Information Security Technology and Management from the Information Networking Institute at Carnegie Mellon University, and a BS in Computer Science from King Abdul-Aziz University in Jeddah, Saudi Arabia. [hhibshi@cmu.edu]

Travis D. Breaux - is an Associate Professor of Computer Science, appointed in the Institute for Software Research of the School of Computer Science at Carnegie Mellon University. Dr. Breaux's research program searches for new methods and tools for developing correct software specifications and ensuring that software systems conform to those specifications in a transparent, reliable and trustworthy manner. This includes demonstrating compliance with U.S. and international accessibility, privacy and security laws, policies and standards. Dr. Breaux is the Director of the Requirements Engineering Laboratory at Carnegie Mellon University. Dr. Breaux has several publications in ACM and IEEE-sponsored journals and conference proceedings. Dr. Breaux is a member of the ACM SIGSOFT, IEEE Computer Society and USACM Public Policy Committee. [breaux@cs.cmu.edu].

THIS PAGE LEFT INTENTIONALLY BLANK

CMU-AM-20-010

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Risk Management and Information Assurance Decision Support

November 14, 2019

Hanan Hibshi

Dr. Travis Breaux

Institute for Software Research
Carnegie Mellon University

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. HQ0034-18-1-0014. 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

THIS PAGE LEFT INTENTIONALLY BLANK

Introduction

Cybersecurity decision-making relies on multiple composed security requirements. Systems are increasingly becoming more complex with different technologies, configurations, and the introduction of modern means like IoT.

Checklists are convenient because they generally apply to systems, however, they lack the context needed to assess the threat against a specific configuration (Haley, Laney, Moffett, & Nuseibeh, 2008). Claims that negative events are unlikely is difficult without being explicit about one's trust assumptions (Haley et al., 2008). Moreover, mapping the checklist to threat scenarios or other requirements is laborious process repeated by an analyst for each system. Finally, security requirements are not independent; instead, they work together in composition with different priorities and inter-dependencies to improve overall security (Garfinkel, 2005, p. 05).

Recently, we examined the effect of context and requirements composition on security requirements expert ratings (Hibshi, Breaux, & Broomell, 2015; Hibshi & Breaux, 2017). In that work, we used factorial vignettes in which requirements and system constraints are variables in a scenario description. We use scenarios from four technical areas: networking, operating systems, databases and web applications (Hibshi et al., 2015; Hibshi & Breaux, 2017). The result is a new method that we call the Multi-factor Quality Measurement (MQM) method that relies on using scenarios expressed in natural language text. The next is to introduce automation that involves using a tool where less experienced IT personnel can create scenarios that correspond to their own system architecture. The IT personnel could crowdsource security assessments from experts, and the tool would then analyze the collected data and send the results back to the IT personnel.

In this technical report, we prototype the tool for scenario elicitation from IT personnel. Since eliciting scenarios in natural language text format can be an ad hoc process with possible ambiguity, we build our tool prototype using a scenario language based on a simplified process model of iterative scenario refinement. The model consists of three steps: 1) eliciting an interaction statement that describes a critical action performed by a user or system process; 2) eliciting one or more descriptive statements about a technology that enables the interaction; and 3) refinement of the technology into technical variants that correspond to design alternatives. In the upcoming sections of the report we will provide more details about the prototyped model and the results of its evaluation.

Background and Related Work

Our approach for the prototype relies on providing context to security requirements through the use of scenarios and user stories (Cohn, 2004). A user story is defined as: “short, simple descriptions of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system (Cohn, 2004)”. The three-step model that we use in this paper uses language templates that are similar to user stories defined by Cohn. This research intersects the fields of cyber security, software engineering, requirements engineering, Human-Computer Interaction, and social sciences.

In this section of the report, we will review work from requirements engineering that investigates scenario-based methods.

SCRAM (A. G. Sutcliffe & Ryan, 1998) is a scenario requirements analysis method that aims at eliciting and validating user requirements. The method uses scenarios of user actions situated in scripts written by the requirements analyst, and presented to participants of a case study for verification (A. G. Sutcliffe & Ryan, 1998). This is different than our proposed approach as our scenarios originate from the stakeholders themselves. In addition, the SCRAM method needed an in-person training session with the use of a lecture, which is more time and effort consuming when compared to our model that uses online training text.

