2021-05-10

Contractual Safety of Model-Based Requirements: Preliminary Results of an Experimental Study

Shadab, Niloofar; Salado, Alejandro

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/68110

SYM-AM-21-046

EXCERPT FROM THE PROCEEDINGS
OF THE
EIGHTEENTH ANNUAL
ACQUISITION RESEARCH SYMPOSIUM

Contractual Safety of Model-Based Requirements: Preliminary Results of an Experimental Study

May 11–13, 2021

Published: May 10, 2021

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 any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Contractual Safety of Model-Based Requirements: Preliminary Results of an Experimental Study

Niloofar Shadab—is a PhD student in systems engineering in the Department of Industrial and Systems Engineering at Virginia Tech, Blacksburg, VA. Her research track is on using systems theory to align systems engineering practices for Engineering Intelligent Systems. Shadab received her MSc in model-based systems engineering (MBSE) from the University of Maryland, College Park, MD, in 2016; she focused on Unmanned Aerial Vehicles' safety in civil applications. She has used MBSE languages such as SysML and UML. She worked as a Systems Analyst for AvMet Applications, a consulting company in the aviation industry. She received her BSc in aerospace engineering at the Sharif University of Technology, specializing in flight dynamics and controls. [nshadab@vt.edu]

Alejandro Salado—is an assistant professor in the Grado Department of Industrial and Systems Engineering and the director of its systems engineering program at Virginia Tech. He conducts research in problem formulation, design of verification and validation strategies, model-based systems engineering, and engineering education. Salado holds a BS/MS in electrical and computer engineering from the Polytechnic University of Valencia, an MS in project management and an MS in electronics engineering from the Polytechnic University of Catalonia, the SpaceTech MEng in space systems engineering from the Technical University of Delft, and a PhD in systems engineering from the Stevens Institute of Technology. [asalado@vt.edu]

Abstract

Requirements form the backbone of contracting in acquisition programs. Requirements define the problem boundaries within which contractors try to find acceptable solutions (design systems). At the same time, requirements are the criteria by which a customer measures the extent that their contract has been fulfilled by the supplier. Therefore, requirements are instrumental in the success of acquisition programs. In this context, the quality of a requirement set is determined by the level of contractual safety that it yields. From a technical perspective, contractual safety is driven by the accuracy, precision, and level of completeness of the requirement set. Unfortunately, textual requirements do not provide acceptable levels of contractual safety, as they remain a major source of problems in acquisition programs. This is partly caused by the inherent limitations of natural language to statically capture written statements with precision and accuracy. In addition, natural language is difficult (often impossible) to parse into consistent logical or mathematical statements, which limits the use of systematic and/or automated tools to explore completeness. Model-based requirements have been proposed as an alternative to textual requirements, with the promise of enabling higher accuracy, precision, and completeness when eliciting requirements. However, this promise has not been demonstrated yet. Therefore, research is needed to understand the contractual impacts of using model-based requirements instead of textual requirements before model-based requirements can be widely adopted to support acquisition programs. This paper presents preliminary results of a research project that measures the contractual safety yielded by model-based requirements. Specifically, the research addresses the main question of whether using model-based requirements improves the contractual safety of acquisition programs compared to using textual requirements. The accuracy, precision, and completeness achieved by model-based requirements are empirically measured using an experimental study. We employ a notional airborne solution to a surveillance and detection problem.

Introduction

Requirements form the backbone of contracting in acquisition programs. Requirements define the problem boundaries within which contractors try to find acceptable solutions (design systems; Salado et al., 2017). At the same time, requirements are the criteria by which a customer measures the extent that their contract has been fulfilled by the supplier (INCOSE, 2015). Hence, it is not surprising that some authors consider requirements the cornerstone of systems engineering (Buede & Miller, 2016).

Within an acquisition context, the quality of a requirement set is determined by the level of contractual safety that it yields. In the experience of the second author in acquisition programs (leaving contractual mechanisms aside and focusing only on the technical side of acquisition) contractual safety is driven by the accuracy, precision, and level of completeness of the requirement set:

- Achieving accuracy is necessary to guarantee that the requirements capture the real needs of the customer (Salado & Nilchiani, 2017c).
- Achieving precision is necessary to guarantee that the supplier interprets the requirements exactly as the customer intended when writing them (Salado & Wach, 2019b).
- Achieving completeness is necessary to avoid gaps in the problem formulation (Salado et al., 2017). If requirements are missing, a supplier may reach contractually acceptable solutions that do not fulfill the needs of the customer.

Unfortunately, textual requirements do not provide acceptable levels of contractual safety, as they remain a major source of problems in acquisition programs (GAO, 2016; Gilmore, 2011). This is partly caused by the inherent limitations of natural language to statically capture written statements with precision and accuracy (Pennock & Wade, 2015). In addition, natural language is difficult (often impossible) to parse into consistent logical or mathematical statements (Fockel & Holtmann, 2014; Gervasi & Zowghi, 2005; Tjong et al., 2006), which limits the use of systematic and/or automated tools to explore completeness (Carson et al., 2004; Salado & Nilchiani, 2017b; Salado et al., 2017).

Model-based requirements have been proposed as an alternative to textual requirements, with the promise of enabling higher accuracy, precision, and completeness when eliciting requirements (Salado & Wach, 2019b). Hence, we suggest that model-based requirements will improve contractual safety in acquisition programs. However, this statement remains to be proven. Although prior work has provided some indication in this direction (Salado & Wach, 2019b; Wach & Salado, 2021a, 2021b), we currently do not completely understand how engineers will interact with and interpret model-based requirements.

In this paper, we present preliminary results of an experimental study that evaluates the contractual safety of model-based requirements by studying their precision, accuracy, and potential for completeness compared to textual requirements. Particularly, we show some initial results that measure the impact of both types of approaches in the elicitation of requirements that are properly bounded and free of unnecessary constraints.

Literature Review

Model-based Requirements

Most of the literature in model-based requirements deals with aspects related to requirements management (e.g., requirements traceability and allocation (Badreddin et al., 2014; Borgne et al., 2016; Holder et al., 2017; Holt et al., 2012; Marschall & Schoemnakers, 2003; Mordecai & Dori, 2017; Ribeiro, 2018; Schmitz et al., 2010) or requirements engineering and management processes (Holt et al., 2012; Holt et al., 2015; Holt et al., 2012; Holt et al., 2015). Modeling the actual requirements is generally accomplished with one of two approaches. The first approach defines a specific type of model object that encapsulates the requirement, which is formulated using a textual statement. For example, SysML uses elements called requirement element and requirement diagram (Friedenthal et al., 2015). Given the inherent vagueness of natural language to formulate requirements (Salado & Wach, 2019b), such modeling approach provides minimal improvement with respect to working with textual documents (from the perspective of enabling computational assessment of requirement completeness). While parsing textual requirement statements into a set of properties or constraints associated to objects has been shown to be feasible in software systems (Lu et al., 2008), since parsing protocols rely on the structure of natural language, and not on meaning, the resulting requirements model inherits the vagueness of natural language.

