2021-08

Phase 2: Investigation of Leading Indicators for Systems Engineering Effectiveness in Model-Centric Programs

Rhodes, Donna H.

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/70194

MIT-SE-21-242

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Phase 2: Investigation of Leading Indicators for Systems Engineering Effectiveness in Model-Centric Programs

18 August 2021

Dr. Donna H. Rhodes

Massachusetts Institute of Technology

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. HQ0034-20-1-0008. 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.

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

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

Abstract

This technical report summarizes the work conducted by Massachusetts Institute of Technology under contract award HQ0034-20-1-0008 during the performance period May 22, 2020 – July 31, 2021. Digital engineering transformation changes the practice of systems engineering, and drives the need to re-examine how engineering effectiveness is measured and assessed. Early engineering metrics were primarily lagging measures. More recently leading indicators have emerged that draw on trend information to allow for more predictive analysis of technical and programmatic performance of the engineering effort. By analyzing trends (e.g., requirements volatility) in context of the program's environment and known factors, predictions can be forecast on the outcomes of certain activities (e.g., probability of successfully passing a milestone review), thereby enabling preventative or corrective action during the program.

Augmenting a companion research study under contract HQ0034-19-1-0002 on adapting and extending existing systems engineering leading indicators, this study takes a future orientation. This report discusses how base measures can be extracted from a digital system model and composed as leading indicators. An illustrative case is used to identify how the desired base measures could be obtained directly from a model-based toolset. The importance of visualization and interactivity for future leading indicators is discussed, especially the potential role of visual analytics and interactive dashboards. Applicability of leading edge technologies (automated collection, visual analytics, augmented intelligence, etc.) are considered as advanced mechanisms for collecting and synthesizing measurement data from digital artifacts. This research aims to provide insights for the art of the possible for future systems engineering leading indicators and their use in decision-making on model-centric programs. Several recommendations for future research are proposed extending from the study.

THIS PAGE INTENTIONALLY LEFT BLANK

About the Author

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

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

THIS PAGE INTENTIONALLY LEFT BLANK

MIT-SE-21-242

ACQUISITION RESEARCH PROGRAM SPONSORED REPORT SERIES

Phase 2: Investigation of Leading Indicators for Systems Engineering Effectiveness in Model-Centric Programs

18 August 2021

Dr. Donna H. Rhodes

Massachusetts Institute of Technology

Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. HQ0034-20-1-0008. 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

This report *HQ0034-20-1-0008 Phase 2: Investigation of Leading Indicators for Systems Engineering Effectiveness in Model-Centric Programs* discusses results of an exploratory investigation related to systems engineering leading indicators. The focus of this research has been on the future, investigating opportunities for existing and new leading indicators afforded by use of model-based toolsets, as well as applying leading-edge techniques to collect, compose and display measurement data for proactively assessing engineering effectiveness on model-based acquisition programs.

This research was performed by Massachusetts Institute of Technology. Involved research team members include: Dr. Donna H. Rhodes, Principal Investigator; Dr. Eric Rebentisch, Research Associate; Mr. Allen Moulton, Research Engineer; and Dr. Adam Ross, Research Scientist.

A related research investigation under the NPS Acquisition Research Program, HQ0034-19-1-0002: *Investigation of Leading Indicators for Systems Engineering Effectiveness in Model-Centric Programs,* was initiated prior to this work and continued in parallel with this research project. It focused on the present, investigating adaptation and extension of existing leading indicators for model-centric programs.

Selected background information on leading indicators is included in each of these reports.

Background

Defense programs have long used engineering metrics to provide status and historical information, but implementation has been limited by the nature of the traditional, document-based engineering approach. Further, early systems engineering metrics were primarily lagging measures, providing information for the next program instead of the current one. Systems engineering leading indicators were subsequently developed to allow for more timely predictive analysis of the technical and programmatic performance of the engineering effort on a program. Leading indicators use an approach that draws on trend information to allow for more predictive insight (Rhodes, Valerdi, & Roedler, 2009).

A systems engineering leading indicator is a measure for evaluating the effectiveness of how a specific program activity impacts engineering effectiveness, which may affect the system performance objectives. Both lagging and leading indicators are found to be useful in many fields (e.g., economic, health, social science). (Zheng, et al., 2019). While lagging measures (e.g., system defects) continue to provide useful information over time for an enterprise, they are insufficient for real-time decisions during a program. Relatively little evidence exists on the application of leading indicators in the engineering of systems. The value of leading indicators comes from examining trends (e.g., requirements volatility) in context of the program's characteristics and known factors. This information can be used to make predictions to forecast the outcomes of certain activities (for example, likelihood of successfully passing a milestone review). Leading indicators have provided some improved ability to assess ongoing engineering effort, and where necessary, take preventative or corrective action during the program.

Motivation and Research Approach

The broad motivation for the work is to enable more timely and informed decisions on systems engineering activities and resources. The use of systems engineering measures is a standard part of traditional practice, though its limitations are acknowledged. Systems engineering leading indicators overcome some of the limitations but until recently collecting the underlying data and performing analysis has been constrained by document-driven engineering practice. As the use of model-based systems engineering increases, the increased ease of generating systems engineering leading indicators will make these more tractable for systems programs. Model-based systems engineering (MBSE) is defined as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." (OMG). The transformation to digital engineering has prompted a need to re-examine the systems engineering leading indicators for this new context.

The investigation aims to provide findings for model-centric programs seeking to use the leading indicators, as well as contribute recommendations to inform the larger effort of the systems engineering community to establish the next generation of digital engineering effectiveness measurement.

Digital engineering tools are recognized as a means to increase engineering efficiency (DoD, 2018, p.17) and to provide access to vast data. Leading indicators are especially important to monitoring of effectiveness on a continuous basis, and also to ensure that effectiveness is not compromised for sake of efficiency. The strategy calls for leadership to "establish accountability to measure, foster, demonstrate, and improve tangible results across programs and the enterprise" (DoD, 2018, p. 22). Common enabling technologies used in digital environments to generate, analyze and display measurement data will encourage a common foundation for cross-program comparison and learning.

Existing leading indicators were developed under the document-based engineering approach. The introduction of digital engineering, or model-based engineering, practices has potentially radical or disruptive impact on the processes, tools, and timelines of engineering programs. Rapidly accelerating analytical and design capabilities will likely have limited impact on overall program pace and effectiveness if reviews and decision-making processes fail to adapt to the processes and cadence of digital engineering and management. Research is necessary in order to understand and adapt existing systems engineering indicators for digital engineering and management practice in model-centric programs. Exploring the art of the possible can reveal how digital system model information could be extracted from toolsets and used to compose base measures into indicators. Further, investigation is needed to understand how newer sciences and technologies (e.g., data science, visual analytics, and interactive dashboards) could better inform timely leadership decisions in model-centric programs.

The proposed research seeks to address these two questions:

- How does digital engineering enable current and new leading indicators to be obtained, composed and made available to program leaders for the purpose of assessment of engineering effectiveness on model-centric programs?

- How can leading- edge technologies (visual analytics, interactive dashboards, etc.) enable leading indicators in model-centric programs?

The digital engineering environment and newer technologies open new possibilities for providing program leaders with leading insights into the effectiveness of systems engineering efforts. Since each of the indicators requires some additional considerations under digital engineering, the first year in this research (under contract HQ0034-19-1-0002) focused on identifying potential modifications and interpretation guidance (Rhodes, 2020). Model-based implications were identified for each of the existing eighteen leading indicators.

For this phase of research (under contract HQ0034-20-1-0008), the research team used literature review and knowledge gathering to explore how information from descriptive system models could be extracted and composed as leading indicators. This includes examining the possibility composing useful sets of leading indicators, and the potential for aggregated indicators. The investigation also explored opportunities for how interactive dashboards could be used to extract and more effectively display measurement information to enhance human decision-making, thereby positively impacting program reviews and decisions.

Knowledge gathering from subject matter experts through technical exchanges and workshops provided insights regarding adaptation of leading indicators and potential new indicators of interest. This included investigation of publications, studies, workshop reports and interim research findings from academic research groups, professional and industry societies and cross-industry initiatives. Literature review is used to explore newer leading edge techniques and approaches for collection and synthesis of measurement data, as enabled by digital engineering practices and environments.

Leading Indicators and Measurement Specifications

The interest in having leading indicators for acquisition and development programs has been discussed within the systems community for some time. In this context, a leading indicator is a measure for evaluating the effectiveness of how a specific activity is applied on a program in a manner that provides information about the impacts of engineering effectiveness that are likely to affect the system performance objectives. Leading indicators are designed to assist program leadership in delivering value to stakeholders, informing interventions and corrective actions to avoid problems, rework and wasted effort. Conventional systems engineering measures provide status and historical information, while leading indicators use an approach that draws on trend information for more predictive insight (Rhodes, Valerdi, & Roedler, 2009).

The foundational work on systems engineering leading indicators was initiated in 2004. The early efforts produced a systems engineering leading indicators guide (Roedler & Rhodes, 2007) with thirteen leading indicators defined using measurement specifications. This work was subsequently evolved through collaboration from organizations and individuals across the systems engineering community with over twenty organizations as contributors. The result was a second version of the guide (Roedler G. J., Rhodes, Schimmoler, & Jones, 2010), with five additional leading indicators and several appendices added. Related studies and papers have been published by various authors (Elm et al., 2008; Rhodes, et al., 2009; Montgomery & Carlson, 2010; Gerst & Rhodes, 2010; Knorr, 2012; Elm & Goldenson, 2013; Gilbert et al., 2014; Orlowski et al., 2015; Shirley, 2016; Orlowski, 2017; Zheng, et al., 2017; Zheng, et al., 2019).

The eighteen leading indicators for systems engineering programmatic and technical performance are:

Requirements Trends: Rate of maturity of the system definition against the plan. Additionally, characterizes the stability and completeness of the system requirements that could potentially impact design, production, operational utility, or support.

System Definition Change Backlog Trends: Change request backlog which, when excessive, could have adverse impact on the technical, cost and schedule baselines.

Interface Trends: Interface specification closure against plan. Lack of timely closure could pose adverse impact to system architecture, design, implementation and/or V&V any of which could pose technical, cost and schedule impact.

Requirements Validation Trends: Progress against plan in assuring that the customer requirements are valid and properly understood. Adverse trends would pose impacts to system design activity with corresponding impacts to technical, cost & schedule baselines and customer satisfaction.

Requirements Verification Trends: Progress against plan in verifying that the design meets the specified requirements. Adverse trends would indicate inadequate design and rework that could impact technical, cost and schedule baselines. Also, potential adverse operational effectiveness of the system.