CREWS-SAVRE is a systematic scenario generation prototype that was introduced by Maiden et al. (Maiden, Minocha, Manning, & Ryan, 1998). Their proposed approach relies on allowing stakeholders to express scenarios using use cases. However, the scenarios used lack a theme or a storyline that helps stakeholders place requirements into a real-world context. In our approach, we use scenarios expressed in natural language text, and participants are asked to identify the scenario domain of interest and provide interaction statements that help stakeholders systematically add context to the scenario.

Scenario *samplin*g is a challenge that face scenario-based techniques in requirements engineering (Alistair Sutcliffe, 1998). To address the challenge, we need to generate and collect a number of scenarios that can be appropriately enough to represent a problem (Alistair Sutcliffe, 1998). Some generation techniques were introduced that use a single scenario as a single seed to generate multiple variations (A. G. Sutcliffe & Ryan, 1998; AG Sutcliffe, Shin, & Gregoriades, 2002; Alistair Sutcliffe, 2002), but these techniques relied on using scenario representations that are closer to a model (Alistair Sutcliffe, 1998). Our proposed model relies on eliciting scenarios from stakeholders using natural language text. We project that future steps of our method include automating the elicitation using crowdsourcing tools, which would increase the number of scenarios that can represent a problem and help address the scenario sampling challenge.

Makino and Oshnishi proposed a method for scenario generation that uses scenarios written in natural language text and presented to stakeholders' viewpoints (Makino & Ohnishi, 2008). To our knowledge, the proposed theoretical method and its prototype was not evaluated by stakeholders, however.

In this technical report, we understand that domain knowledge affects natural language scenarios (Kamsties & Peach, 2000) and we take a different approach by proposing a language model that elicit scenarios from the stakeholders who possess the domain knowledge of cybersecurity, which is our application of interest.

Systematic Scenario Elicitation

We now describe our approach to study the activity of systematic scenario elicitation. The approach assumes a model of structured scenario elicitation that results in a user story (Cohn, 2004) in natural language text that we refer to as scenario throughout this paper. To describe the model, consider the example text scenario shown in Figure 1. The example starts with an *interaction statement*, which is a statement that describes a critical action performed by a user or a system process. The *interaction statement* used in the example is specific to a domain (healthcare) but can also be stated more generically with no domain. Next, appears the *descriptive statement*, which describes a technology that enables the interaction.

For any type of technology, based on the stakeholder's needs and environment, there could be a variety of design alternatives to identify. To accommodate this diversity, the model allows a stakeholder to define a *variable* for a technology and list the design alternatives as different *levels* of that variable. In the example shown in Fig. 1, we define a $Network variable with three possible levels.

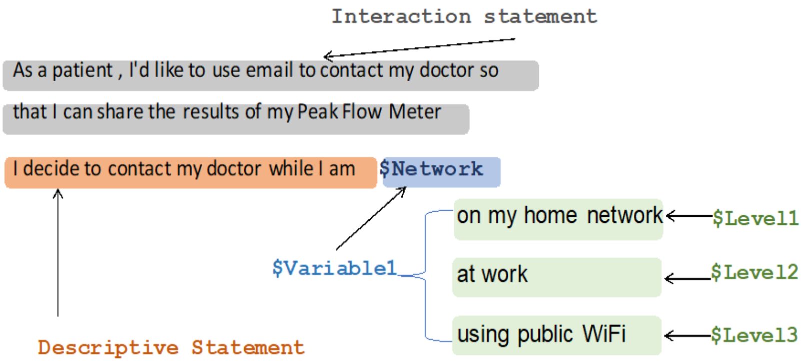

Interaction statement
As a patient , I'd like to use email to contact my doctor so
that I can share the results of my Peak Flow Meter
I decide to contact my doctor while I am
$Network
on my home network
$Level1
$Variable1
at work
$Level2
using public WiFi
$Level3
Descriptive Statement

Figure 1. Example of a text scenario

By looking at the example in Figure 1, we can see how that although the domain is defined as healthcare, but the different variable values have different impacts on cybersecurity. It is intentional in our approach that the scenario used in this example reflects real-world scenarios in cybersecurity. In other words, cyber security affects different domains and examples in addition to healthcare include, but not limited to, education, finance, and IoT.

The model is intentionally limited to these three elements: *interaction statement*, one or more *descriptive statements* that each contains a variable with *levels*. This limitation is necessary to identify and isolate sources of error in scenario generation. In the future, one could imagine studying more advanced scenarios with nested levels of interaction and description.