In the second approach, system models are directly flagged as requirements. They often use behavioral models and/or state machines to capture functional requirements (Aceituna et al., 2011; Aceituna et al., 2014; Adedjouma et al., 2011; Soares & Vrancken, 2008; Ouchani & Lenzinia, 2014; Pandian et al., 2017; Siegl, 2010), and non-functional requirements are captured as properties or attributes of the system (Reza, 2017; Saadatmand, 2012). For example, in SysML, this is achieved by defining values for the physical block that represents the system for which requirements are being formulated (Fockel & Holtmann, 2014; Holt et al., 2015). However, this second approach presents two weaknesses, which are discussed at length in Salado and Wach (2019b). First, the separation between functional and non-functional requirements is ambiguous (Salado & Nilchiani, 2014), since, from a systems-theoretic standpoint, such a distinction does not really exist (Salado & Wach, 2019b). Requirements modeled in such a way may therefore inaccurately capture the requirement of concern. Second, since directly using behavioral models of the system of interest imposes a solution as the requirement (INCOSE, 2012), such requirements may also unnecessarily constrain the solution space (Salado et al., 2017).

To overcome these problems, formal requirement models that prescribe a requirement structure without relying on pre-existing textual statement have been proposed (Borgne et al., 2016; Micouin, 2008). For example, Micouin defines a requirement as a combination of a condition (e.g., when flying), a carrier (e.g., the system), a property (e.g., power consumption), and a domain (e.g., less than 100 W) (Micouin, 2008). While internally consistent, these structures do not prescribe the type of property that may be defined. As a result, they allow for imposing a system solution by defining design-dependent requirements, which are considered a poor practice in requirements engineering (INCOSE, 2012) because they unnecessarily constrain the solution space (Salado et al., 2017).

Alternatively, requirements have also been modeled as exchanges in which the system of interest participates. Three approaches are predominant: model requirements as data exchanges (Teufl, 2013), as exchanges between actors (Miotto, 2014), and as input/output transformations through physical interfaces at the system boundary (Salado & Wach, 2019b). The first approach is insufficient to model space system requirements because it is only capable of modeling data exchanges, not physical aspects of the problem space. While the second approach (modeling requirements as exchanges between actors) may be promising to model stakeholder needs, it is not adequate for system requirements. This is because the requirement remains unbounded (dependent on the actions of external systems), which is considered a poor practice in requirements engineering (INCOSE, 2012). The third approach, which is the basis of True Model-Based Requirements (TMBR), is consistent with the principles of systems theory and the guidelines for writing good requirements (Salado & Wach, 2019b). It was used as the basis to develop the TMBR approach used in this paper and described in the next section.

The True Model-Based Requirements Approach

In this paper, TMBR is implemented as an extension of behavioral and structural model elements of SysML (Friedenthal et al., 2015). The usage of the different model elements relies on semantics that differ from those corresponding to the original model elements in SysML. Specifically, the models presented in this paper capture solution spaces (sets of solutions), not systems (single solutions). While SysML models are used for diagrammatic purposes, their *meaning* differs from the traditional SysML specification. In particular, TMBR's implementation in SysML is architected as follows:

1. An extended sequence diagram captures the required logical transformation required to the system.
2. Signals capture required logical inputs and outputs with their required attributes.
3. Ports in block elements capture the required physical interfaces and their required properties through which inputs and outputs are conveyed.
4. An extended state machine diagram is used to capture mode requirements, which capture the simultaneity aspects of requirements applicability.

A visual representation of the basic construct of a requirement modeled as an input/output transformation is shown in Figure 1. *Blocks* are used to represent the system of interest for which requirements are being defined and *sequence diagrams* are used to capture the required flow of inputs (and outputs) to (and from) the system. In this way, the system remains a *solid line*, preventing the modeler from defining design-dependent or inner aspect of the system; only the system's behavior at its boundary in the form of external inputs and outputs is allowed. In (and out)-flows to (and from) the system are defined as items (i.e., energy, information, or material) not as actions, hence guaranteeing consistency with systems theoretic principles for system requirements.

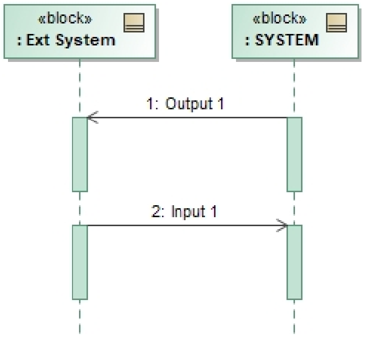

Figure 1. Input/Output Transformation Sequence Between a System and an External System

The *signal* element is used to capture the required input and output characteristics. Attributes of the signal are used to capture the required characteristics of the inputs and outputs. Examples of an input and an output are shown in Figure 2. The required interfaces through which the required inputs and outputs must be exchanged are captured using *ports*. An example of this is shown in Figure 3. The required properties of the interfaces are captured using *InterfaceBlocks*. Properties and values are used to capture requirements on the physical and transport (data) layers of the interface. An example of a modeled interface is shown in Figure 4.

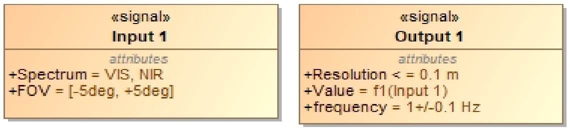

Figure 2. Example of Input/Output Signals

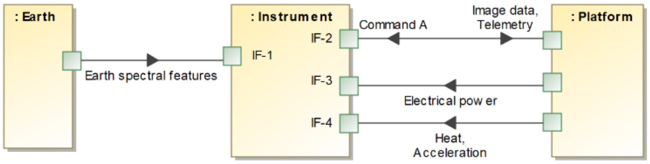

Figure 3. Example of Logical Capture of Interfaces

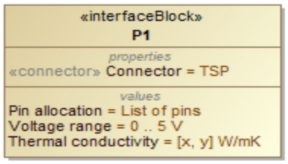

Figure 4. Example of Use of an InterfaceBlock to Capture the Characteristics of an Interface

Simultaneity of requirements applicability is captured by extending the use of SysML state machine diagrams rather than capturing all requirements in one large sequence diagram. This is defined as a *mode requirement*, which captures all requirements that "do not have conflicting requirements and that must be fulfilled simultaneously" (Salado & Wach, 2019b). An example of this is shown in Figure 5. In this example, the *Accept external energy* requirement and the *Compute tasks* requirements need to be fulfilled simultaneously.