Work Product Approval Trends: Adequacy of internal processes for the work being performed and also the adequacy of the document review process, both internal and external to the organization. High reject count would suggest poor quality work or a poor document review process each of which could have adverse cost, schedule and customer satisfaction impact.

Review Action Closure Trends: Responsiveness of the organization in closing post-review actions. Adverse trends could forecast potential technical, cost and schedule baseline issues.

Technology Maturity Trends: Risk associated with incorporation of new technology or failure to refresh dated technology. Adoption of immature technology could introduce significant risk during development while failure to refresh dates technology could have operational effectiveness/customer satisfaction impact.

Risk Exposure Trends: Effectiveness of risk management process in managing / mitigating technical, cost& schedule risks. An effective risk handing process will lower risk exposure trends.

Risk Treatment Trends: Effectiveness of the systems engineering organization in implementing risk mitigation activities. If the systems engineering organization is not retiring risk in a timely manner, additional resources can be allocated before additional problems are created.

Systems Engineering Staffing & Skills Trends: Quantity and quality of systems engineering personnel assigned, the skill and seniority mix, and time phasing of their application throughout project lifecycle.

Process Compliance Trends: Quality and consistency of the project defined systems engineering process as documented in SEP/SEMP. Poor/inconsistent systems engineering processes and/or failure to adhere to SEP/SEMP, increase project risk.

Technical Measurement Trends: Progress towards meeting the Measures of Effectiveness (MOEs) / Performance (MOPs) / Key Performance Parameters (KPPs) and Technical Performance Measures (TPMs). Lack of timely closure is an indicator of performance deficiencies in product design and/or team's performance.

Standardizing leading indicators of engineering effectiveness across programs is facilitated through measurement specifications. The systems engineering community has been using measurement specifications for many years, based on foundational work of PSM in software and systems measurement (PSM, 2020). The systems engineering leading indicators initiative adopted the PSM measurement specification format. Accordingly, each of the systems engineering indicators is characterized using a measurement specification with detailed description, insights provided, interpretation guidance and usage guidance. Detailed contents of the measurement specifications for leading indicators is described in Roedler et al. (2010), and summarized in Table 1.

Table 1. Systems Engineering Leading Indicator Specification Fields from (Roedler G. J., Rhodes, D.H., Schimmoler, H. & Jones, C. 2010), adapted by (Zheng L. , et al., 2019).

1.Information need description	
Information need	Specifies what the information need is that drives why we need this leading indicator to make decisions
Information category	Specifies what categories (as defined in the PSM) are applicable for this leading indicator(for example, schedule and progress, resources and cost, product size and stability, product quality, process performance, technology effectiveness, and customer satisfaction)
2. Measurable concept and leading insight	
Measurable concept	Defines specifically what is measurable
Leading insight provided	Specifies what specific insights that the leading indicator may provide in context of the Measurable concept - typically a list of several or more
3. Base measure specification	
Base measures	A list of the base measures that are used to compute one or more leading indicators - a base measure is a single attribute defined by a specified measurement method
Measurement methods	For each base measure, describes the method used to count the base measure,for example simple counting or counting then normalized
Unit of measurement	Describes the unit of measure for each of the base measures
4. Entities and attributes	
Relevant entities	Describes one or more particular entities relevant for this indicator – the object is to be measured (for example, requirement or interface)
Attributes	The function for computing the derived measure from the base measures
5. Derived measure specification	
Derived measure	Describes one or more measures that may be derived from base measures that will be used individually or in combination as leading indicators
Measurement function	The function for computing the derived measure from the base measures
6. Indicator specification	
Indicator description and sample	A detailed specific description and display of the leading indicator,including what base and/or derived measures are used
Thresholds and outliers	Would describe thresholds and outliers for the indicator; this information would be company (and possibly project) specific
Decision criteria	Provides basic guidance for triggers for investigation and when possible action to be taken
Indicator interpretation	Provides some insight into how the indicator should be interpreted; each organization would be expected to tailor this
7. Additional information	
Related processes	Lists related processes and sub-processes
Assumptions	Lists assumptions for the leading indicator to be used, for example, that a requirements database is maintained
Additional Analysis Guidance	Any additional guidance on implementing or using the indicators
Implementation Considerations	Considerations on how to implement the indicator (assume this expands with use by organization)
User of Information	Lists the role(s) that use the leading indicator information
Data Collection Procedure	Details the procedure for data collection
Data Analysis Procedure	Details the procedure for analyzing the data prior to interpretation

In the near term, the existing measurements specifications can be augmented with model-based implications. In the future, modified and new measurement specifications are envisioned in a new release of the leading indicators guide. In this research, composability of leading indicators is considered from the perspective of the composition of base measures into indicators. Leading indicators for assessing the effectiveness of systems engineering on a program are expected to be more tractable and more useful in model-centric programs of the future.

Value of Leading Indicators

Leading indicators provide the most value when they give a proactive assessment that informs programmatic decisions and/or corrective actions. The Requirements Trend indicator, for instance, is used to evaluate trends in the growth, change, completeness and correctness of the definition of system requirements. Traditionally, this indicator provides insight into the rate of maturity of the system definition against the plan. Additionally, it characterizes stability and completeness of the system requirements that could potentially impact design, production, operational utility, or support.

One of the trend indicators, requirements volatility, has been used to drive milestone technical reviews. The graph (Figure 1) illustrates the rate of change of requirements over time.

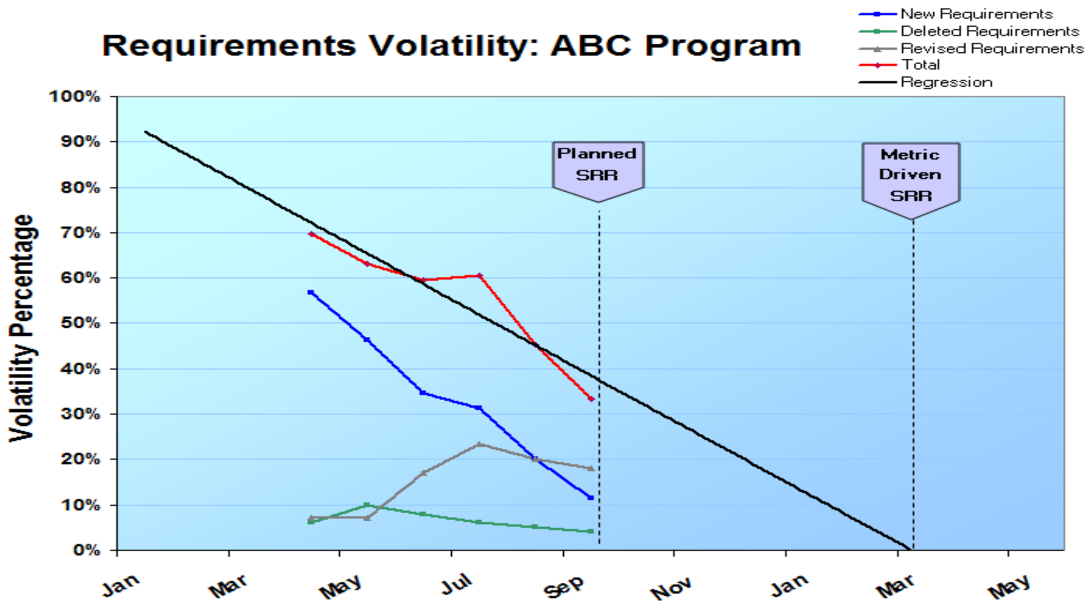


Figure 1. Illustrative Application of Leading Indicators on a Program (Rhodes, Valerdi, & Roedler, 2009)

It also provides a profile of the types of change (new, deleted, or revised) which allows root-cause analysis of the change drivers. By monitoring the requirements volatility trend, the project team was able to predict the readiness for the System Requirements Review (SRR) milestone. In this example, the project team initially selected a calendar date to conduct the SRR, but in subsequent planning made the decision to have the SRR be event driven, resulting in a new date for the review wherein there could be a successful review outcome.

In traditional documentation-based engineering practice, requirements are the central objects used for assessing maturity of system definition. In digital engineering, however, there are many other model constructs (e.g., activity diagrams) that are available and potentially composable to inform assessment of maturity of system definition.

SE Leading Indicators in the Digital Engineering Context

With the transformation from traditional (document-based) engineering to digital engineering, there is a need to consider impacts in regard to how leading indicators are generated and used. The topic of composability has been explored in this effort, and is an area for additional future research.

Composability

Leading indicators involve base measures and derived measures, which are used to generate leading indicators information. Systems engineering metrics have been in use under document-based engineering (e.g., requirements productivity measured using page count), but model-based engineering approaches enhance the opportunity for collection of additional base measures (e.g., use cases completed). Model-based metrics has been a topic of interest for many years (e.g., Friedenthal et al., 2009). More recently, the maturation of systems modeling languages and model-based tools and techniques has prompted increased interest in what can be measured and how this might be automated. The downside is that generating metrics is much easier and therefore there is a risk of focusing on quantity of metrics, rather than the most useful set. With digital engineering transformation, research is needed to understand the most useful and impactful measures that can be composed, focusing on leading information.

Composability has been discussed in the systems literature as "the capability to select and assemble components in various combinations to satisfy specific user requirements meaningfully" (DMSO, 2004). A characteristic of composability is described as the ability to combine and recombine components into different systems for different purposes. Using this frame of reference for composability as applied to leading indicators, base measures can be thought of as components. In context of this research, the interest is composability of base measures extracted from a digital system model or digital process model that would, along with derived measures, comprise a leading indicator. We can think of this in three aspects, as follows:

Composing Base Measures into Leading Indicators. Leading indicators are composed from base measures, for example *requirements volatility* is based on requirements change base measures, as has been done in document-based engineering projects. Our investigation suggests that model-based toolsets enable automated collection and aggregation of base measures from a system model, providing the means for real-time generation of leading indicators – both currently used indicators and newly defined ones.

Composing Useful Sets of Leading Indicators. Eighteen leading indicators have been defined in prior research, and additional indicators of interest have already been described. Further, digital engineering is likely to influence several new leading indicators. An important aspect concerning composability is to determine what useful sets of leading indicators could be composed. These sets could then be displayed on a dashboard, where the decision maker could interpret the information by looking at the collected set of displayed indicators. Orlowski (2017) and other researchers have explored useful sets of indicators for a specific purpose, such as use in technical reviews. Further consideration needs to be given for model-centric programs, where more frequent reviews are likely to be used. Additionally, there is a need to explore what sets of leading indicators might be most useful for various types of program leaders.