Stakeholders Input

To elicit scenarios from stakeholders, our approach involves three steps corresponding to the model elements described above:

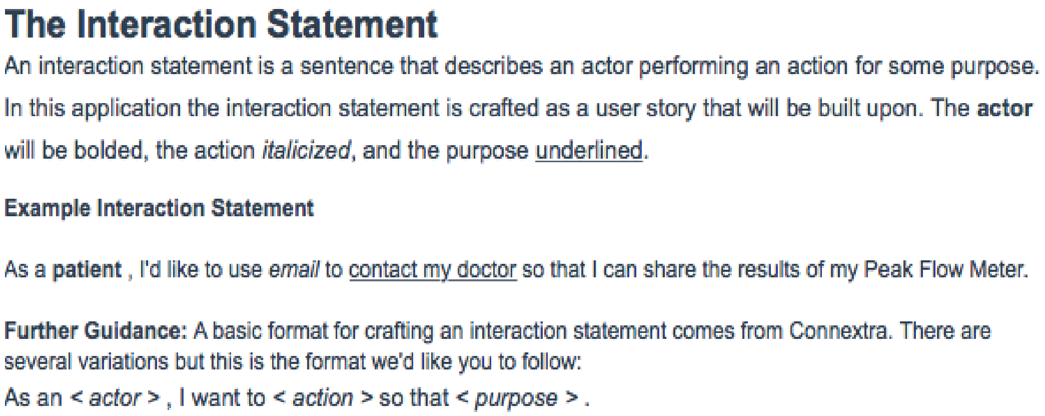
1) *Interaction statement elicitation*: where stakeholders are asked to provide a domain of interest and a related interaction statement in the following format:

   As an **< actor >**, I want to **< action >** so that **< purpose >**.

2) *Descriptive statement(s) elicitation*: where stakeholders are asked to provide one or more descriptive statements.

3) *Technology refinement:* where stakeholders define variables to represent the chosen technology and define a number of levels representing different design alternatives. After defining their own variables, stakeholders are asked to rank these variables based on a certain quality (e.g. security).

Scenario collection from users is completed online through online forms that prototypes the forms used in the design of the tool. The scenario elicitation process is accompanied with explanatory text and training material. For example, we use the text shown in Figure 2 below to explain interaction statements to stakeholders. We follow a similar approach to explain the descriptive statements, the variables, and the levels.



The Interaction Statement

An interaction statement is a sentence that describes an actor performing an action for some purpose. In this application the interaction statement is crafted as a user story that will be built upon. The **actor** will be bolded, the action *italicized*, and the purpose underlined.

Example Interaction Statement

As a **patient** , I'd like to use *email* to contact my doctor so that I can share the results of my Peak Flow Meter.

Further Guidance: A basic format for crafting an interaction statement comes from Connextra. There are several variations but this is the format we'd like you to follow:
As an < *actor* > , I want to < *action* > so that < *purpose* > .

Figure 2. Training and example text for the interaction statement of a text scenario

THIS PAGE LEFT INTENTIONALLY BLANK

Evaluation of the Model

We designed a prototype and test the model on stakeholders in the form of an online survey. The survey consists of several forms that corresponds to the forms used in the prototype. Our target population is stakeholders interested in the cybersecurity domain. At the beginning of the survey, we explain to participants that the end goal of these tasks is to construct a *vignette,* which we define to participants of the survey as: *a story that people read before making an important decision. The vignette adds context to help the person make a more informed decision*.

Going through each step in the model, we provide stakeholders with definitions and running examples to help understand the concepts needed to perform the task related to that step (see Fig. 1, and Fig. 2). The study participants are asked to provide their input following each explanation and training. For example, following the training shown in Figure 2 above, participants are asked to provide an interaction statement for their domain of interest (they have been presented with training materials and example domains prior to being introduced to the interaction statement).

Upon task completion, we ask participants to rate their own experience performing the tasks in the user study. We ask them to rate the difficulty of each individual task on a 7-point scale. In addition, we ask participants about the likelihood (using a 7-point scale) of using a tool for scenario creation that is similar in design to the exercise that they just completed. We repeat this likelihood-of-use question twice: for someone inside the participant's organization, and for someone outside the participant's organization. This repetition encourages participants to think more broadly about the possible broader benefits of the tool prototype that they just have tried even if they do not see a direct benefit to themselves in using such tool. We also allowed participants to provide additional open-ended comments.

Lastly, we ask participants to answer 14 security knowledge questions and standard demographic questions (e.g. gender, age, and years of experience).

We piloted the study by recruiting participants who attended the Black Hat 2017 conference in Las Vegas, USA and providing a $25 Amazon gift card to each participant. The Black Hat conference is an annual security conference that is recognized worldwide as a highly professional event series providing cutting-edge technical information and training in the computer and network security (“Black Hat,” n.d.). Due to the fast pace of the conference, noise, and multiple other distractions, participants did not show signs of providing focused responses to our training material neither the survey questions. Hence, a lesson learned was to recruit participants who would finish the survey at their convenience and then return later to collect their compensation.