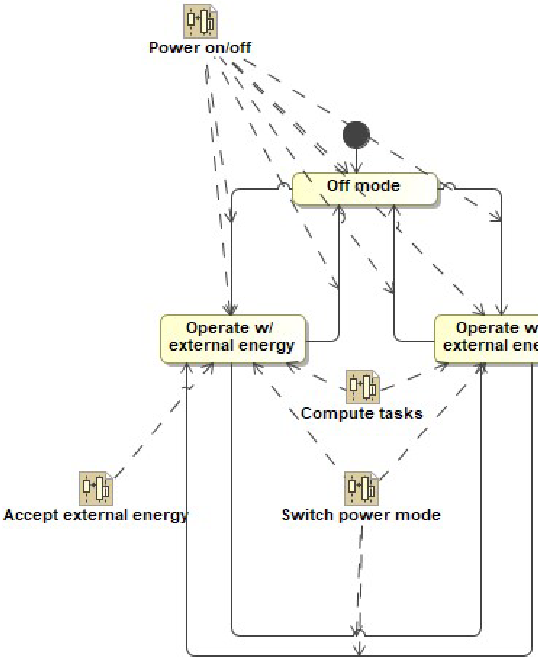

Figure 5. Example of a Mode Requirement Used to Capture Requirement Simultaneity

It is critical to note that the state objects in the state diagram do not represent *states* in the traditional sense of SysML. They are only used in this implementation of TMBR to capture operational scenarios under which different requirements are expected to be fulfilled at the same time. The model does not impose any design constraint for the system, such as what states it will have; such design decision is left open. Eventually, a *real* state machine diagram that captures the actual behavior of a potential system (not the required one) may have a completely different set of state elements (in the diagram) than the mode requirement diagram contains for indicating requirements applicability (i.e., as used in this paper). This is because in a solution model, state elements capture system states, whereas, as indicated, state elements in TMBR capture operational conditions that differ in the requirements that need to be fulfilled simultaneously. This choice avoids unnecessarily constraining the solution space (Wach & Salado, 2020).

Research Method

Participants

A total of 44 participants participated in the study, and 40 participants finished the study and turned in their artifacts. Participation was voluntary and participants received a compensation of $15 per hour, up to a total of $225, for participating in the study. Only adults participated in the study. The following inclusion criteria were used when selecting participants:

- Undergraduate or graduate student in systems engineering at Virginia Tech.
- Undergraduate or graduate student in aerospace engineering at Virginia Tech.

The following exclusion criteria were used when selecting participants:

- Not meeting the inclusion criteria.
- Students registered in a course taught by the authors during the study.
- Minors, prisoners, and adults incapable of consenting on their own behalf.

These criteria were considered appropriate because of three reasons. First, students are easier to recruit than professional engineers and can devote a significant amount of time to the study on short notice. Second, we could control the base knowledge of all participants more easily, factoring out the effects of prior experiences or individual preferences on the results of the study. For example, whereas a professional engineer may confront a conflict between applying a newly learned approach (i.e., model-based requirements) with their experience using a different approach, a student is embedded in a natural dynamic of learning and applying new methods. Third, we did not consider the inexperience of students to be a factor in this study. In fact, such inexperience was a necessary condition for this study because of the difficulty in controlling for the experience of engineers when eliciting requirements.

Determination of compliance of participants to selection criteria was performed by the authors at the start of the study.

Students were assigned to each group randomly. To do this, we separated aerospace engineering students and systems engineering students. Then we randomly split them into two groups. To ensure that all the participants meet the experiment criteria and to avoid any conflicts of interests, we conducted a survey to gather demographic information.

Instruments

Five instruments were developed for use in the experiment (see Table 1). All instruments were developed before the study was initiated. The table is organized sequentially, i.e., the order indicates the sequence in which the different instruments were provided to the participants.

Table 1. Research Instruments Used in the Study

Instrument	Description
Consent form	This instrument was used to inform the participants about the conditions of the study.
Survey	This instrument was used to gather demographic information of the participants.
Training material textual requirements	This instrument consisted of a slide deck and synchronous online presentations by an instructor.
Training material model-based requirements	This instrument consisted of a slide deck, research papers, and synchronous online presentations by an instructor.
Problem description	This instrument was used as a problem statement. It also listed the expected behavior of the participants during the study. It is provided in the Appendix.

Design

In this section, the study design is discussed. This discussion includes the statement of the hypotheses, factors in the design, a discussion on the validity and reliability of the design, detailed procedures for executing the study, and a summary of the data analysis methods.

Hypotheses

The study was designed to test the following three hypotheses:

> H1. Model-based requirements yield fewer unbounded requirements than textual-based requirements.
>
> H2. Model-based requirements yield fewer unnecessary constraints than textual-based requirements.
>
> H3. Model-based requirements achieve higher completeness than textual-based requirements.

All hypotheses focus on the performance of the groups. It was expected that the results would confirm the three hypotheses.

Experimental Design

Two groups of engineers were asked to elicit requirements from potential users of a surveillance and detection system. One group acted as the control group and the other group acted as the experimental group. The control group employed textual requirements, while the experimental group employed model-based requirements.

Each group consisted of 22 students. Eleven students were in the aerospace engineering major and the other 11 were in the systems engineering major. Aerospace engineering students participated to both bring subject matter expertise on the problem statement and prevent investigators' biases on the subject. Each aerospace engineering student was teamed up with another study participant (i.e., a system engineering student). The one-to-one allocation was intended to avoid coupling effects between the study participants, which eases the factorial analyses necessary to test the hypotheses of this study. However, later in the experiment, four students from the model-based requirements engineering dropped out of the study before they finished their artifacts. Therefore, the study consisted of 11 groups of textual requirements and 9 groups of model-based requirements.

After splitting participants into two groups, each group was trained in just one of the methodologies, either model-based requirements or textual requirements. Group 1 received 10-hour training on textual requirements. Group 2 received 10-hour training on Model-based Requirements. This split helps avoiding confounding effects between knowing both methods but applying only one of them. The 10-hour training was divided into two blocks of 5 hours apiece.

Training was not provided by the researchers, but by independent instructors. Training in model-based requirements was provided by an author of a seminal Model-Based Requirement paper. In this way, we could control for adequate learning and application of the model-based requirements framework, by having an instructor who developed such a framework. Training in using textual requirements was provided based on material in (Buede & Miller, 2016; Lee et al., 2009; Wasson, 2016). In this way, we mitigated potential biases that the researchers might have introduced if provided the training of both methods. The instructor had over 5 years of experience in eliciting requirements for large-scale engineered systems and prior experience in conducting this type of training.

An important observation was that during or after the training sessions for Model-Based Requirements (i.e., Group 2), three students dropped out of the experiment. They cited difficulties in understanding Model-Based Requirements Engineering as the main reason for their decision. Therefore, we started with 22 students for each group and ended up with 22 students in Group 1 and 18 students in Group 2.

The one-on-one study was conducted in five 1-hour sessions. Each session was separated by 1 week to allow the participant to reflect and process the insights and data collected during the elicitation session.

Each team performed the elicitation sessions in isolation from other participants. The teams performed their task only with the knowledge they gained from the training sessions. That is, the elicitation sessions were conducted sequentially and not for the entire sample at once. No outsider source was used during the elicitation process. To ensure that the teams worked in isolation with no outside help, all the sessions were video recorded.

The surveillance and detection need of the case study defined in Larson et al. (2009) was used as the problem statement. The hypothesis was tested using stakeholder needs for surveillance and detection of fire over the U.S. map. In the study, participants in both groups developed a requirement set for an Earth Observation Satellite. The stakeholder need was to build a system that could detect and monitor potentially dangerous wildfires throughout the United States. This satellite would survey the United States daily to give the Forest Service a means for earlier detection to increase the probability of containment and to save lives and property.

A survey was employed to gather demographic information of the participants.

Factors in the Design

The independent variable in this investigation is the *requirements approach*. There are two alternatives: (1) textual requirements and (2) model-based requirements as defined in Salado & Wach, 2019b.