Composing Aggregated Leading Indicators. Leading indicators are most useful when applied for predictive purpose to facilitate programmatic decisions and/or corrective actions. Requirements Trend indicators, for instance, are used to evaluate trends in the growth, change, completeness and correctness of the definition of system requirements. Traditionally, this indicator provides insight into the rate of maturity of system definition against the plan. Further, it characterizes stability and completeness of system requirements that could potentially impact design, production, operational utility, or support. In traditional document-based engineering practice, requirements are central objects that can be used for assessing maturity of system definition. In model-based engineering, however, there are many other potential base measures that can be obtained from the system model. With the advantages of model-based approaches, a leading indicator used to assess progress of system definition that uses base measures for requirements (e.g., requirements changes, requirements defects) would be limited to providing a leading indicator as requirement volatility. A potential leading indicator of interest with digital engineering is *model volatility*, based on change information of myriad objects in a system model. In a model-based environment, it may be possible to generate a model volatility leading indicator that uses base measures from the various model diagrams (e.g., in SysML requirements diagrams could be augmented with base measures from activity diagrams, use case diagrams, state machine diagrams, parametric diagrams, etc.). Model volatility may be a richer indicator for making decisions on review readiness than a requirements volatility indicator.

Impact of Model-based Toolsets

Model-based toolsets enhance the ability to extract base measure information (e.g., number of requirements changes) and produce derived measures (e.g., percent requirements modified). Collected or aggregated measurement data is then used to compose leading indicator information (e.g., requirements volatility, presenting a profile of added/modified/deleted requirements with projected trend lines (see Figure 1). To investigate this under the digital engineering context, an illustrative case was used to explore how digital engineering is expected to modify and/or enable four existing leading indicators the research identified as most likely to be implemented with direct use of model-based toolsets. These are requirements trends, interface trends, requirements verification trends and requirements validation trends.

Illustrative Case Discussion

The following example illustrative case involves systems engineering for new development of a self-driving fully autonomous vehicle meeting the SAE International Level 5 (“Full Driving Automation”) standards (see https://www.synopsys.com/automotive/autonomous-driving-levels.html). For more in-depth case examples, beyond the small illustrative case presented here, see Tepper (2010) for a more complete application of MBSE to Naval Ship Design using SysML and Dam (2019) for an application of LML and MBSE to a hypothetical establishment of a permanent manned base on the moon.

For the purposes of this case, Innoslate® is used to conduct a number of small-scale exercises. Innoslate® is an integrated MBSE software package that implements the open source LML Ontology, which is compact but comprehensive (Dam, 2019, p.5). The LML Ontology provides a guiding structure for investigating how information needed for leading indicators is represented in MBSE. The ontology is a compact, but comprehensive, organized, structured and customizable terminology for systems engineering from the earliest concept stage throughout the lifecycle to system disposal (Dam, 2019, p. 6, 10). Vaneman (2018) also reports that LML has sufficient constructs to be able to represent knowledge expressed or expressible in other modeling languages, such as SysML and DODAF.

Innoslate enforces the important principal of concordance, which facilitates single source of truth by requiring that a given piece of information in the systems engineering knowledge base will have the same meaning when viewed through different language or visualization lenses. The version of the Innoslate tools used in these experiments is implemented on a central cloud database. Current database and semantic web technology would also support more complex configurations with virtual integration of data stored in multiple physical locations provided that the semantics of the data can be made compatible.

The LML ontology knowledge framework is built on an Entity-Relationship-Attribute (ERA) data model. ERA was first introduced by Chen (1976) and is widely used today for conceptual modeling including by UML (https://www.omg.org/spec/UML/) and other methods. Entities represent things of interest (analogous to nouns in natural language). Relationships represent connections across entities. Attributes represent information about an entity or relationship. Standard entity attributes are defined in the LML Ontology along with standard relationships and the entity types they connect (LML Steering Committee, 2015). The LML Ontology provides definitions of typical relationship types for each combination of entity types (see Vaneman, 2018). Relationships are directional and matched with complementary relationships that go in the opposite direction (e.g., “performed by” and “performs” are complements). The primary entity structure of the LML Ontology is shown in Figure 2.

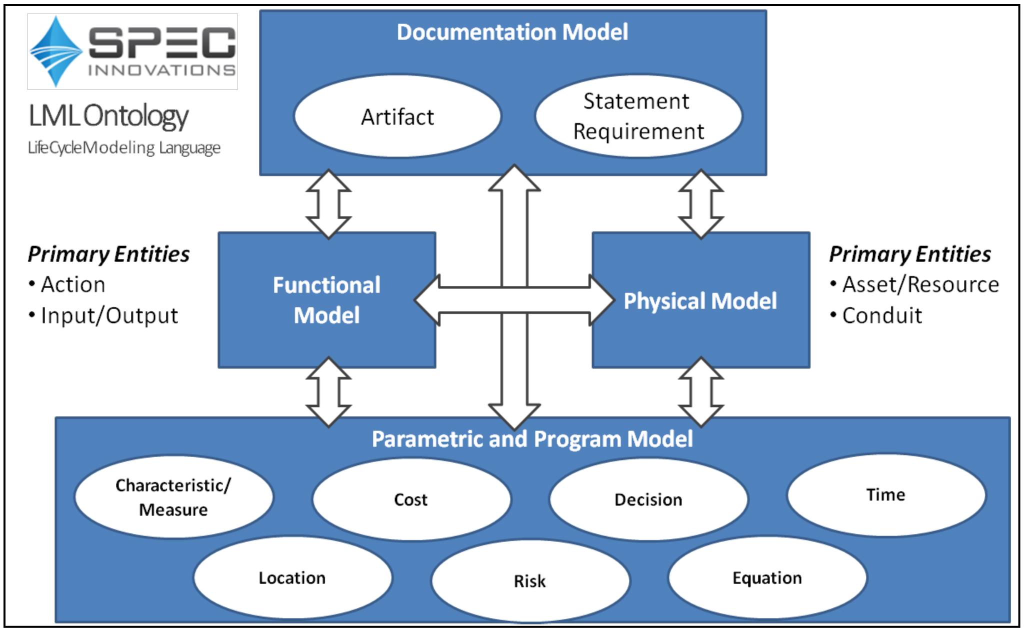
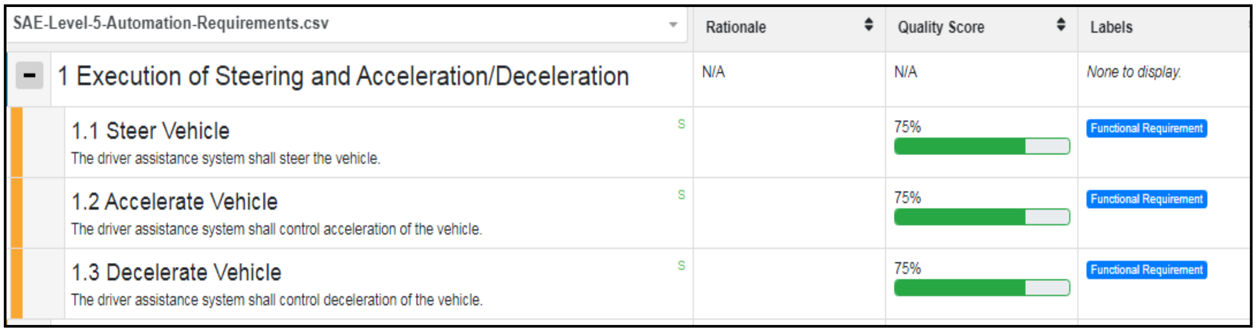


Figure 2. LML Ontology Primary Entity Classes (Dam, 2019)

The Documentation section at the top of the ontology diagram includes entity classes that are ancillary to the system model and intended for exchanging information with humans or other outside systems. The Artifact entity represents a document, spreadsheet, test plan, or other source of information that is referenced by, or generated into, the knowledgebase. A Statement entity specifies text usually drawn from an Artifact. A Requirement entity is a Statement that expresses a capability or characteristic of a system that must be present for the system to have value to users. The LML Requirement entity is similar to a Requirement block in SysML. Both are structures that encapsulate a textual requirement statement. All of the Documentation section entities can be loaded from the outside or generated for use by people or systems.

Figure 3 shows an excerpt from a sample LML requirement document. The artifact name appears at the top along with three of the statement entities below. The three visible statements are also requirement entities (1.1, 1.2, and 1.3). Figure 3 also shows examples of attributes for ID number, name, and description of each entity along with quality score, and labels attributes in the columns on the right side of the table.



SAE-Level-5-Automation-Requirements.csv	Rationale	Quality Score	Labels
1 Execution of Steering and Acceleration/Deceleration	N/A	N/A	None to display.
1.1 Steer Vehicle The driver assistance system shall steer the vehicle.		75%	Functional Requirement
1.2 Accelerate Vehicle The driver assistance system shall control acceleration of the vehicle.		75%	Functional Requirement
1.3 Decelerate Vehicle The driver assistance system shall control deceleration of the vehicle.		75%	Functional Requirement

Figure 3. Sample of a Structured Requirements Document

The Functional Model and Physical Model sections along with the Parametric and Program Model section have entity classes used for building and executing system models. The Action entity, on the left of the ontology diagram in Figure 2, is the primary building block for functional-behavioral models. The Asset entity, on the right, is the primary building block for physical models. Every entity has a “type” property, which allows many variants of Actions and Assets to be represented. For example, Actions may be described as Activity, Capability, Event, Function, Process, or Task. Assets may be described as Component, Entity, Service, Sub-system, or System as needed in a particular modeling context. Assets may represent human actors as well as physical components and software systems.

Another key basic concept in functional models is Input/Output (IO) which represents the flow of information or other resources in or out of an Action, including Item, Trigger, Information, Data, and Energy. The corresponding basic concept in a physical model is the Conduit, which might be implemented as a Data Bus, Interface, or Pipe. A resource in a functional model IO entity is transferred from one physical model Asset to another via a Connection or Conduit.

Relationships describe connections among entities. For example, decomposition is denoted with the decomposed by/decomposes relationships. Similarly, when an Asset is allocated to perform a functional model Action, a “performed by/performs” relationship is established. A functional model Input/Output entity may be allocated to a physical model Conduit via the “transferred by/transfers” relationship where the functional flow thereby becomes constrained by the properties of the physical device implementing the Conduit.

