SYM-AM-21-047

Excerpt from the Proceedings of the Eighteenth Annual Acquisition Research Symposium

Adapting Systems Engineering Leading Indicators to the Digital Engineering & Management Paradigm

May 11–13, 2021

Published: May 10, 2021

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School

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).

Adapting Systems Engineering Leading Indicators to the Digital Engineering & Management Paradigm

Donna H. Rhodes—is a Principal Research Scientist at Massachusetts Institute of Technology Sociotechnical Systems Research Center, and director of the Systems Engineering Advancement Research Initiative (SEAri). She conducts research on innovative approaches and methods for architecting complex systems and enterprises, digital transformation, model curation, and human–model interaction. Previously, she held senior leadership positions at IBM, Lockheed Martin, and Lucent. Rhodes is a Past President and Fellow of the International Council on Systems Engineering (INCOSE), and INCOSE Founders Award recipient. She received her PhD in systems science from T. J. Watson School of Engineering at Binghamton University. [rhodes@mit.edu]

Eric Rebentisch—is a Research Associate at MIT's Sociotechnical Systems Research Center. His research focuses on implementation of digital engineering and model-based systems engineering in enterprises. He is Editor in Chief of *Integrating Program Management and Systems Engineering*, and teaches systems engineering in the MIT System Design and Management (SDM) program. Rebentisch received a doctorate in management of technological innovation from Sloan School of Management at the Massachusetts Institute of Technology, a master's degree in organizational behavior from Brigham Young University, and a Bachelor of Science degree in aerospace engineering from California State Polytechnic University, Pomona. [erebenti@mit.edu]

Allen Moulton—is a Research Scientist at MIT's Sociotechnical Systems Research Center. His research has addressed issues of capability-based defense acquisition requirements planning and cybersecurity in DoD; critical success factors in achieving economic benefits from information quality across enterprises; and applying abductive logic programming for automated reasoning to resolve semantic differences in the interchange of information among autonomous, heterogeneous data sources and receivers. Prior work included applications of information technology to substantive foreign policy analysis. He is co-author of *Managing International Conflict: from Theory to Policy* and the CASCON software for analyzing international conflict. [amoulton@mit.edu]

Abstract

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. This paper discusses continuing research on the adaptation of existing systems engineering leading indicators (developed under the assumptions of document-based engineering) for digital (model-based) engineering. Model-based implications identified in the research are discussed in support of the use of existing leading indicators in digital engineering programs. An illustrative example describes how measurement data can be extracted from a digital system model and composed into indicators. The importance of visualization and interactivity is discussed, especially the potential role of visual analytics and interactive dashboards. Several recommendations for future research are proposed based on interim research findings.

Introduction

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 et al., 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., economics, 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 then 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.

Existing leading indicators were developed under the document-based engineering approach. The introduction of digital engineering practices can have a potentially radical or disruptive impact on the processes, tools, and time lines of engineering programs. Rapidly accelerating analytical and design capabilities will 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 indicators for digital engineering and management practice. Additionally, the art of the possible needs to be explored including how digital system model information could be used to extract and compose base measures into indicators. Investigation is also needed to understand how newer sciences and technologies—such as data science, visual analytics, and interactive dashboards—could better inform timely leadership decisions in model-centric programs.

Background

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 13 leading indicators defined using measurement specifications. This work was subsequently evolved through collaboration from organizations and individuals across the systems engineering community with over 20 organizations as contributors. The result was a second version of the guide (Roedler et al., 2010), with five new leading indicators and several appendices added. Additional studies and papers have also been published by various authors (Elm et al., 2008; Elm & Goldenson, 2013; Gerst & Rhodes, 2010; Gilbert et al., 2014; Knorr, 2012; Montgomery & Carlson, 2010; Orlowski, 2017; Orlowski et al., 2015; Rhodes et al., 2009; Shirley, 2016; Zheng et al., 2017; and Zheng et al., 2019).

Prior work on leading indicators was done under the assumption of traditional, document-based engineering practice. This paper shares interim findings of a continuing research effort to investigate adaptation of the systems engineering leading indicators for digital engineering and use in model-centric programs.

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 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.

Since each of the indicators requires some additional considerations under digital engineering, the first year in this research focused on identifying potential modifications and interpretation guidance (Rhodes, 2020). The digital engineering environment and newer technologies open new possibilities for providing program leaders with leading insights into the effectiveness of systems engineering efforts. Accordingly, in the second year of the research the focus has been on defining specific model-based implications for the 18 leading indicators, and possible enhancements through three areas of inquiry. First, the research explores how metrics data can be obtained from digital system models that are produced by the systems engineering team. Second, it explores how information from descriptive system models can be extracted and composed as composite leading indicators that give overall indication of effectiveness versus a set of separate indicators. An illustrative case is used to show how measurement data can be extracted directly from descriptive system models and composed as enhanced leading indicators that can provide insight into effectiveness of engineering on a model-centric program. (*Note: Integrated model-based tools were used in investigations during this research project. The selection of tools in the examples is not intended as an endorsement; the software used by the research team was selected based on pre-existing availability and case examples).* Third, the research seeks to understand how interactive dashboards can be used to extract and more effectively display measurement information to positively impact program reviews and decisions.

The justification for pursing this research approach extends from the DoD Digital Engineering Strategy (DoD, 2018), which discusses five goals. Goal 3, Incorporate Technological Innovation to Improve Engineering Practice, has a Subgoal 3.2 that discusses the use of technological innovations to improve digital engineering practice. There are many technological innovations of interest; some are specifically relevant to both the measures of digital engineering effectiveness on a program and the enabling technologies to support collection, analysis, and display of leading indicators of engineering effectiveness. As noted in the strategy, “data analytics can help gain great insights from existing model data” (DoD, 2018, p. 14). The strategy recommends that stakeholders use technological innovations to improve decision-making and performance of computationally intensive engineering activities.