Following pilot, we recruited participants from who are enrolled in a well-recognized information security master’s degree program in a top university in the USA. Each participant was compensated with a $25 Amazon gift card.

Analysis of Participant Responses

We are interested in the effectiveness, efficiency, and user-satisfaction of the proposed three-step scenario elicitation model. We will describe below how we analyze and measure these components.

Effectiveness is concerned with a stakeholder success in completing a task while maintaining an acceptable level of accuracy (Frøkj\a er, Hertzum, & Hornb\a ek, 2000). In our results we measure effectiveness using task completion rates. To account for task accuracy, we differentiate between *full task success*, where participants complete the task with no missing information or errors; and *partial task success* where participants complete the task with some errors or missing information.

Efficiency is concerned with the resources a stakeholder consumes to complete a task while maintaining an acceptable level of accuracy (Frøkj\a er et al., 2000). In our study, we use task completion time to measure efficiency.

Satisfaction is concerned with stakeholders’ attitudes when using a system (Frøkj\a er et al., 2000). To measure participants satisfaction with our model, we use rating scales to ask study participants to provide their perception of task difficulty, and their projection of likelihood-of-use.

The constructs shown above rely on qualitative analysis of study participants responses. We use grounded analysis (Corbin & Strauss, 2007; Glaser, 1978) and coding theory (Saldaña, 2012) to code participants open-ended, text responses.

Below, we will explain how we analyzed the data to help measure the three constructs listed above and to provide qualitative insights.

- ***Domains****:* Participants were asked to list their domains of interest and the interaction statement. Using open coding, we review participant answers and categorize the elicited domains into a broader domain category. For example, the forensics domain is categorized into the broader domain of cybersecurity, and the banking domain is categorized into the broader domain of finance (finance can include corporate investment for example).
- ***Interaction Statement***: A full interaction statement should contain the actor, action and purpose. We coded interaction statements as *complete*, if the participant provides a full interaction statement, and *incomplete*, if participant provided an interaction statement that is missing the purpose. We coded empty responses with N/A, and non-statement responses (e.g. words and phrases) as not provided.
- ***Descriptive Statement***: A correct descriptive statement should follow the format shown in the example shown in Figure 1 and must contain a variable preceded by the ($) sign. We coded descriptive statements as *correct*, if the participant provides a descriptive statement using a format similar to the training, *partial*, if the participant provides partial text that still can be comprehensible as a descriptive statement but is missing the variable or the dollar sign ($) preceding the variable, and *incorrect* if otherwise. We also coded the relationship between descriptive statements and interaction statements with one of the following codes: *related,* if a strong relationship can be derived from the text; *semi-related,* if the relationship can be derived but is not obvious; and *not related,* if otherwise.
- ***Variables*****:** Initially, we coded a variable *correct,* if it correctly represents a technology that can have multiple design alternatives (levels), and incorrect, otherwise. Later, we added the code: *level*, if the variable is not perceived as a broader category of its level, but rather is perceived as another level (e.g. the variable “home network” is coded as level, if the participant provides “employer network” and “public network” as levels). Variables that are missing the dollar sign ($) are coded as *partial*.
- ***Variable/level structure*****:** We coded the structure as *correct,* if the participant provided variables and levels in the expected format where variables are a broader technology category of the levels, and we coded the variable/level structure to be *incorrect,* if otherwise.

Training material used in the experiment includes an example of a $Network variable with three possible levels (see Fig. 1). The levels shown to participants are technical variants of different network configurations that vary in their security strength (some levels are more secure than others). For each variable/level combination, we assigned codes that best describe the relationship between the levels and the variable they are supposed to refine. In cases where the variable is missing or wrong, then we code the relationship between the levels themselves. The codes, or concept labels, follow the Glassier view of *open coding*, wherein the codes emerge from the data without any pre-defined initial code set (Glaser, 1978).

Inter-Rater Reliability

When coding qualitative data that is subject to different interpretations, it is a recommended to use multiple raters and calculate inter-rater reliability where researchers use statistical measures like Cohen's Kappa to measure above chance agreement (Cohen, 1968) and be able to judge the quality of the code set being used (Cohen, 1968; Saldaña, 2012). We use two coders for our data set (the first and second authors) and we calculate Cohen's Kappa for each coded data type separately. Our calculated Kappa averaged at 0.9, which is considered good agreement (Cohen, 1968). Next, the disagreements were resolved to reach complete agreement to finalize the dataset for analysis.