Figure 4 shows a top level Action Diagram for the functional model in the autonomous vehicle example. The diagram depicts three parallel functional flows with actions A.1, A.2, and A.3 performed by the User, Autonomous Vehicle and Environment physical assets respectively. The physical assets here are viewed functionally with physical properties captured elsewhere in the model as appropriate. Even though three assets are present, everything in the Action Diagram depicts functional requirements. The two IO entities describe the flow of Destination Location from A.1 to A.2 and Environmental Conditions from A.3 to A.2. As depicted, each of these IO flows is a trigger that enables A.2 Drive Vehicle to start.

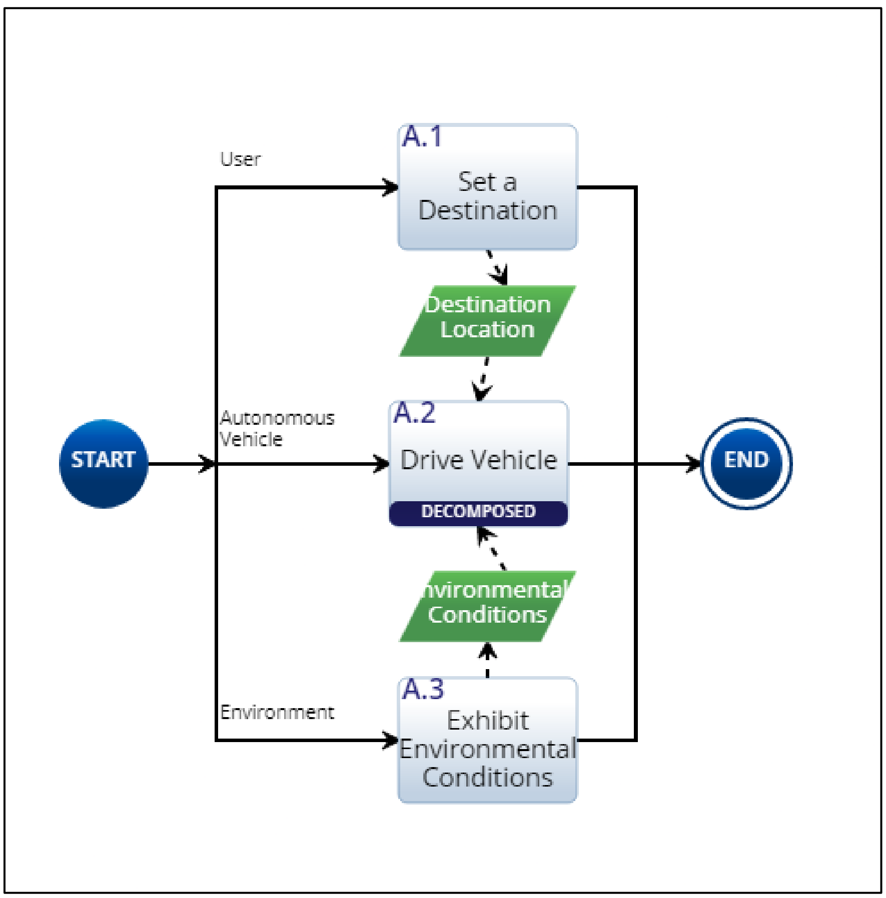


Figure 4. Top Level Action Diagram

The Figure 4 diagram also states that the A.2 Drive Vehicle action is decomposed. Figure 5 shows that decomposition. Figure 5 shows another three-way parallel functional flow inside a loop functional flow depicting the vehicle continuing to drive until the destination is reached.

On the top branch, the Sensors asset performs the A2.2 Monitor Environment action. On middle branch the Control System asset performs the A2.3 Calculate Waypoint and Obstacles action. On the bottom parallel branch, the Drive System performs the A2.4 Navigate Vehicle action. The two IOs on the left of Figure 5 are inherited as context from Figure 4 with connections adjusted to fit the decomposition. New IOs are introduced to describe sensor Camera Data and Lidar Data moving from A2.2 to A2.3 and for the calculated Waypoint and Obstacles moving from A2.3 to A2.4. These are all continuous parallel functional flows that continue until the A2.1 loop condition of Destination Reached is satisfied. The diagram also shows that A2.2 and A2.3 are further decomposed into other action diagrams not shown.

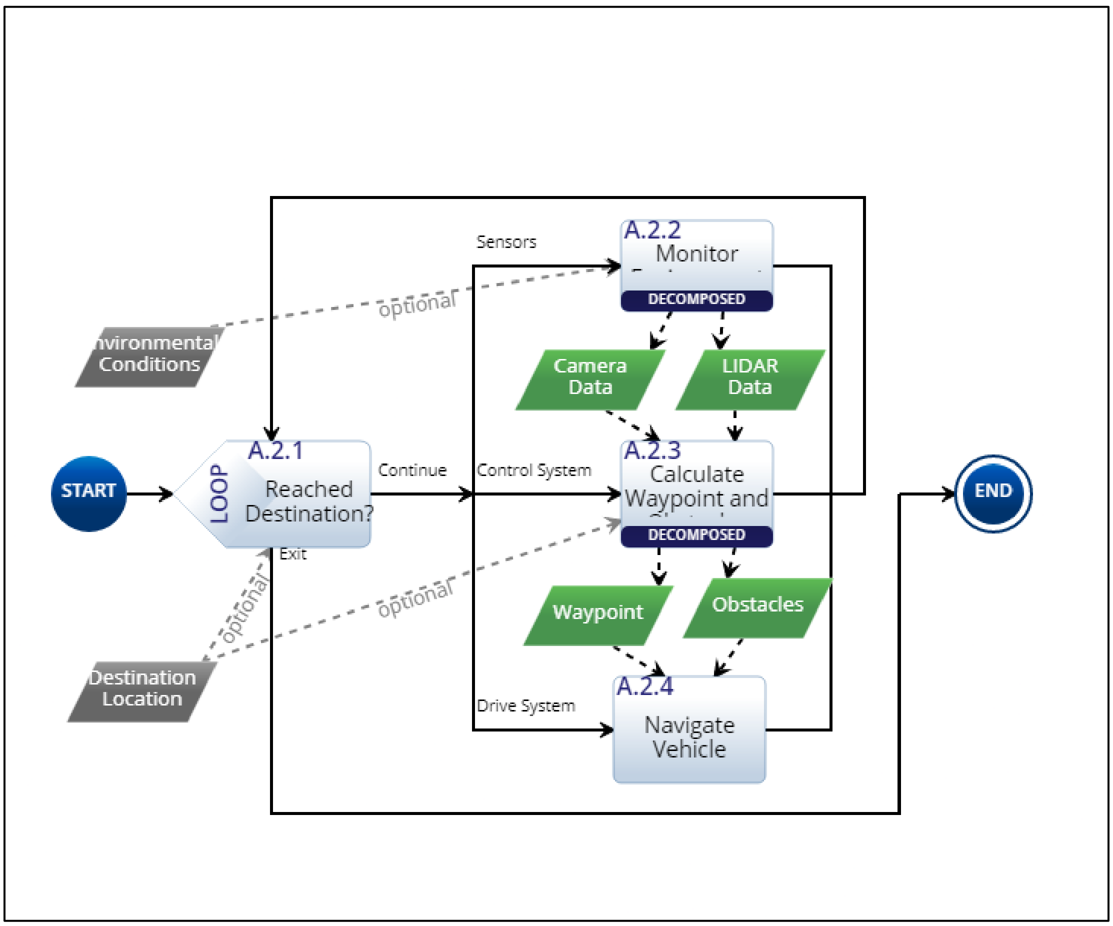


Figure 5. Drive Vehicle – Decomposition Action Diagram

For V&V, the relationships among model elements can be used to trace the derivation of the model. As mentioned above, the diagrams tell only part of the story with relationships filling in additional detail. Figure 6 is a “spider diagram” showing how some of the model elements trace back to the originating requirements for an SAE Level 5 Autonomous Vehicle. Similar diagrams can be used to review other connections across elements of the model, which can be helpful in establishing model validity.

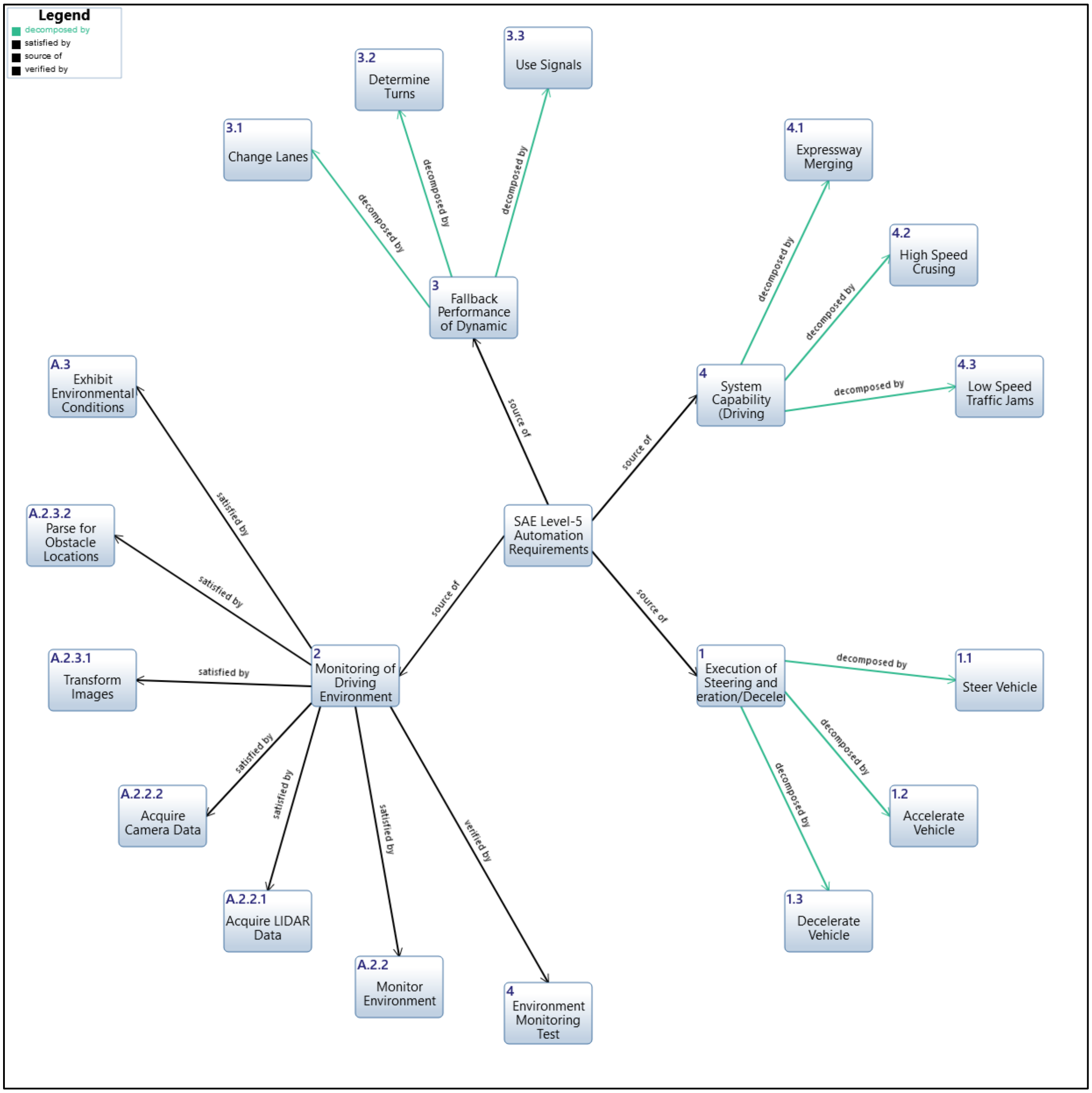


Figure 6. Relationship for Traceability to Originating Requirements

Since these models are executable, another form of validation is to run discrete event and other simulations and inspect the results to see if expectations of system behavior appear to be met. Figure 7 shows the output of a discrete event simulation run of the sample model. As performance characteristics of model elements are refined, the timing results will change. As the fidelity of the model improves, additional Monte Carlo and other simulations can be used to explore optimization of models.