Four dependent variables were measured:

- Number of inapplicable requirements. This variable provides a measure of the actual effectiveness of both the control method and the experimental method to elicit inapplicable-free requirements.
- Number of unnecessary constraints. This variable provides a measure of the actual effectiveness of both the control method and the experimental method to elicit unnecessary requirements, such as solution-dependent ones.
- Number of unbounded requirements. This variable provides a measure of the actual effectiveness of both the control method and the experimental method to elicit adequately bounded requirements.
- Level of completeness of the requirement set. This variable provides a measure of the completeness of the resulting requirement sets when using both the method employed by the control group and the method employed by the experimental group.

Effects related to experience, competence, and specific knowledge were controlled by the inclusion and exclusion criteria of the participants.

In designing the experiment, several constraints were imposed that could have restricted the ways in which the independent variables could be manipulated. Three primary factors constrained the experiment:

- Time. The elicitation problem was limited to 5 hours per problem. This is considered much lower than what would be allocated in a real-life development for the given system of interest. Therefore, this limitation poses a threat to completeness in the elicitation effort. However, since all participants are subjected to the same limitation, we suggest that the effectiveness of the method can still be measured.
- Participants. First, the elicitation activity was performed in isolation (that is, one analyst and one stakeholder) and not in teams of analysts. This is not necessarily representative of a real-life development for the given system of interest. Therefore, this limitation poses a threat to correctness in the elicitation effort due to potential lack of domain knowledge. However, since all participants are subjected to the same limitation, we suggest that the effectiveness of the method can still be measured.
- Single domain. The problems only address one type of system, a satellite. This poses a threat to generality of the results.

Threats to Internal Validity

In an internally valid experiment, the relationships between observed differences on the independent variable are a direct result of the manipulation of the independent variable, not some other variable. Table 2 lists internal threats to validity and their potential interference with the experimental design, if any.

Table 2. Threats to Internal Validity

Threat	Factor	Justification
Ambiguous temporal precedence	No	The cause variable (requirements formulation method) was used as an input to create the effect variable (formulated requirements).
Confounding	No	Prior knowledge and experience in the field of systems engineering was controlled through recruiting.
History	No	The study was conducted in a short time and no extraneous event was recorded during the study.
Maturation	Yes (to be mitigated)	The study was conducted in several sessions separated in time. Although participants were instructed to not read or learn anything on the topics relevant to the study until their responses were delivered, the researchers had no mechanism to control maturation. However, the video recording of the different sessions could indicate if maturation happened. This assessment has not been done for this paper.
Repeated testing	No	No pre-test was given.
Instrumentality	Potentially	Factor: The researchers could link the artifact under evaluation to the different groups. Mitigations: The experiment did not use pre-test instruments in conjunction with post-test instruments.
Statistical regression	No	Random allocation of participants to groups.
Selection bias	No	(1) Groups were randomly created. (2) Pre-testing was not performed.
Mortality	No	All participants that conducted the first 1-to-1 session completed the study.
Selection-Maturation interaction	No	Random allocation of participants to groups.
Diffusion	Yes	Although participants were instructed to not exchange any information or opinion about the experiment with other participants until cleared out by the researcher, the researcher had no mechanism to control it.
Compensatory rivalry/resentful demoralization	No	(1) The experiment did not have intermediate results gates. (2) Participants did not have access to the results of the other group.
Experimenter bias	Yes (mitigated)	No verbal interaction between the researcher and the participants regarding the experiment besides pre-produced instruments and minor clarifications about the expected deliverables.

Threats to External Validity

In an externally valid experiment, the results are generalizable to groups and environments outside of the experimental setting. Table 3 lists internal threats to validity and their potential interference with the experimental design, if any.

Table 3. Threats to External Validity

Threat	Factor	Justification
Pre-test treatment interaction	No	No pre-test was given.
Multiple treatment interference	No	There was a single treatment in the study (one training session before the executing of the requirements elicitation activity).
Selection-treatment interaction	Yes	All participants were students from controlled departments.
Reactivity and Situation	Yes	
Rosenthal effects	No	(1) Instruments prepared before the experiment and provided in written form. (2) Participants were randomly assigned to groups. (3) Administration of treatment (instruction) was not performed by the researchers. (4) Evaluation criteria not defined by the researchers.

Data Analysis

For textual requirements, requirements were de-categorized and compiled as a single list for each participant. For model-based requirements, requirements were transformed into individual statements using the template described in Salado and Wach (2019a) and consolidated as a single list for each participant.

Each requirement in each list was evaluated and classified as inapplicable, unbounded, unnecessary constraint, or adequate. The following criteria, which were derived from industry guidelines (INCOSE, 2012) and used in prior research for the same purpose (Salado & Nilchiani, 2017a), were used to classify the requirements:

- *Inapplicable requirement*: A requirement that addresses a system external to the system of interest. Indications of this include statements where the subject of the requirement is not the system of interest or one of its parts or the requirement addresses aspects of the development process.
- *Unbounded requirement*: A requirement that the system of interest cannot fulfill on its own, but which fulfillment depends on the action of systems external to the system of interest. An indication of this is that the statement contains more than one system.
- *Unnecessary constraint*: A requirement that enforces a particular design solution. Indications of this include the use of terms such as *use*, *be composed of*, *consist of*, or *include*, among others, or the use of a system's part as the subject of the statement.
- *Adequate requirement*: Any requirement that is not classified in any of the other three.

Completeness was assessed in two steps. First, all generated requirements were aggregated, consolidating those that refer to the same aspect or mutually exclusive aspect that the system had to fulfill as a single requirement. Second, the coverage of each set of requirements from the participants was assessed against this aggregated list. While completeness of a requirement set cannot be proven (Carson, 1998; Carson et al., 2004; Carson & Shell, 2001), we suggest, as in Salado and Nilchiani (2017a), that this coverage provides valuable insights about how the different approaches might impact completeness.

These evaluations were performed independently by both authors and then consolidated. The authors knew the approach used to define the requirements when performing the evaluation, which leads to some of the biases described in the section Threats to Internal Validity.

Descriptive statistics were used to characterize the responses of the participants, as well as their demographics. Inapplicable requirements were removed from the comparisons of unbounded requirements, unnecessary constraints, and completeness to enable fair comparisons between the two approaches. While inferential statistics were initially planned to quantitatively compare both approaches, we found some problems to process the deliverables of the group using model-based requirements. Instead, a qualitative assessment was performed.

Results

In this section, we show preliminary results of a subset of the gathered data during the experiment. In particular, we randomly picked the deliverables of three participant pairs from the control group and three from the experiment group.

Group Composition

Figure 6 and Figure 7 show the distributions of prior experience using textual requirements and using MBSE, respectively. While experience in textual requirements among the two groups was a bit imbalanced, the group had, in general, little or no experience. A similar situation was given with respect to experience in MBSE. Comparing both the control and the experimental groups, the experience relevant to their approach was similar.