Results

We now present our analysis results. We collected scenarios from 30 participants. The mean time that a participant used to complete the scenario elicitation tasks including training is 24 minutes.

Demographics

All participants have a bachelor's degree in computer science or a related field and are currently enrolled in a graduate information security program at a top US university. Out of the 30 students, three participants already work for industry and one works for the US government. The mean score for participants on the security knowledge test is 58%. Table I below summarizes the demographics statistics of study participants.

TABLE I. Demographics Information

Description		**Participants**	
		Number	*Percentage*
Gender	Male	21	70%
	Female	8	27
	Prefer not to say	1	3%
Years of Computer Security Experience (Mean=2)	Less than 1	6	20%
	1 – 2 years	13	43 %
	3 – 4 years	7	23 %
	5 – 7 years	4	13%
Age range	18 -24	18	60%
	25-34	12	40%
Took job training in security		27	40%
Self-taught security knowledge		12	57%
Security Knowledge Score	Scored above 60%	12	31%
	Scored between 40% and 60%	16	41%
	Scored below 40%	2	5%

Task Completion

All 30 participants completed the user study from start to end, and they provided a domain of interest. The task completion rate that maps to our research questions is related to constructing a scenario using the three steps of providing an interaction,

We define three completion categories: full completion, when a participant completes the interaction statement, and at least one descriptive statement with its associated variables and levels with full accuracy; partial completion if a participant completes the interaction statement, and at least one descriptive statement with its associated variables and levels with partial accuracy, and failure if a participant did not provide an interaction statement and did not provide any description statements with an associated variable. Since our evaluation of responses relies on qualitative analysis, we show in Table II how we classify full accuracy vs. partial accuracy based on the codes used in the grounded analysis.

Based on our definitions above, our study data shows that 57% of participants achieve full completion (17 responses), 43% achieve partial completion (13 responses), and 0% failures.

When analyzing the 13 partial completions, we found four participants providing incomplete interaction statements that did not include a purpose, five participants did not precede the variables with a dollar sign ($), three participants used another level instead of a broader category for levels, and one participant who provided a variable with levels that do not relate or show a clear variable/level structure.

TABLE II. Tasks Accuracy Definitions Based on Codes

Coded Task	Codes		
	Full accuracy	Partial accuracy	Failure
Interaction statement	complete	incomplete	Not provided, N/A
Descriptive statement	correct	partial	incorrect
Variable	correct	partial, level	incorrect

Participant Satisfaction

We measure participants interaction using participants ratings of task difficulty and likelihood of use. All 30 participants provided ratings for task difficulty and likelihood of use, and only eight participants provided additional open-ended comments.

Task difficulty. Table III summarizes the participant feedback about the task difficulty involved in scenario creation. For the first four tasks: understanding vignettes (i.e. scenarios), understanding interaction statements, crafting interaction statements, and understanding descriptive text; almost half (between 48-63%) of participants were skewed toward easy ratings (somewhat easy, easy, and very easy combined). For the later four tasks shown in Table III, participants feedback is less skewed in any direction. By assigning numeric values to the 7-point scale (with 1=Very Easy and 7= Very Hard), we found that the mean value for all task's ranges between 3.1 and 3.9, which is slightly below Neutral (Neutral=4), leaning towards the easy category.

TABLE III. Participants Feedback about Task Difficulty

Task	Very Easy	Easy	Somewhat Easy	Neutral	Somewhat Hard	Hard	Very Hard
Understanding vignettes	13%	33%	7%	27%	17%	3%	0%
Understanding interaction statements	7%	27%	30%	20%	10%	3%	3%
Crafting interaction statements	3%	14%	31%	24%	21%	0%	7%
Understanding descriptive text	3%	20%	33%	20%	17%	3%	3%
Crafting descriptive text	0%	13%	30%	13%	40%	3%	0%
Understanding variables	7%	17%	17%	30%	23%	3%	3%
Crafting variables	7%	7%	23%	23%	30%	7%	3%
Understanding levels	10%	13%	13%	27%	17%	10%	10%
Crafting levels	7%	13%	20%	27%	17%	7%	10%

Likelihood-of-Use. Table IV summarizes participant feedback about the likelihood of using a tool similar to what was presented in the study by the participants themselves or someone else inside or outside their organization. In general, participants were slightly more skewed towards unlikely. Three participants explained in their open-ended comments that they did not fully understand the end goal of the tool presented in the survey. By looking at their performance, these three participants still managed to complete the required tasks. These observations suggest that participants might not been able to project the benefit of using the language proposed in the tool, which affected their projection of likelihood-of-use.