Model diagrams, as well as spider and hierarchy diagrams and model simulators can be incorporated into dashboards for interactive exploration of the model and its implications.

Figure 7. Sample Validation with Discrete Event Simulation

Tracking the trends needed for the leading indicators requires taking snapshots of metrics values at intervals over time as the program proceeds. If integrated into an LML meta-model, this program management data would be stored as baselines or other objects in the database to facilitate integrative analysis with other program data.

Generating Leading Indicators from Digital System Model

Using the illustrative case, we examined four leading indicators (Table 2) that relate to aspects of requirements management. These four were found to be in the subset of the eighteen leading indicators that were most likely to be implemented with direct use of model-based toolsets based on availability of the base measure information in the system model (Rhodes, et al. 2021).

Table 2. Four leading indicators most likely to be implemented with direct use of model-based toolsets (Rhodes et al., 2021)

Leading Indicator	Insight Provided (source: 2010 guide)
Requirements Trends	Rate of maturity of the system definition against the plan. Additionally, characterizes the stability and completeness of the system requirements that could potentially impact design, production, operational utility, or support.
Interface Trends	Interface specification closure against plan. Lack of timely closure could pose adverse impact to system architecture, design, implementation and/or V&V any of which could pose technical, cost and schedule impact.
Requirements Validation Trends	Progress against plan in assuring customer requirements are valid and properly understood. Adverse trends would pose impacts to system design activity with corresponding impacts to technical, cost & schedule baselines and customer satisfaction.
Requirements Verification Trends	Progress against plan in verifying design meets the specified requirements. Adverse trends would indicate inadequate design and rework that could impact technical, cost and schedule baselines. Also, potential adverse operational effectiveness of the system.

In the current state of practice, requirements are typically collected and stored in a specialized requirements database, often using software (e.g., DOORS® or other similar packages suited to needs of the project/enterprise). These types of packages are generally interoperable with and/or loosely coupled to other systems engineering model-based toolsets. It is the assumption of this research team that the specific details of this will vary based on chosen model-based tools used.

Requirements Trends and Interface Trends

The metrics required for *Requirements Trends* and *Interface Trends* can be composed by counting explicit and implicit requirements identified in the Innoslate database. Explicit requirements are found in Requirements entities that contain natural language statements, which are (1) sourced from documents loaded into the system; (2) entered directly into the database by engineers; or (3) generated from other data and stored in the database.

Implicit requirements are derived from the functional and physical models developed by engineers during requirements analysis. Functional Requirements may be defined by Action entities and the flows, relationships, and properties that describe them. Innoslate also has a tool that converts Actions in a functional model into implied Assets and Conduits in a physical model.

Interface Requirements can be inferred from Conduit entities that connect Assets by transferring IO entities in the physical model. The technical characteristics of the endpoint Assets and the Conduit combine to specify the interface requirements. Performance Requirements often come from data related to Asset entities and connections. External interfaces would be represented by connecting a Conduit to an Asset that is outside the system boundary, as the User and Environment assets in Figure 4.

As requirements analysis progresses, the model and the requirements will grow deeper and broader. In traditional practice, the requirements are frozen in text and isolated from the models that engineers use for analysis. Whether explicit or implicit, a requirement in MBSE is linked by relationships to other elements of the model giving greater context to understanding the meaning of a requirement. For example, by running simulations on executable models, the engineer can identify whether a set of requirements has face validity or meets expectations. Spider charts and hierarchy charts can be used to visualize the structure of the model and the requirements.

As systems understanding develops, some information will be less refined than other information. For example, the value for a parameter in a requirement may be unknown (TBD) or estimated (TBR). LML Decision entities can be attached to the model to represent the both the uncertainty and the process for finding the missing information as well as defining assumptions. When the TBD/TBR is resolved, the updated Decision entities provide a record of how the value was obtained. A burn-down chart for progress on resolution of Decisions would also be informative as a leading indicator.

Base measures are used in composing leading indicators through manual, semi-automated or automated counts (Table 3. Change impacts may be estimated from the extracted information. In the case of digital engineering, model-based toolsets and supporting analysis software could provide more continuous calculation. Digital engineering enhances the "push" of leading indicators to the decision maker. An open area of inquiry concerns the frequency and specific points at which the measurement data should be updated in the dashboard display. Too frequent refresh of the information could be an issue, as requirements and interface changes are expected as part of normal system development. Over time, digital engineering would be expected to provide a rich set of historical data for use in establishing thresholds and projecting trends. A desired future enhancement is to have augmented intelligence capability to guide interpretation of the leading indicator.

Table 3. Extracting Base Measures for Requirements and Interface Trends

Base Measures	Traditional Engineering	Digital Engineering
Requirements	Manual or semi-automated count	Continuous automated count, on demand
Requirements TBDs/TBRs	Manual or semi-automated count	Continuous automated count, on demand
Requirements Changes	Manual or semi-automated count	Continuous automated count, on demand
Requirement Change Impact	Estimated impact	Calculated impact on demand
Interfaces	Manual or semi-automated count	Continuous automated count, on demand
Interface TBDs/TBRs	Manual or semi-automated count	Continuous automated count, on demand
Interface Changes	Manual or semi-automated count	Continuous automated count, on demand
Interface Change Impact	Estimated impact	Calculated impact on demand

Requirements Validation Trends and Requirements Verification Trends

Systems engineering best practice involves beginning requirements validation and verification early in the project as requirements are found and entered into the database. Table 4 shows the base measures, and how obtaining these varies under traditional versus digital engineering. As with the requirements and interface trends, while base measures can be available on demand, consideration needs to be given as to when this information is pushed to the decision maker. Historical information from model-centric programs can inform setting this interval. At some points in a program, the frequency of looking at these leading indicators varies with program lifecycle.

Table 4. Extracting Base Measures for Requirements Validation and Verification Trends

Base Measures	Traditional Engineering	Digital Engineering
Requirements	Manual or semi-automated count	Continuous automated count, on demand
Requirements Validated	Manual or semi-automated count	Continuous automated count, on demand
Requirements Verified	Manual or semi-automated count	Continuous automated count, on demand

At the early stage, Innoslate and some other toolsets offer a natural language tool for checking the quality of requirements statements against six of the eight standard criteria (clear, complete, consistent, design, traceable, verifiable but not correct and feasible). Another tool applies heuristics to evaluate models and requirements in more depth. A roll-up of these quality metrics could provide leaders with early insight on how well the requirements are progressing and whether problems are being left to later in the life cycle where they will be more difficult to resolve.

Innoslate also includes a Test Center where test plans and scenarios can be built for early or later use and VCRM Reports generated. Figure 8 shows a snapshot of part of a test plan and results with verification status. The leading indicators for requirements verification and requirements validation could be improved by adding measures for progress on developing test plans to complement the metric for successful completion of validation and verification testing.

Validation and verification also need to be considered holistically at the system level. The model used with simulation tools provides capability to predict behavior of the whole system or subsystems.

Level 5 Test Suite	Expected Result	Actual Result	Status	Status Roll-Up
1 Vehicle Steering Test 1. Approach obstacle course while driving the autonomous vehicle. 2. Come to a full stop and take your hands off the steering wheel. 3. Engage the Driver Assistance System (DAS). 4. Proceed into the obstacle course allowing the DAS to take over driving. 5. Continue to allow DAS to drive until the obstacle course has been completed or the vehicle goes out of bounds. 6. Dis-engage the DAS and resume driving manually.	The DAS will steer the vehicle through the obstacle course without hitting any obstacles or traveling out of bounds.	Meets all criteria.	Passed	Passed
2 Vehicle Acceleration Test 1. Approach the straight, fast test track while driving the autonomous vehicle. 2. Come to a full stop and take your foot off the gas pedal. 3. Engage the Driver Assistance System (DAS). 4. Proceed into the obstacle course allowing the DAS to take over driving. 5. Continue to allow DAS to accelerate the vehicle until the obstacle course has been completed or the vehicle goes out of bounds. 6. Dis-engage the DAS and resume driving manually	The DAS will accelerate the vehicle in the obstacle course without traveling out of bounds.	Meets all criteria.	Passed	Passed
3 Vehicle Deceleration Test 1. Approach the straight, fast test track while driving the autonomous vehicle. 2. Come to accelerate to 40 mph and take your foot off the gas pedal. 3. Engage the Driver Assistance System (DAS). 4. Proceed into the obstacle course allowing the DAS to take over driving. 5. Continue to allow DAS to decelerate the vehicle until either the system stops before the obstacle or it is deemed unsafe to continue. 6. Dis-engage the DAS and resume driving manually.	The DAS will decelerate the vehicle in the obstacle course without traveling out of bounds or hitting the obstacle.	Meets all criteria.	Passed	Passed
4 Environment Monitoring Test 1. Approach obstacle course while driving the autonomous vehicle. 2. Come to a full stop and take your hands off the steering wheel. 3. Engage the Driver Assistance System (DAS). 4. Proceed into the obstacle course allowing the DAS to take over driving. 5. Engage the dummy obstacles along the course track. 6. Continue to allow DAS to drive until the obstacle course has been completed or the vehicle goes out of bounds. 7. Dis-engage the DAS and resume driving manually.	The DAS will not hit any dummy obstacles.		In Progress	In Progress