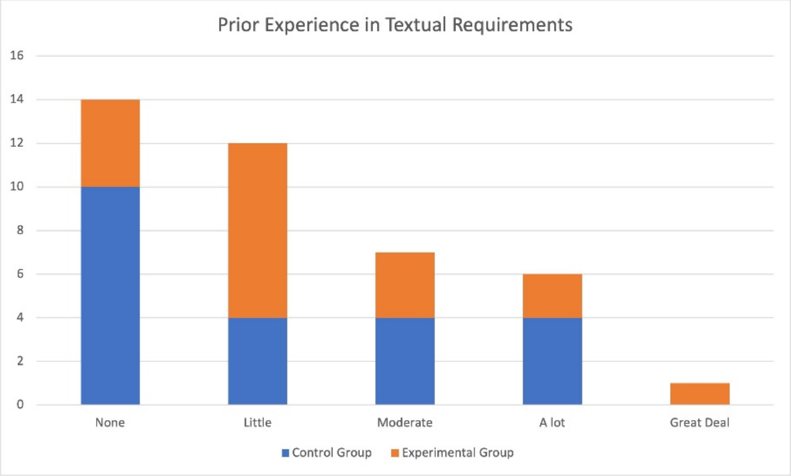

Figure 6. Prior Experience Using Textual Requirements

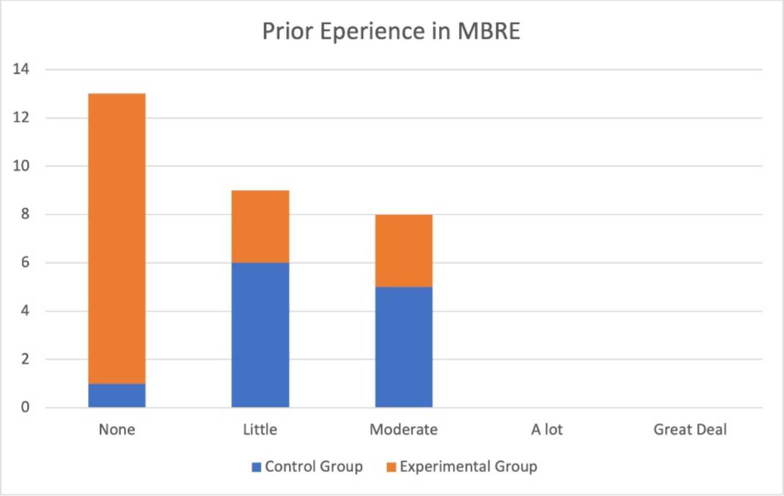

Figure 7. Prior Experience Using MBSE

Figure 8 shows the distribution of prior experience in designing or working with space systems among the different groups. The responses indicated more dispersion, which could be explained by the fact that around half of the participants were aerospace students and half not aerospace students. It should be restated that each pair of participants in each group were formed by one aerospace student and non-aerospace student.

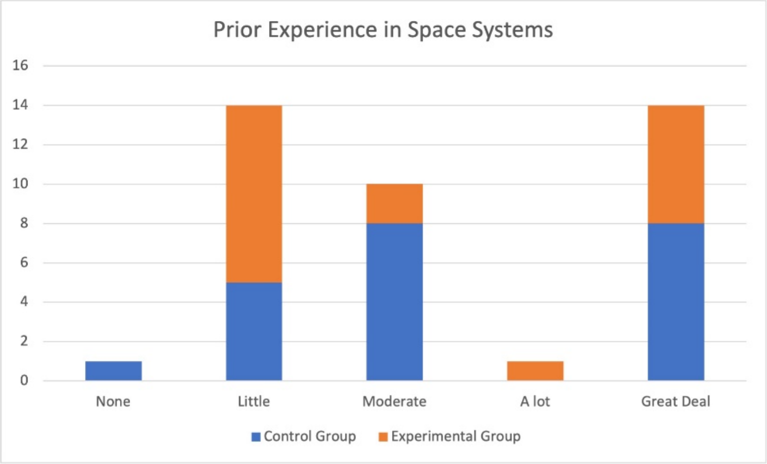

Figure 8. Prior Experience in Designing or Working with Space Systems

Evaluation of Textual Requirements

Table 4 shows the results of assessing the requirements delivered by three participant pairs in terms of the number of unbounded requirements, inapplicable requirements, and unnecessary constraints. Table 5 shows a few examples of such requirements directly taken from the participants' responses.

Table 4. Summary Assessment of Textual Requirements

Project	Total Req	Unbounded Req	Unnecessary constraint	Inapplicable Req	Adequate Req
1	45	16	10	16	4 (10%)
2	145	49	40	55	7 (5%)
3	98	27	41	1	32 (33%)

Table 5. Requirement Examples from the Participants' Responses

Unbounded requirements	The system shall provide space-based "fire-scouts" that survey the United States daily. The system shall provide space-based "fire-scouts" that survey the United States daily. The satellite shall be deployed in low Earth orbit.
Unnecessary constraints	The antenna shall allow the satellite to communicate with the ground. Propellant shall be an ionized thrust that can be recharged using solar cells. The satellite shall utilize GPS.
Inapplicable requirements	The rockets shall withstand temperatures from XXX-to-XXX degree Fahrenheit. Separation shall occur once satellite is in specified orbit. The satellite shall pass all the Vega Launch Vehicle Manual's quality inspection requirements.

To evaluate the fidelity of the requirements activity, we compare the results presented in Table 4 with those obtained in a similar study, which was conducted by one of the authors with professional engineers and is reported in Salado and Nilchiani (2017a). The comparison is presented in Table 6. While data are not available in Salado and Nilchiani (2017a) for unbounded requirements, participants employed textual requirements a bit less effectively (that is, with more unnecessary constraints and inapplicable requirements). We suggest that this difference is, however, not dramatic, and consider that the participants correctly used the training to formulate textual requirements.

Table 6. Comparison Against Performance of Practicing Engineers (Salado & Nilchiani, 2017a)

Variable		**Practicing Engineers**	**This Experiment**
Relative number of unbounded requirements	Mean	n/a	32%
	Median	n/a	34%
Relative number of unnecessary constraints	Mean	27%	31%
	Median	26%	28%
Relative number of inapplicable requirements	Mean	16%	25%
	Median	18%	36%

Evaluation of Model-Based Requirements

Two of the three responses we randomly selected for this pilot evaluation show significant misuse of TMBR. Two main issues were found. First, some participants used signal elements to represent actions instead of items that are exchanged between systems. An example of this, directly taken from a participant's response, is shown in Figure 9. Second, some participants decomposed the system of interest into its components. An example is shown Figure 10. Therefore, they cannot be used in the assessment.