TABLE IV. Participants Feedback about Likelihood of Using a Vignette Generation Tool

If this tutorial was integrated into an online tool for crafting vignettes that can be used later for running user study, how likely	**Very Unlikely**	**Unlikely**	**Somewhat Unlikely**	**Neutral**	**Somewhat Likely**	**Likely**	**Very Likely**
would YOU use such a tool	10%	13%	23%	17%	13%	23%	1%
would someone IN your organization use such a tool	10%	3%	7%	33%	27%	17%	3%
would someone OUTSIDE your organization use such a tool	17%	10%	13%	17%	30%	7%	7%

Domain Selection and Interaction Statement

Participants were asked to specify a domain of interest for the scenario creation. Recall from above that we coded the domains into broader domain categories. Table V lists the domain categories and the frequency, or number of participants who provided a domain within that category.

It was expected to see the dominance of the cybersecurity domain because of the participants security background, and the cybersecurity examples used in the training. By investigating the 13 responses related to domains other than cybersecurity, we found that except for one response, all responses had a relationship to the cybersecurity domains that was prevalent in the descriptive statements, the variables, and/or the levels.

TABLE V. Categories of Domains Selected by Participants

Category	Frequency
cybersecurity	17
healthcare	5
finance	6
education	1
social media	1

Out of 30 responses, only one participant did not provide an interaction statement. For the remaining 29 participants, 24 participants (80%) provided complete interaction statement that conforms to the structure shown in training; and five participants (17%) provided incomplete interaction statements four of which is missing the purpose part of the statement (see Figure 1 for interaction statement example).

Descriptive Statements

We expect participants to provide a descriptive statement for each variable to help explain the role of the technology represented by the variable in a way that relates back to the interaction statement. Recall from above, participants were asked to provide four descriptive statements along with their related variables and levels. Table VI summarizes the participant performance with regards to providing descriptive statements. An important observation from Table VI, is how participants tend to provide fewer statements as they proceed through the survey. This observation could be related to the fatigue effect.

TABLE VI. Descriptive Statements Performance

Evaluation	Descriptive Statement Performance			
	1st	*2nd*	*3rd*	*4th*
Descriptive Statement Validity				
Correct	2	15	16	14
Partially correct	10	7	4	2
Incorrect	0	1	3	2
Not provided	0	7	7	12
Descriptive Statement Relationship to Interaction Statement				
Related	28	19	19	13
Semi-related	2	3	3	4
Not-related	0	1	1	1
*N/A	0	7	7	12

*Either the interaction statement or the descriptive statement is not provided

Variables and Levels

Providing variables and their levels in the correct format is an important part of scenario elicitation. In our analysis, in addition to evaluating the correctness of the variable formats and the structure of the variable/level combination see we investigate the relationship between each technology variable and its levels that emerged as a result of our open coding. Below, we list the discovered relationship codes along with their definitions:

- *Technical variants*: if the levels are technical variants that correspond to design alternatives. Each level will have a different impact on system's quality. For example, the levels: public Wi-Fi, home Wi-Fi, and employer Wi-Fi are technical variants for a $Network variable and each of these levels will impact security, differently.
- *Environmental variant*: The levels describe comparable environment factors that have varying effect on a quality. For example, "at normal business hours" and "outside business hours," are environmental variants that impact the security of a system.
- *Alternatives*: The levels are not technical variations of the variable but are considered alternative technologies to the variable. This is more likely to occur when a participant could not distinguish the difference between a variable and its levels. For example, the level "Home Wi-Fi" is an alternative to the variable $PublicWiFi.

- *Actors/Agents*: The levels describe actors or agents. For example: the levels “employee,” “contractor,” “vendor,” and “guest” represent different actors for the $user variable and they affect the security of the system differently.
- *Actions*: The levels are different actions performed by an actor for the variable. For example, “create,” “delete,” “modify,” and “transfer” are levels for a variable like: $adminActions.
- *Events*: The levels describe an event. For example, “sensor alert” and “error has occurred” are two possible levels for an $IDS (intrusion detection system) variable.
- *No relationship*: we use this code when a participant provides levels, but we could not infer a relationship among these levels based on our own security knowledge.
- *N/A*: we use this code, if a participant does not provide input for a variable or at least one level for the variable is empty or incoherent

Table VII summarizes all results of variables and levels analysis. Similar to what was observed for the descriptive statements, the number of inputs from participants decrease for variables and levels as participants proceed through subsequent steps of the survey.