Figure 8. V&V Sample Level 5 Test Suite

Summary

Model-based toolsets used in digital engineering, as described in the illustrative case, are now a source of base measures for selected leading indicators. With the progression of digital engineering, it is anticipated that many of the leading indicators will be available in models, rather than documents or spreadsheets. Four leading indicators most likely to be generated from model information are described for an illustrative case using a selected toolset and ontology. There are multiple ontologies and numerous toolsets in use in digital engineering, so an area of further research would be run experiments to understand available of base measures and calculated derived measures. Further investigation is needed to understand how other fields of the leading indicator specification (Table 1) related to the information in models and the digital engineering environment. For example, units of measurement may be different of digital engineering versus what has been used in document-based engineering. Interpretation of leading indicators could in the future involve some augmented intelligence capability. The impact of model-based toolsets as suggested by our exploratory research is enhanced capability to provide selected leading indicators, and the possibility of new leading indicators. Models would enable more timely insight into requirements volatility, for example. Model-based toolsets also offer the opportunity to compose a model volatility leading indicator that reflects changes in full set of diagrams being used. The illustrative example was based on Lifecycle Modeling Language, LML Many model-based toolsets use SysML, with nine types of diagrams: Block definition diagram; Internal block diagram; Package diagram; Use case diagram; Requirement Diagram; Activity diagram; Sequence diagram; State machine diagram; and Parametric diagram.

-

Visualization and Interactivity

More complex leading indicators are likely in the digital engineering context, resulting from increased information, synthesis, and composability of measurement data. Accordingly, decision-makers will face challenges in comprehending the information, including a need to understand the underlying assumptions and uncertainties in the constituent data elements. Investigating the approach to display such leading indicators is an important area of inquiry. Measurement dashboards have been used extensively for decades, typically providing static display of information. Visual analytics and interactive technologies provide the opportunity to create dynamic dashboards that would enable a decision-maker to be able to interact with the data. Such interaction allows for visual simplification while providing more transparency to the underlying data, as well as enabling the development of understanding and trust in the information.

Visual Analytics

Visual analytics is fundamentally about collaboration between a human and a computer using visualization, data analytics, and human-in-the-loop interaction. More than just visualization tools, visual analytics aims to take advantage of a human's ability to discover patterns and drive inquiry to make sense of data. Thomas (2007) defined visual analytics as "the science of analytical reasoning facilitated by interactive visual interfaces" that "provides the last 12 inches between the masses of information and the human mind to make decisions." As engineering becomes increasingly model-based, the available information to draw on to generate measures of effectiveness is vast and complex. It is foreseeable that decision-makers could be presented with large amounts of data that would be cognitively challenging to comprehend and difficult to find patterns that could be used to judge the effectiveness of the engineering on an ongoing program. For this reason, best practices knowledge as well as recent advancements in visual analytics may offer significant support in processing and displaying measurement data for decisions.

Vitiello and Kalawsky (2012) state the “guiding process in visual analytics is a synergy between interactive visualization and automated analysis of the data.” Such human-computer teaming to discover insights is discussed as an approach that builds upon a visual analytic based workflow with the notion of sensemaking. The authors describe using visual analytics to support systems thinking to make sense of complex systems interactions and interrelationships, enabling rapid modeling of the systems of interest for systems engineering design and analysis processes. The visual analytic based sensemaking framework they describe aims toward providing the means to rapidly gain valuable insights into the data. A key strength of modern visual analytic approaches is that is makes insights accessible with less effort, as well enabling the ability to discover new insights beyond the intention of a static visualization (Yalcin, Elmqvist, and Bederson 2018).

Interactive Dashboards

Classically, information dashboards (example, circa 2008, shown in Figure 9) provide a high level summary of key information, accessible to a user at a glance, in order to support targeted decision making (Few 2013). Applications of such dashboards necessarily simplifies the underlying information as a means to an end of their purposes, and must account for challenges that hinder their effective use (Alhamadi 2020).

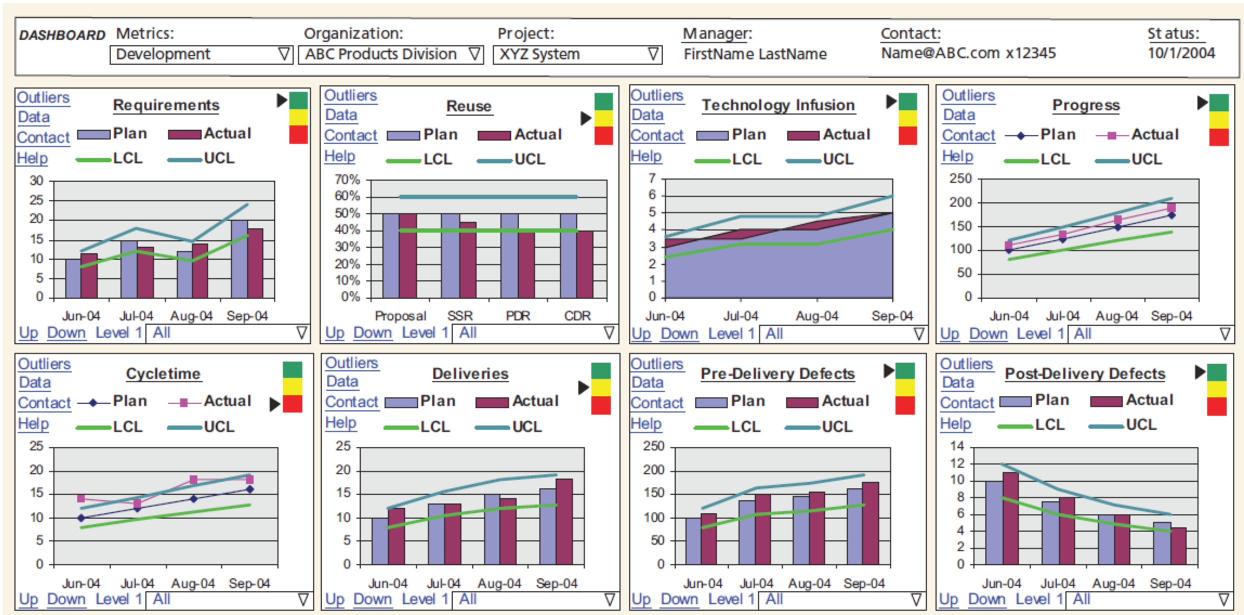


Figure 9. Example metrics dashboard (circa 2008) from Selby (2009)

For situations where multi-level data is needed, especially for details on demand, static dashboards are insufficient. Instead, leveraging techniques from visual analytic workflows, interactive dashboards enable both the benefit of apparent simiplfication of complex data, and context-dependent customizations of data representation. Strategic decision-making support in dashboard design has been shown to embody both interactivity as well as updatable, multipage visual features (Sarikaya et al. 2019). Strategic decision-making support dashboards focus on enabling drill-down as well as top-level synthesis of information, whereas static orgnaizational dashboads seek to improve awareness within an organization of the state of affairs (Figure 10). Systems engineering applications necessarily reside in both of these spaces, but moreso the former, as programs progress over time and require interventions from such information. Model-centric environments naturally lend themselves to providing time-varying data and would benefit from the promise of interactive dashboards.

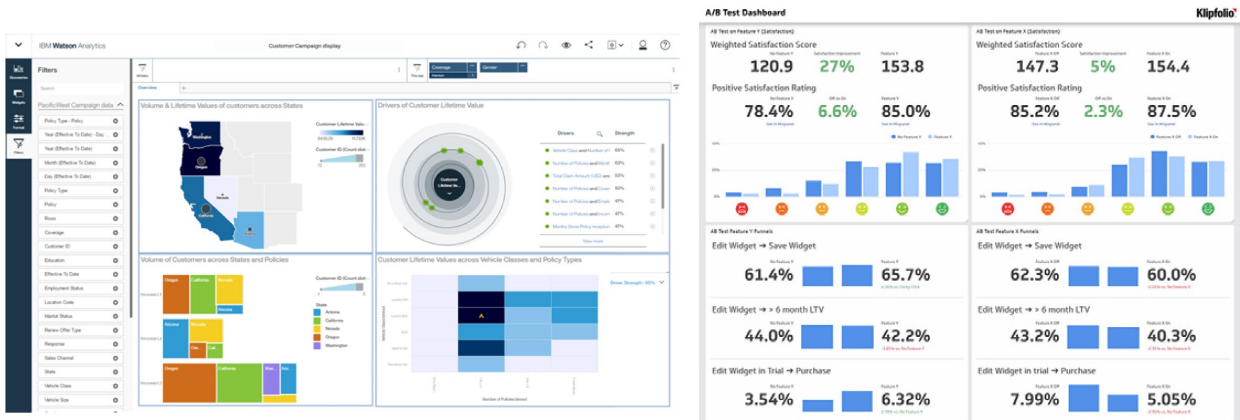


Figure 10. Example strategic decision-making dashboard (left) and static organizational dashboard (right) (circa 2019) (Sarikaya et al. 2019)

Systems engineers, managers, and government sponsors all rely on creative work products of systems engineering and all need to glean an appropriate level of understanding of the work as it progresses. The mean time for a warfighting system to move from well-defined concept to initial operating capability can be substantial, regularly averaging six to seven years (Dwyer, 2020). Leading indicators can help stakeholders see how a project or program is progressing throughout the lifecycle and whether it is on target to deliver what is needed when it is needed at an affordable cost. The complexity of understanding the status and trajectory of a program is high and larger than any one person can hold in one's head. Systems engineering methods, languages, and models are intended to leverage visualizations, structure, and compuational representations to make the task manageable for all the humans who must be involved. Model-based systems engineering incorporates all of those features and authors have previously pointed out how structured representations can improve MBSE accessibility. Sindiy et al. (2013) demonstrates how clean visual representations can help in making MBSE models accessible. Dam (2019) argues that, in addition to visualizations, modeling language and ontology matters, since a representation that is inherently fragmented and lacks a well-structured ontology will be less cognitively accessible to users. As a program progresses through the lifecycle, an interactive dashboard could provide most recent information, while also providing the transparency for the user to pull up prior year data. It follows that the quality of the measurement information relates to the quality of the systems model.