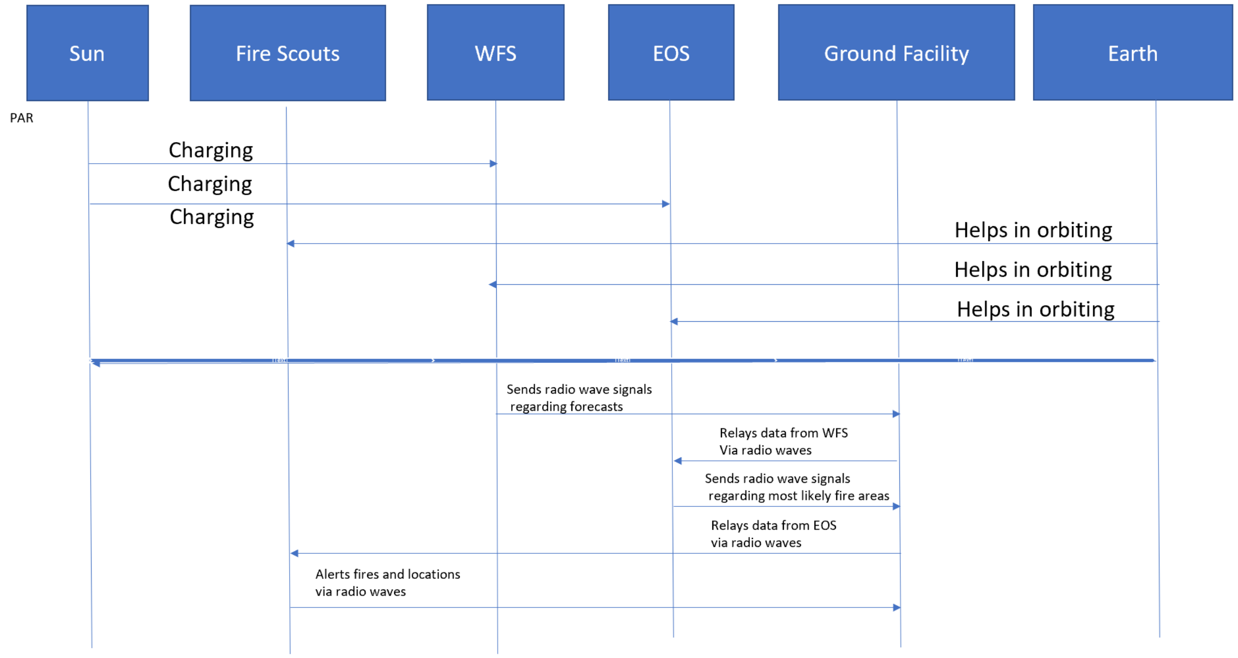

Figure 9. Example of TMBR Misuse: Use of Actions Instead of Signals in Exchanges Between Systems

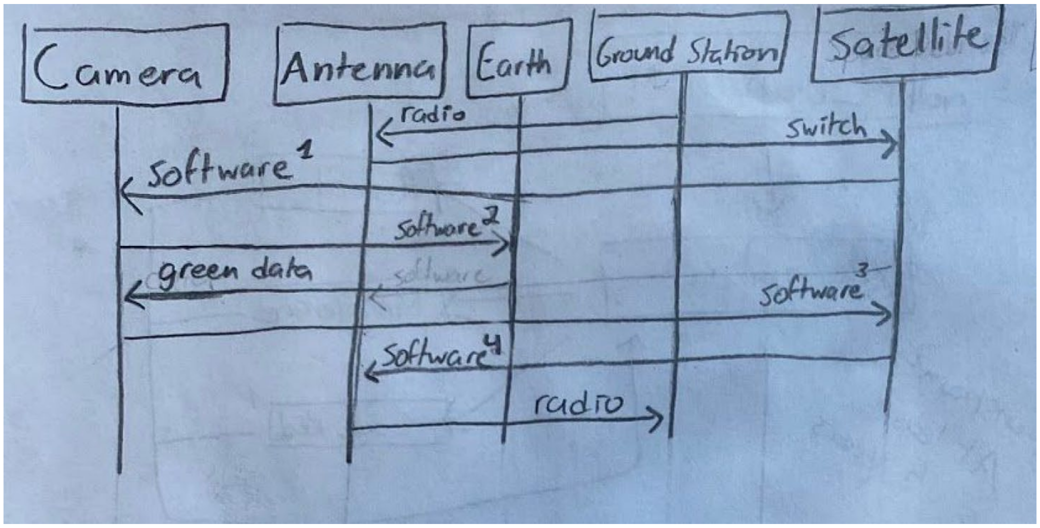

Figure 10. Example of TMBR Misuse: Decomposition of System of Interest into Components.

The third response, while adequately using TMBR, did not generate complete models (a partial example is shown in Figure 11). Specifically, the characteristics of the signals and the interfaces through which they ought to be exchanged were not modeled. Yet, we consider that this response is valid for the pilot assessment, since effects on the hypotheses listed in the section Hypotheses can be evaluated.

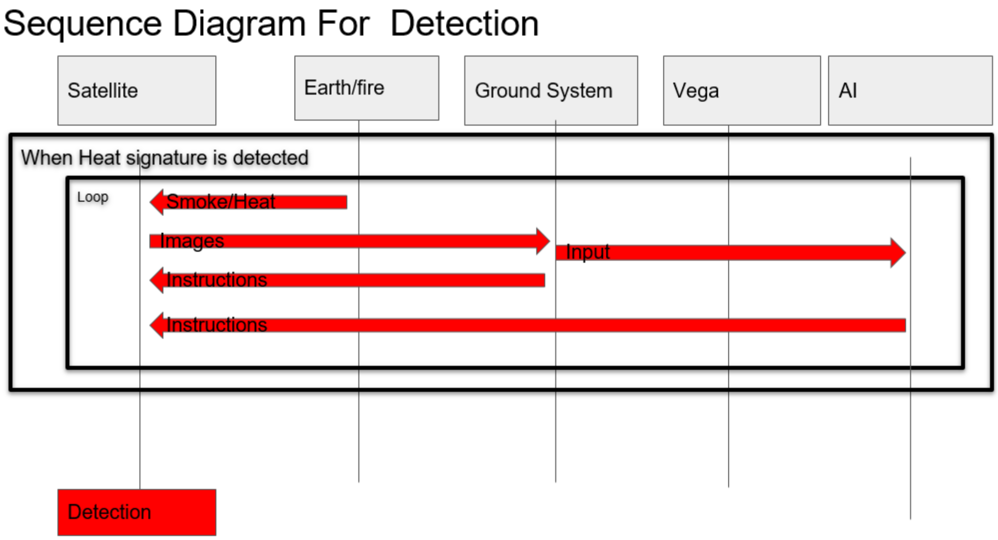

Figure 11. Example of Adequate Yet Incomplete Use of TMBR

The requirement models led to 31 unique requirement statements, none of which was an unbounded requirement, an unnecessary constraint, or an inapplicable requirement. A few examples are shown here:

R1. The satellite shall accept force of launch.
R2. The satellite shall accept instructions 1 when heat signature is detected.
R3. The satellite shall provide images when heat signature is detected.

As discussed, according to TMBR, the models should have incorporated required properties of the items and of the interfaces through which they are transferred. For example, the item “force of launch” in R1 should have been completed with the actual mechanical forces injected into the satellite (e.g., a random vibration profile). Similarly, the models should have been completed with the description of the satellite’s attachment/physical point through which such vibration profile was to be injected. These were not provided in the participant’s response.

Comparison

The performance of the control and experiment groups with respect to the hypotheses listed in Section 3.3.1 are compared in Table 7. While these results are preliminary, given the small size of the sample and the problems encountered with the responses in the experiment group, they provide some indication of the superiority of model-based requirements to textual requirements. In terms of unbounded requirements, inapplicable requirements, and unnecessary constraints, these results were expected because such types of requirements are avoided by design in TMBR (Salado & Wach, 2019b).