TABLE VII. Variable/Level Combinations Performance

Evaluation	1st Var.	2nd Var.	3rd Var.	4th Var.
Variable Format Correctness				
Correct	16	14	15	13
Level	4	0	1	1
Partial	8	6	4	3
Incorrect	2	2	2	0
Not provided	0	8	8	13
Variable/Level Structure Correctness				
Correct	22	17	17	15
Incorrect	8	5	5	2
*N/A	0	8	8	13
Levels relationship to Variable				
Technical Variants	22	15	10	9
Environmental variants	1	2	4	3
Alternatives	4	2	4	2
Actors/agents	1	0	0	0
Actions	1	0	3	1
Events	0	1	0	1
No relationship	0	2	1	1
*N/A	0	8	8	13

*Either the variables or is not provided

Interestingly, participants seem to understand the concept of varying levels that impact security, even when they fail to provide the correct corresponding variable. We observed in our dataset that a participant could write an incorrect variable format, repeat some text from the descriptive text, or create a variable that is actually another level for the levels that they provided, but the same participant appears to understand how to provide a number of varying levels that share a meaningful relationship.

Threats to Validity

In this section we will report possible threats to validity and our approach to address these threats.

Construct validity is whether measures used in the study actually measure the construct of interest (Yin, 2009). In this study, the constructs of interest either: based on codes assigned by researchers to open-ended text responses or based on participants ratings. One threat to construct validity is the definitions of the codes in the coding frame could be ambiguous, such that the codes are inaccurately applied to the wrong statements. To address this threat, we had two researchers (the first and second authors) identify the points of disagreement and reconciled differences in a subsequent meeting. Recall from above, we computed the inter-rater reliability statistic Cohen's Kappa that showed good agreement. Another threat to construct validity is the participants subjective ratings of difficulty and likelihood-of-use. To reduce the effect of this threat, we designed the questions using statements in a neutral tone that are less likely to bias participants in a certain direction. We present the question as follows:

Please describe the difficulty of the following tasks in the survey:
Understanding interaction statements (7-point scale displayed)

This approached is less biased than asking participants to rate their agreement level on a scale with a statement like: *I find interaction statements easy to understand.* This latter design could lead to more biased responses.

Internal validity refers to whether the conclusions drawn from the data are valid (Yin, 2009). The completeness of the data threatens internal validity, because participants may experience fatigue that result in empty responses. Hence, we design the survey to be completed in 25 minutes on average to avoid possible fatigue. We also coded empty response with N/A during the analysis to assess the quality of the dataset.

External validity refers to the extent to which the results of this study can be generalized to other situations (Yin, 2009). *Reliability* refers to the extent to which the study procedures can be repeated and achieve the same results (Yin, 2009). Two researchers have validated the coding frame, which increases its reliability and generalizability, and which can thus be reused by other investigators. In addition, details about the study procedures are documented for future reuse. However, this study is based on grounded analysis, which limits generalizations to only this data set. While some might argue that our findings are thus too limited, qualitative research contributes to theory generation that supports follow-on studies and controlled experiments.

Discussion, Future Work and Conclusions

In this research we introduced a language for scenario elicitation that is based on a three-step model that elicit structured parts of natural language text from stakeholders. When the natural language text parts are combined, the end result is short scenario template with a variable that can take different values of varying levels of technologies. The varying technologies allows to compare different technology alternatives that can be further evaluated by other analysts, stakeholders, or domain experts. We present results from our evaluation of a user study where we examine the usability of our introduced method. Our analysis results for this preliminary study suggest a promising future in this area, because we had no empty responses or failures. The task completion is 100% divided between 57% full accuracy, and 43% partial accuracy.

Unlike previous research in requirements engineering where scenarios were produced from formal representations that more closely correspond to models, our method relies on guiding stakeholders to create scenarios presented in natural language text. Using a structured approach in collecting statements, have shown a benefit in collecting scenarios that share similar syntax and differ in semantics. This uniformity has a number of benefits that we list below:

Scalability and more systemized collection process where a requirement engineer can tailor our method based on the domain of interest and use it to collect natural language scenarios from a larger participant pool. Systemizing natural language scenario elicitation offers more scalability and coverage compared to collecting unstructured stakeholder narratives.