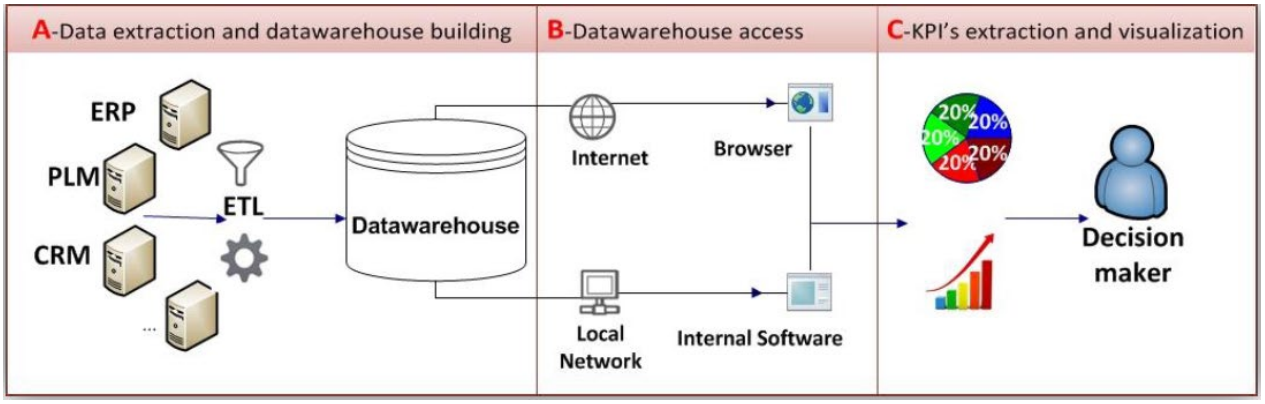


Figure 11. Example dataflow from data warehouse to KPI visualization (Fradi et al. 2017)

Dashboards are often created as views into program data that has been extracted and loaded into a data warehouse. This warehouse then provides a centralized location from which heterogenous access is possible, including targeted dashboards that roll up indicators for stakeholders, including key decision makers (Figure 11). Dam (2020) proposes that stakeholders should be given controlled direct access to MBSE models to improve the speed and depth of understanding in system reviews. He also argues that prime contractors and subcontractors can achieve better coordination by using MBSE models as a vehicle for communication about the system that is being created, program progress, and how organizations with different roles and incentives will fit together to deliver the capability needed to meet customer objectives. As such, a program may want to consider giving controlled access to interactive dashboards respective to role/authorities.

The use of interactive dashboards has shown promise in supporting management-level decision making, including those relying upon indicator evolution over time (Selby 2009). Leading indicator project data can be presented in a compact form with tools for organizing data, drilling into the underlying data, and connecting data to analytic tools and models. Orlowski (2017) and Orlowski et al. (2015) propose a framework for guiding leading indicator development and usage to support technical reviews and audits.

The continued advancement of underlying visualization and data pipeline technologies of interactive dashboards has made their use even more accessible.

Recent work by Thiruvathukal et al. (2018) shows the potential for using open source software repositories in the development of software metrics dashboards. Yalcin, Elmqvist and Bederson (2018) demonstrate a semi-automated dashboard development environment allowing for easily customized interactive dashboards that accelerate insights for data analysis novices. On the other side of the experience spectrum, Nadj (2020) addresses how interactive dashboards help managers in gaining and maintaining situational awareness to understand the context of metrics. This deeper understanding of context, along with the metrics, enable decision makers to establish more trust of the data, and therefore making them more effective.

Story and Treude (2019) cite the goal of dashboards is to transform the raw data into consumable information. These authors describe possible risks of dashboards in context of software engineering projects and software developer productivity. Table 5 suggests how interactive dashboards for systems engineering leading indicators in model-centric programs mitigate several of the risks.

Table 5. Risks Mitigated by Interactive Dashboard Capability in Model-Centric Programs

Risks of Using Dashboards (Story & Treude, 2019)	***Potential mitigation of risks through interactive dashboards***
Dashboards favor numbers over text	Interactive capability could enable a dashboard user to access associated underlying text information in models
Dashboards might not display relevant context	As cited by Story & Treude, interactive dashboards allow users to drill down into more complete information that provides better understanding of context
Dashboards often don't explain	Interactive dashboards could be connected to supporting explanatory information in the underlying model
Dashboards can only be as good as the underlying data	Static dashboards have latency in information. A dashboard connected to the model would enable real-time information to be displayed.
Dashboards can only display data that has been tracked somewhere	Interactive dashboards have potential to access data that is extracted/computed from the model itself
Performance-related data on dashboards can easily be misinterpreted as productivity data	As part of the dashboard design, leading indicators have associated interpretation guidance that could be included as displayed information

Summary

Digital engineering will extend use of models into all facets of engineering of systems throughout the lifecycle. This will offer extensive opportunities for use of newer technologies that support the visualization and interaction with program information. Knowledge from visual analytics and emerging technology for interactive dashboards provide a foundation for further research in the digital engineering community. A discussion of recent research follows.

THIS PAGE LEFT INTENTIONALLY BLANK

Discussion and Future Directions

This research has accomplished initial investigation for how ontologies and model-based toolsets enable the collection and composability of base measures to generate leading indicators. An initial investigation of interactive dashboards suggests that program leaders will be able to make improved and accelerated decisions using leading indicators if these are integrated with model-based environments to provide on-demand trend information. Implications identified in this research, including potential new leading indicators, can inform ongoing efforts in the systems community to define new or revised metrics for digital engineering programs and enterprises.

Benefits, Impacts and Risks

Effectiveness of systems engineering has been shown to have a positive relationship to the performance outcomes of projects and programs (Elm J. , Goldenson, El Eman, Donatelli, & Neisa, 2008); (Elm & Goldenson, 2013). A study by Orlowski (2017) shows the use of systems engineering measurement on a project as positively impacting the performance of the project. He found that 59% of higher performance programs in his study had higher use of systems engineering leading indicators (Orlowski C. T., 2017).

A reexamination of existing leading indicators in the context of digital engineering includes understanding where enabling infrastructure, analytic approaches, frameworks and constructed dashboards are beneficial. Understanding the impacts as well as the risks is important. Some promising outcomes are emerging within the systems engineering community. Orlowski et al. (2015) state that “premature transition through key decision gates is likely to lead to cost and schedule overruns” but that program risks can be monitored through systems engineering measurements. These authors propose a framework for implementing systems engineering leading indicators for technical reviews and audits. Dashboards for each technical review and audit may be beneficial. These authors state that “an aggregate of the leading indicators will assist with assessing the risks with exiting decision milestone” and that “leveraging leading indicators to update the risk assessment will strengthen the end confidence around execution". (Orlowski, Blessner, Blackburn, & Olson, 2015), (Orlowski C. T., 2017).

Ongoing work by other researchers and practitioners is beginning to identify model-based systems engineering artifacts used throughout the lifecycle. An excellent example of this is work by Parrot and Weiland at NASA regarding using MBSE to provide artifacts for NASA project life-cycle and technical reviews (Parrot & Weiland, 2017). According to these authors, "…the use of MBSE can reduce the schedule impact usually experienced for review preparation, as in many cases the review products can be auto-generated directly from the system model". Parrot and Weiland believe leading indicators that might exist within a model (e.g., number of requirement changes, verification burndown status, etc.) could be populated within the model using parametrics or by simple scripting techniques, while other indicators (e.g., drawing percent released) may need scripting or manual entry of the information. (Parrot & Weiland, 2017).

While in the future leading indicators of engineering effectiveness may be available on-demand through interactive dashboards, at present the existing eighteen systems engineering leading indicators will necessitate some manual effort to generate and track, and increasingly software tools include automation and some techniques (e.g., natural language processing) to augment the human decision-making. In the near term, modeling toolsets can aid in generating the base information for generating leading indicators. This availability of measurement information that is more easily collected (and supported with automation) will help to mitigate the burden of collection that presently exists, and composability research will support enhanced leading indicators that have potential to reduce program risk.

Limitations and Future Research

The research largely draws from the defense systems engineering community and literature from that sector. Future research can investigate additional sectors, as well as related disciplines. Expert knowledge was gathered though available workshops and from prior leading indicator project participants in the early phases. The limitations imposed by the COVID-19 pandemic, especially on workshops and conference events other than virtual, resulted in reduced opportunities for access to the community of interest. Planned group discussions were replaced with individual interviews and discussions, which resulted in reduced iteration and feedback opportunities.

This research has included some experimentation with extraction of base measures based on a single systems engineering toolset (selection of toolset was based on ease of use and availability to research team). Future research is needed to investigate extraction and composition of measurement information across the available model-based toolsets. Additionally, variation in implementing digital engineering practice needs to be examined concerning this objective. For example, some of the existing leading indicators depend on disciplined management processes for approval of key program artifacts (e.g., requirements, change orders, interfaces, and test plans). While these processes are not part of the system being developed, they can be modeled and/or tracked through model-based toolsets. This would enable measuring aspects of process compliance. Dam (2019) gives examples of how software could be used in support of measuring management processes. Future research can investigate how technical and management related base measures could be composed into new indicators in such an implementation.

Another recent active vein of research is the translation of model structures into traditional textual requirements statements. London, B. and Miotto, P. (2014) demonstrate the generation of textual requirements from a SysML model. Salado and Nilchiani (2014) and Salado, Nilchiani and Verma (2017) investigate different foundational aspects of the semantic content of models. Salado and Wach (2019a) examine automatic generation of contractual requirements from MBSE artifacts. Salado and Wach (2019b) address improving the quality of requirements expressed in SysML. Other research is looking at models used to support developing practices of agile systems engineering (Wanderley, et al., 2014). Methods for automatically translating system models to equivalent text is important for validation of models by experts who are more facile with natural language than model artifacts. Since leading indicators have developed in the context of textual requirements, study of semantic equivalency of model artifacts and text may help in future research on model-based leading indicators as well.

Model-centric programs have the opportunity to leverage leading-edge technologies in the collection, composition and display of measurement data, as well as enable better decisions to be made throughout the program lifespan. Two aspects for future investigation are techniques emerging from visual analytics and from data science. Model-based acquisition programs will be faced with dealing with four cited challenges of big data: *volume*: the magnitude of digital engineering information; *variety*: existence of digitized assets (e.g., drawings, etc.) that are not in themselves models; *velocity*: rapid information flow (e.g., operational digital twins sending information back to the digital system model); and *veracity*: uncertainty inherent in model data (e.g., artificial data from simulations, incomplete data, subjectivity in models).