Table 7. Results Comparison Between Control and Experiment Groups

Variable		**Control Group (textual reqs)*** *(sample size = 3)*	**Experiment Group (TMBR)** *(sample size = 1)*
Relative number of unbounded requirements	Mean	32%	0%
	Median	34%	0%
Relative number of unnecessary constrains.	Mean	31%	0%
	Median	28%	0%
Relative number of inapplicable requirements	Mean	25%	0%
	Median	36%	0%
Adequate requirements	Mean	14	31
	Median	7	31
Coverage	Mean	26%	51%

*Note: As shown in Table 4, none of the responses in the sample achieved 0% performance in any of the variables, and the maximum number of adequate requirements obtained was 32 with a 54% coverage.

For completeness, in absolute terms, using model-based requirements led to more adequate requirements than using textual requirements. To assess coverage, the requirements from every participant were aggregated to a total of 57 adequate requirements. Model-based requirements achieved higher coverage than textual requirements, although one of the responses in the control group achieved a coverage of around 54%. However, it is important to note that the coverage achieved in the experiment group may not have a high fidelity due to the incompleteness in the models, as discussed earlier.

5. Conclusions and Future Research

We have presented in this paper preliminary results of an experimental study to evaluate the contractual safety of model-based requirements as compared to textual requirements. The preliminary results reported here have been limited to a subset of the all the data collected and to just four variables: number of unbounded requirements, inapplicable requirements, and unnecessary constraints, and coverage. The preliminary data support the claim that model-based requirements are superior in all variables to textual requirements.

However, some issues were encountered with the application of model-based requirements. Particularly, participants failed to use the modeling rules of TMBR, the modeling paradigm for requirements used in this study. We interviewed the TMBR instructor after reviewing the responses from the participants to better understand the results of the training prior to the start of the requirements definition activity. According to the instructor, the two issues described in the previous section were common misunderstandings he encountered during the whole training, and while he addressed them several times, he was not confident that the participants fully grasped them at the conclusion of the training.

We have reviewed the material that was used for the training and conjecture three potential causes. First, the material that was used for training relied heavily on academic papers, which might have been too hard to process for the participants, who were undergraduate students. Second, the participants in charge of modeling the requirements were primarily students in an industrial engineering program, being biased toward process flows over design. Third, TMBR may require more training time than initially scoped. The experience gained in this pilot study informs the development of dedicated training material that is more easily digested by participants, including an evaluation gate during the training and potentially increased training duration.

Finally, we note that future work is planned to complete the evaluation with the full set of responses and to include measures related to precision and accuracy with which the requirements were captured in textual forms and with models.

Appendix

We want to develop an Earth Observation Satellite. Our goal is to build a system that can detect and monitor potentially dangerous wildfires throughout the US. wildfires claim hundreds of lives, threat thousands more, and lay waste to millions of acres, causing losses in the billions of dollars. The system needs to provide space-based “fire-scouts” that would survey the US daily to give the Forest Service a means for earlier detection to increase the probability of containment and to save lives and property.

The following needs, among others you consider necessary, are to be taken into account:

1. Satellite will be placed in LEO.
2. Continuous monitoring.
3. Be operational for at least 5 years.
4. Launched onboard Vega. (Link to the actual user manual was provided.)
5. Use Space@VT ground station. (Link to a datasheet of the actual ground station was provided.)

If you need to quantify any value, use notional ones and specify your assumptions. No need for actual analyses.

Please derive the requirements for this system.

RULES:

1. You cannot use or read requirements related to similar satellite
2. Report all information that you have used for this activity
3. Ideally, you just use your own knowledge
4. No work in between sessions.
5. No learning about requirements engineering during the whole study. If required as part of formal coursework, let us know.
6. You cannot talk to anyone during the duration of the study about the study, not even your peer in the session.
7. You should brainstorm with your peer in each session.
8. You should hand in the final result of requirement derivation process after all the five sessions are finished.

References

Aceituna, D., Do, H., Walia, G. S., & Lee, S. (2011). *Evaluating the use of model-based requirements verification method: A feasibility study*. Paper presented at the Workshop on Empirical Requirements Engineering (EmpiRE 2011), Trento.

Aceituna, D., Walia, G., Do, H., & Lee, S.-W. (2014). Model-based requirements verification method: Conclusions from two controlled experiments. *Information and Software Technology*, *56*(3), 321–334. https://doi.org/10.1016/j.infsof.2013.11.004

Adedjouma, M., Dubois, H., & Terrier, F. (2011, April 27–29). *Requirements exchange: From specification documents to models.* Paper presented at the Proceedings of the 16th IEEE International Conference on Engineering of Complex Computer Systems, Las Vegas, NV.

Badreddin, O., Sturm, A., & Lethbridge, T. C. (2014). *Requirement traceability: A model-based approach*. Paper presented at the 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE), Karlskrona. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6890829&isnumber=6890816

Borgne, A. L., Belloir, N., Bruel, J., & Nguyen, T. (2016). *Formal requirements engineering for smart industries: Toward a model-based graphical language*. Paper presented at the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France. http://ieeexplore.ieee.org.ezproxy.lib.vt.edu/stamp/stamp.jsp?tp=&arnumber=7816956&isnumber=7816796

Buede, D. M., & Miller, W. D. (2016). *The engineering design of systems: Models and methods* (3rd ed.). John Wiley & Sons.

Carson, R. S. (1998). Requirement completeness: A deterministic approach. *INCOSE International Symposium*, *8*, 738–746.

Carson, R. S., Aslaksen, E., & Gonzales, R. (2004). *Requirements completeness.* Paper presented at the INCOSE International Symposium.

Carson, R. S., & Shell, T. (2001). Requirements completeness: Absolute or relative? *Systems Engineering*, *4*, 230–231.

Fockel, M., & Holtmann, J. (2014). *A requirements engineering methodology combining models and controlled natural language*. Paper presented at the 2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE), Karlskrona. http://ieeexplore.ieee.org.ezproxy.lib.vt.edu/stamp/stamp.jsp?tp=&arnumber=6890827&isnumber=6890816

Friedenthal, S., Moore, A., & Steiner, R. (2015). *A practical guide to SysML—The systems modeling language* (3rd ed.). Morgan Kaufman.

GAO. (2016). *Defense acquisitions—Assessments of selected weapon programs* (GAO-16-329SP).

Gervasi, V., & Zowghi, D. (2005). Reasoning about inconsistencies in natural language requirements. *ACM Transactions on Software Engineering and Methodology*, *14*(3), 277–330.

Gilmore, J. M. (2011). Key issues causing program delays in defense acquisition. *ITEA Journal*, *32*, 389–391.