Homogenous stakeholder scenarios that results from using a structured approach in our method. Scenarios written in natural language is known to be more user-friendly to the stakeholder, but without proper structure, the process becomes ad-hoc and scenarios will be highly heterogenous with no unifying pattern that can help an analyst parse different scenario. In our results, all elicited scenarios shared a common structure, even in cases where participants had partial accuracy.

Systemized scenario analysis which is a result of the homogeneity feature of scenarios collected using our proposed method. Following a uniformed syntax is a feature that facilitates the parsing of natural language text, which allows requirements engineers to analyze and validate scenarios using systemized means and automated tools. In our experiment, we were able to systematically analyze the data and we found the process to be less time consuming than analysis done on unstructured natural language text collected, for example, in user interviews and focus groups.

Real capture of stakeholder experiences and domain knowledge because our method allows stake holders to write scenarios using natural language text where they only learn a certain structure to arrange their words. In our experiment results, the security domain knowledge was evident in the elicited scenarios.

Embedding cybersecurity within the domain of interest by using scenarios and providing context, we were able to present cybersecurity problems to stakeholders in the context of the domain area of the application. This is important because the cybersecurity challenges exist within many critical domains such as, healthcare, education, defense, and finance. This relationship is important because it is one way of communicating to the stakeholders the importance and the impact of cybersecurity.

Going forward, our future research involves introducing more automation to the tool. We envision that using our tool, an analyst would be able to build their own scenario and then send out invitations for experts to rate the overall security, the individual security requirements, and provide further requirements that can enhance the ratings. Such a tool would have a great impact the DoD and other organizations in the public and private sectors, because it will help systemize the evaluation of security components using real-experts input.

Another future research direction is to investigate the reasoning in the backend of the tool. Once the analyst chooses a scenario of interest, the tool would be able to show the analyst a security assessment of the created scenario using data previously collected from experts in user studies. The challenge in this research direction, is build the dataset of expert ratings.

THIS PAGE LEFT INTENTIONALLY BLANK

References

Black Hat. (n.d.). Retrieved February 20, 2018, from https://www.blackhat.com/about.html

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213.

Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Corbin, J., & Strauss, A. (2007). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage.

Frøkj\a er, E., Hertzum, M., & Hornb\a ek, K. (2000). Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 345–352. ACM.

Garfinkel, S. (2005). *Design principles and patterns for computer systems that are simultaneously secure and usable* (Massachusetts Institute of Technology). Retrieved from http://dspace.mit.edu/handle/1721.1/33204

Glaser, B. G. (1978). *Theoretical sensitivity: Advances in the methodology of grounded theory*. Sociology Pr.

Haley, C. B., Laney, R., Moffett, J. D., & Nuseibeh, B. (2008). Security requirements engineering: A framework for representation and analysis. *Software Engineering, IEEE Transactions On*, *34*(1), 133–153.

Hibshi, H., Breaux, T., & Broomell, S. B. (2015). Assessment of Risk Perception in Security Requirements Composition. *2015 IEEE 23rd International Requirements Engineering Conference (RE)*, 146–155.

Hibshi, H., & Breaux, T. D. (2017). Reinforcing Security Requirements with Multifactor Quality Measurement. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 144–153. Lisbon, Portugal: IEEE.

Kamsties, E., & Peach, B. (2000). Taming ambiguity in natural language requirements. *Proceedings of the Thirteenth International Conference on Software and Systems Engineering and Applications*.

Maiden, N. A. M., Minocha, S., Manning, K., & Ryan, M. (1998). CREWS-SAVRE: systematic scenario generation and use. *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference On*, 148–155. https://doi.org/10.1109/ICRE.1998.667820

Makino, M., & Ohnishi, A. (2008). A Method of Scenario Generation with Differential Scenario. *2008 16th IEEE International Requirements Engineering Conference*, 337–338. https://doi.org/10.1109/RE.2008.17

Saldaña, J. (2012). *The coding manual for qualitative researchers*. Sage.

Sutcliffe, A. G., & Ryan, M. (1998). Experience with SCRAM, a scenario requirements analysis method. *Requirements Engineering, 1998. Proceedings. 1998 Third International Conference On*, 164–171. IEEE.

Sutcliffe, AG, Shin, J.-E., & Gregoriades, A. (2002). Tool support for scenario-based functional allocation. *Human Decision Making and Control*.

Sutcliffe, Alistair. (1998). Scenario-based requirements analysis. *Requirements Engineering*, *3*(1), 48–65.

Sutcliffe, Alistair. (2002). *User-centered requirements engineering*. Springer Science & Business Media.

Yin, R. K. (2009). *Case study research: Design and methods* (Vol. 5). Sage.

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net