Future research is needed to explore new leading indicators (e.g., model volatility) that are made tractable through model-based toolsets. Automation and augmented intelligence offer opportunities to support the composability of measurement data to provide on-demand leading indicators that exhibit reduced latency of the information. Further, with sufficient context information on lifecycle activities, future capability might include leading indicators that are 'pushed' to the decision makers rather than 'pulled' when a query is made.

Design of interactive dashboards with connectivity to model-based environments is a rich research opportunity. Many new technologies in the commercial market are becoming available to support these dashboards. Storey and Treude (2019) state their expectation that "artificial intelligence, natural language processing, and software bots will impact dashboard design". They suggest AI and NLP could also enable gathering insights on how and when dashboards are used. The design of dashboards needs to be investigated regarding the set of leading indicators most useful to display for a given role and specific decision points. Empirical research studies are needed, as well as experiment-based studies to observe actual behaviors of interactive dashboard users.

Future research is needed to further elicit ideas from the systems community on enterprise-level indicators. Desirable research would be to conduct industry case studies to learn from digital engineering early adopters concerning what metrics and leading indicators they have implemented, as well as novel approaches that have been developed. This includes extraction and composition of leading indicators, the implementation of measurement dashboards, and the specific practices used in making decisions with measurement information. Automation and augmented intelligence are two topics for future exploration. Practical limitations of this research project did not allow for extensive exploration of these topics, and further, these are rapidly evolving areas.

The success of the leading indicators initiative has been enabled through an approach rooted in the foundational work of more than 20 collaborating organizations, with engagement of government, industry and academic stakeholders. Continued and future work will build on the foundational and emerging knowledge to achieve future research goals, and engage with other stakeholders to validate outcomes and transition research to practice.

THIS PAGE LEFT INTENTIONALLY BLANK

References

Alhamadi, Mohammed. (2020) Challenges, Strategies and Adaptations on Interactive Dashboards. UMAP '20, July 14-17, 2020, Genoa, Italy. pp. 368-371.

Bone, M. A., Blackburn, M. R., Rhodes, D. H., Cohen, D. N., & Guerrero, J. A. (2019). Transforming systems engineering through digital engineering. *The Journal of Defense Modeling and Simulation*, 16(4), 339-355.

DoD. (2018, June). *Digital Engineering Strategy,* Office of the Deputy Assistant Secretary of Defense for Systems Engineering.

Dam, S. (2019) *Real MBSE: Model-Based Systems Engineering (MBSE) using LML and Innoslate*.

Dam, S. (2020) How to perform a model-based review (MBR). Whitepaper retrieved April 4, 2021 from https://www.innoslate.com/resource/model-based-review-whitepaper/

Defense Modeling and Simulation Office (DMSO). 2004. Composable Mission Space Environments. Retrieved from https://www.dmso.mil/public/warfighter/cmse.

Dwyer, Morgan (2020). Understanding Acquisition Speed for the Defense Department's Costliest and Most Complex Programs. CSIS Report CSIS-AM-20-159. Retrieved April 4, 2021, from https://dair.nps.edu/handle/123456789/4288

Elm, J. & Goldenson, D. (2013). Quantifying the effectiveness of systems engineering. *IEEE Systems Conference (SysCon)* (pp. 6-13).

Elm, J., Goldenson, D., El Eman, K., Donatelli, N., & Neisa, A. (2008). *A survey of systems engineering effectiveness survey.* Carnegie Mellon University and National Defense Industrial Association.

Gerst, K. J. & Rhodes, D. H. (2010). Strenthening systems engineering leading indicators for human systems integration considerations - Insight from the Practitioner Community. *Proceedings of Conference on Systems Engineering Research.*

Gilbert, D., Yearworth, M., Oliver, L. (2014). Systems approach to the development and application of technical metrics to systems engineering projects, *Conference on Sysems Engineering Research, Procedia Computer Science*,Volume 28, Pages 71-80.

LML Steering Committee. (2015). Lifecycle Modeling Language (LML) specification, Retrieved April 1, 2021 from https://lifecyclemodeling.org/wp-content/uploads/2021/01/LML_Specification_1_1.pdf

London, B. and Miotto, P. (2014). Model-based requirement generation. 2014 IEEE Aerospace Conference, pp. 1-10, doi: 10.1109/AERO.2014.6836450.

McDermott, T.A., Hutchinson, N., Clifford, M., Van Aken, E., Slado, A., & Henderson, K. (2020) Benchmarking the benefits and current maturity of model-based systems engineering across the enterprise. *Systems Engineering Research Center (SERC)* Technical Report SERC-2020-SR-001

Micoun, P., Paper, P., Fabre, L., Razafimahefa, T., Becquet, R. (2018). Property model methodology: A landing gear operational use case. INCOSE International Symposium, hal-01829910

Montgomery, P., & Carlson, R. (2010). *Systems engineering applied leading indicators: enabling assessment of acquisition technical performance.* Naval Postgraduate School, Graduate School of Business and Public Policy, Monterey.

OMG, Object Management Group Standards Organization, https://www.omg.org/hot-topics/syse.htm, Retrieved July 2021.

Orlowski, C. T. (2017). *A framework for implementing systems engineering measures at technical reviews and audits*. Retrieved January 10, 2020, from https://pqdtopen.proquest.com/doc/1878241332.html?FMT=ABS

Orlowski, C., Blessner, P., Blackburn, T., Olson, B. A. (2015). A framework for implementing systems engineering leading indicators for technical reviews and audits. *Procedia Computer Science: Complex Adaptive Systems Conference, 61*, 293-300.

Nadj, M., Maedche, A., & Schieder, C. (2020). The effect of interactive analytical dashboard features on situation awareness and task performance. *Decision support systems, 135*, 113322. https://doi.org/10.1016/j.dss.2020.113322

Parrot, E. & Weiland, K. (2017). Using model-based systems engineering to provide artifacts for nasa project life-cycle and technical reviews. *AIAA Space and Astronautics Forum and Exposition*, (p. 5299).

PSM. (2020). *Practical Software & Systems Measurement*. Retrieved from Practical Software & Systems Measurement: http://www.psmsc.com

Rhodes, D.H. (2021). Adapting Systems Engineering Leading Indicators to the Digital Engineering & Management Paradigm. *18th NPS Annual Acquistion Research Symposium.*

Rhodes, D.H. (2020). Investigation of leading indicators for systems engineering effectiveness in model-centric programs. *17th NPS Annual Acquistion Research Symposium. https://dair.nps.edu/bitstream/123456789/4209/1/SYM-AM-20-060.pdf*

Rhodes, D. H., Valerdi, R., & Roedler, G. J. (2009). Systems engineering leading indicators for assessing program and technical effectiveness. *Systems Engineering, 12*(1), 21-35.

Rhodes, D. H., Valerdi, R., Gerst, K. J., & Ross, A. M. (2009). Extending Leading indicators for human systems integration effectiveness. *7th Conference on Systems Engineering Research*

Roedler, G. J. & Rhodes, D. H. (2007). *Systems engineering leading indicators guide, Version 1.0.* MIT, INCOSE, PSM

Roedler, G. J., Rhodes, D. H., Schimmoler, H., & Jones, C. (2010). *Systems engineering leading indicators guide, Version 2.* MIT, INCOSE, PSM. INCOSE-TP-2005-001-03

Salado, A., Nilchiani, R. (2014). Categorization Model of Requirements Based on Max-Neef's Model of Human Needs. Syst. Eng. 17, 348–360.

Salado, A., Nilchiani, R., Verma, D. (2017) A contribution to the scientific foundations of systems engineering: Solution spaces and requirements. J. Syst. Sci. Syst. Eng. 26, 549–589.Salado, A. & Wach, P. (2019a). Automatic Generation of Contractual Requirements From MBSE Artifacts. Naval Postgraduate School Report SYM-AM-19-042. https://dair.nps.edu/handle/123456789/1729

Salado, A. & Wach, P.. (2019b). Constructing True Model-Based Requirements in SysML. Systems. 7. 10.3390/systems7020019.

Selby, R. (2009). Analytics-driven dashboards enable leading indicators for requirements and designs of large-scale systems. *IEEE Software 26*. 41 - 49. 10.1109/MS.2009.4

Sindiy, O., Litomisky, K.,Davidoff, S., & Dekens, F. (2013). Introduction to information visualization (infovis) techniques for model-based systems engineering. *Procedia Computer Science. 16*. 49–58. 10.1016/j.procs.2013.01.006

Shirley, E. (2016). A*pplication of system engineering leading indicators to scrum agile projects*, Master's thesis, Air Force Institute of Technology.

Storey MA., Treude C. (2019) Software Engineering Dashboards: Types, Risks, and Future. In: Sadowski C., Zimmermann T. (eds) Rethinking Productivity in Software Engineering. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4221-6_16

Systems Engineering Research Center (SERC). (2020). *SERC Website*. Retrieved from http://sercuarc.org

Tepper, Nadia (2010). Exploring the use of Model-Based Systems Engineering (MBSE) to develop Systems Architectures in Naval Ship Design. MIT unpublished SM thesis. https://dspace.mit.edu/handle/1721.1/61910

Thiruvathukal, G. K., Hayward, N. J., Läufer, K. (2018). Metrics dashboard: a hosted platform for software quality metrics. *arXiv preprint arXiv:1804.02053*.

Thomas, J. (2007). Director, USDHS National Visualization and Analytics Center, *Visual analytics: an agenda in response to dhs mission needs*.

Vaneman, W.K., Ph.D. (2018), Evolving Model-Based Systems Engineering Ontologies and Structures. INCOSE International Symposium, 28: 1027-1036. https://doi.org/10.1002/j.2334-5837.2018.00531.x

Vitiello, P. & Kalawsky, R.S. (2012). Visual analytics: A sensemaking framework for systems thinking in systems engineering, *Proceedings IEEE International Systems Conference* .

Wanderley, F., Silva, A., Araujo, J., Silveira, D.S. (2014) SnapMind: A framework to support consistency and validation of model-based requirements in agile development. In Proc.2014 IEEE 4th International Model-Driven Requirements EngineeringWorkshop (MoDRE), Karlskrona, Sweden, 25 August 2014, pp. 47–56

Zheng, L., Baron, C., Esteban, P., Xue, R., Zhang, Q., & Yang, S. (2017). Considering the systems engineering leading indicators to improve project performance measurement. IFAC-PapersOnLine, *50*(1), pp. 13970-13975.

Zheng, L., Baron, C., Esteban, P., Xue, R., Zhang, Q., & Yang, S. (2019). Using leading indicators to improve project performance measurement. *Journal of Systems Science and Systems Engineering, 28*(5), pp. 529-554.

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

www.acquisitionresearch.net