Holder, K., Zech, A., Ramsaier, M., Stetter, R., Niedermeier, H.-P., Rudolph, S., & Till, M. (2017). Model-based requirements management in gear systems design based on graph-based design languages. *Applied Sciences*, *7*(11), 1112. https://doi.org/10.3390/app7111112

Holt, J., Perry, S., Brownsword, M., Cancila, D., Hallerstede, S., & Hansen, F. O. (2012). *Model-based requirements engineering for system of systems*. Paper presented at the 2012 7th International Conference on System of Systems Engineering (SoSE), Genova, Italy. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6384145&isnumber=6355890

Holt, J., Perry, S., Payne, R., Bryans, J., Hallerstede, S., & Hansen, F. O. (2015). A model-based approach for requirements engineering for systems of systems. *IEEE Systems Journal*, *9*(1), 252–262. https://doi.org/10.1109/JSYST.2014.2312051

INCOSE. (2012). *Guide for writing requirements*.

INCOSE. (2015). *Systems engineering handbook: A guide for system life cycle processes and activities* (version 4.0). John Wiley & Sons.

Larson, W. J., Kirkpatrick, D., Sellers, J. J., Thomas, D., & Verma, D. (2009). *Applied space systems engineering.* McGraw Hills.

Lee, R. E., Hooks, I., & Barker, B. G. (2009). Engineering and managing system requirements. In W. J. Larson, D. Kirkpatrick, J. J. Sellers, L. D. Thomas, & D. Verma (Eds.), *Applied space systems engineering* (pp. 95–139). McGraw Hill.

Lu, C., Chang, C., Chu, W. C., Cheng, Y., & Chang, H. (2008). *A requirement tool to support model-based requirement engineering*. Paper presented at the 2008 32nd Annual IEEE International Computer Software and Applications Conference, Turku. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4591654&isnumber=4591503

Marschall, F., & Schoemnakers, M. (2003). *Towards model-based requirements engineering for web-enabled B2B applications*. Paper presented at the 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Huntsville, Alabama. http://ieeexplore.ieee.org.ezproxy.lib.vt.edu/stamp/stamp.jsp?tp=&arnumber=1194813&isnumber=26873

Micouin, P. (2008). Toward a property based requirements theory: System requirements structured as a semilattice. *Systems Engineering*, *11*(2).

Miotto, B. L., & a. P. (2014). *Model-based requirement generation*. Paper presented at the 2014 IEEE Aerospace Conference, Big Sky, MT.

Mordecai, Y., & Dori, D. (2017). *Model-based requirements engineering: Architecting for system requirements with stakeholders in mind*. Paper presented at the 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8088273&isnumber=8088244

Ouchani, S., & Lenzinia, G. (2014). *Attacks generation by detecting attack surfaces*. Paper presented at the 5th International Conference on Ambient Systems, Networks and Technologies.

Pandian, M. K. S., Suri, K., Cadavid, J., Barosan, I., Brand, M., Alférez, M., & Gérard, S. (2017, April 3–7). *Towards industry 4.0: Gap analysis between current automotive mes and industry standards using model-based requirement engineering.* Paper presented at the Proceedings of the 2017 IEEE International Conference on Software Architecture Workshops (ICSAW).

Pennock, M. J., & Wade, J. P. (2015). The top 10 illusions of systems engineering: A research agenda. *Procedia Computer Science, 44*, 147–154. https://doi.org/10.1016/j.procs.2015.03.033

Reza, H., Sehgal, R., Straub, J., & Alexander, N. (2017). *Toward model-based requirement engineering tool support*. Paper presented at the 2017 IEEE Aerospace Conference, Big Sky, MT. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7943647&isnumber=7943554

Ribeiro, F. G. C., Pereira, C. E., Rettberg, A., & Soares, M. S. (2018). Model-based requirements specification of real-time systems with UML, SysML and MARTE. *Software & Systems Modeling*, *17*(1), 343–361. https://link.springer.com/article/10.1007/s10270-016-0525-1

Saadatmand, M., Cicchetti, A., & Sjödin, M. (2012). *Toward model-based trade-off analysis of non-functional requirements.* Paper presented at the 2012 38th Euromicro Conference on Software Engineering and Advanced Applications, Cesme, Izmir. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6328142&isnumber=6328119

Salado, A., & Nilchiani, R. (2014). A categorization model of requirements based on Max-Neef's model of human needs. *Systems Engineering*, *17*(3), 348–360. https://doi.org/10.1002/sys.21274

Salado, A., & Nilchiani, R. (2017a). Reducing excess requirements through orthogonal categorizations during problem formulation: Results of a factorial experiment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(3), 405–415. https://doi.org/10.1109/TSMC.2015.2502953

Salado, A., & Nilchiani, R. (2017b). The tension matrix and the concept of elemental decomposition: improving identification of conflicting requirements. *IEEE Systems Journal*, *11*(4), 2128–2139. https://doi.org/10.1109/JSYST.2015.2423658

Salado, A., & Nilchiani, R. (2017c). Understanding how problem formulation in systems engineering influences systems affordability: A systems thinking investigation. *International Journal of Applied Systemic Studies*, *7*(4).

Salado, A., Nilchiani, R., & Verma, D. (2017). A contribution to the scientific foundations of systems engineering: Solution spaces and requirements. *Journal of Systems Science and Systems Engineering*, *26*(5), 549–589. https://doi.org/10.1007/s11518-016-5315-3

Salado, A., & Wach, P. (2019a). *Automatic generation of contractual requirements from MBSE artifacts.* Paper presented at the 16th Annual Acquisition Research Symposium, Monterey, CA.

Salado, A., & Wach, P. (2019b). Constructing true model-based requirements in SysML. *Systems*, *7*(2), 19.

Schmitz, D., Nissen, H. W., Jarke, M., & Rose, T. (2010). *Relating domain model based requirements management and situational method engineering*. Paper presented at the 2010 Third International Workshop on Managing Requirements Knowledge, Sydney, NSW.

Siegl, S., Hielscher, K., & German, R. (2010). *Model based requirements analysis and testing of automotive systems with timed usage models*. Paper presented at the 2010 18th IEEE International Requirements Engineering Conference, Sydney, NSW. http://ieeexplore.ieee.org.ezproxy.lib.vt.edu/stamp/stamp.jsp?tp=&arnumber=5636645&isnumber=5636534

Soares, M. S., & Vrancken, J. (2008). Model-driven user requirements specification using SysML. *Journal of Software*, *3*(6), 57–68.

Teufl, S., Mou, D., & Ratiu, D. (2013). *MIRA: A tooling-framework to experiment with model-based requirements engineering*. Paper presented at the 2013 21st IEEE International Requirements Engineering Conference (RE), Rio de Janeiro.

Tjong, S. F., Hallam, N., & Hartley, M. (2006, September 20–22). *Improving the quality of natural language requirements specifications through natural language requirements patterns.* Paper presented at the Sixth IEEE International Conference on Computer and Information Technology.

Wach, P., & Salado, A. (2020). *The need for semantic extension of SysML to model the problem space*. Paper presented at the Conference on Systems Engineering Research (CSER), Redondo Beach, CA.

Wach, P., & Salado, A. (2021a). *True model-based requirements (TMBR) of a satellite TTC transponder.* Paper presented at the IEEE Aerospace Conference.

Wach, P., & Salado, A. (2021b). True model-based requirements (TMBR): Application to an Earth observation satellite. In *AIAA Scitech 2021 Forum*.

Wasson, C. S. (2016). *Systems engineering analysis, design, and development—Concepts, principles, and practices*. John Wiley & Sons.

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

www.acquisitionresearch.net