The collection and analysis of systems engineering measurement data falls under that category of activities. 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 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.

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 includes 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.

Measurement Specifications

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 (Roedler et al., 2010, adapted by Zheng 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. Developing the next version of the guide necessitates a community effort extending from implications identified in this research, insights from practitioners, and results of ongoing investigations and initiatives on digital engineering metrics.

Model-Based Implications for Leading Indicator Implementation

The existing 18 leading indicators, as investigated through semi-structured interviews and technical exchange workshops, were shown to have varying implications related to model-based systems engineering. The implementation of a leading indicators in context of digital engineering will be based on many factors, such as nature of the program, processes used by the enterprise, model-based toolset selection and implementation, engineering culture of the enterprise, and maturity of digital engineering in the enterprise, as well as external influences (e.g., customer preferences, etc.).

Based on research findings, the leading indicators are grouped into three subsets: (1) leading indicators most likely to be implemented with direct use of a model-based toolset; (2) leading indicators most likely to be partially implemented with use of a model-based toolset; and (3) leading indicators less likely to be implemented with use of a model-based toolset. The three groups of leading indicators are then summarized in Tables 2, 3, and 4 to highlight model-based implications. Prior to the summary tables, two indicators, *requirements trends* and *facility and equipment trends,* are discussed in greater detail.

Requirements Trends Leading Indicator

The *Requirements Trends* leading indicator is used to evaluate the stability and adequacy of the requirements to understand the risks to other activities toward providing required capability, on time and within budget. This is done through an evaluation of trends in the growth, change, completeness, and correctness of the system requirements definition, as well as the quality of and consensus around the system operations concept. This indicator provides insight into rate of maturity of the system definition against the plan, and whether the system definition is maturing as expected. Additionally, it characterizes the stability and completeness of system requirements that could potentially impact design, production, operational utility, or support.

Requirements growth, changes, or impacts that exceed expectations or exhibit a lower closure rate of TBDs/TBRs than planned may indicate insufficient quality of architecture, design, implementation, verification, and validation efforts. This in turn could result in elevated schedule and cost risks, and/or a future need for different levels or types of resources/skills.

Near Term: The use of requirements management tools and databases is a mature practice in systems engineering. Tracking the growth trends and volatility of requirements is therefore a relatively straightforward matter of the compilation of data on the requirements within the database and the development of processes for regular review and action where implied. These functions could be incorporated into or added to existing requirements management tools within the MBSE environment to assist program decision-makers in assessing progress during the system development.

Longer Term: MBSE tools and methods introduce a number of new ways to assess and understand the quality of requirements and the degree to which they are being met over the course of the system development life cycle. A transition to primary use of an MBSE approach in system development could enable a broader range of analysis and model checking.

The expression of requirements as executable models has been demonstrated to improve the quality of requirements and decrease errors relating to poorly-defined requirements (Micoun et al., 2018). Model-based requirements provide the ability to validate that the system model is logically consistent, and the ability to answer questions such as the impact of a requirement or design change, or the assessment of how a failure could propagate through a system. Using this approach, it is possible to verify design models using a simulation-based verification process in order to detect and remove design errors. Model-based requirements may be included in a curated database for reuse in other development efforts, with the potential for savings in time and resources.

Model-based requirements may be used in early system analysis to assess requirements completeness and correctness through the identification of gaps, conflicts, or redundancies in the existing requirements set, prior to the development of more detailed engineering models and analysis. MBSE analysis using model-based requirements could validate the requirements themselves and ensure they don't contribute to undesirable emergent behaviors at the system level. A potential indicator of requirements quality in the MBSE environment might include the percentage of requirements that are formatted and expressed as models and the rate and total proportion of requirements validation through modeling and simulation at both the component and system level.

Model-based requirements may be archived and reused across multiple development projects. Any issues that are identified with the requirements in one project could potentially be traced to other projects that use the same models. The traceability inherent in using these archived requirements models enables enhanced root cause analysis and system refinement, triggering actions to correct and validate the originating requirement to prevent continuing propagation of errors. An indicator of requirements maturity in an MBSE environment might include the proportion of requirements models that include a validation pedigree. The presence of requirements models without a validation pedigree (at least to a specific standard defined by the enterprise) could indicate greater risk of potential future requirements changes and instability in the system baseline.

Facilities and Equipment Trends Leading Indicator

The *Facility and Equipment Availability Trends* leading indicator is used to determine the availability of critical facilities and equipment needed for systems engineering activities over the project life cycle. The indicator is composed of two metrics, measuring facility availability and equipment availability. The intent of this indicator is to provide a view of facility and equipment availability on the project over time. Facilities and equipment are of different types and may provide key capabilities to the program. The facility availability measurement provides insight into the difference between the planned need for a facility type and the existing inventory of available facilities that meets the need for the desired capability. Insufficient facilities—labs, test ranges, floor space, etc.—of various types may cause a project to be unable to meet its customer needs, create costly overruns, and inability to meet schedule targets. Similarly, a project requires various types of equipment that also may provide key capabilities for the program. Equipment availability measurement provides insight into the difference between the planned need for an equipment type and the existing inventory of available inventory that meets the need for the desired capability. Insufficient equipment (fabrication equipment, measurement equipment, cleanroom equipment, test equipment, software and systems applications, etc.) may cause a project to be unable to meet its customer needs, create costly overruns, and inability to meet schedule targets. Facility Availability and Equipment Availability as measurable concepts assess whether adequate facilities and equipment can be allocated to the project to meet life cycle milestones. This reveals differences between systems engineering needs on the project and available facilities and equipment based on projected needs. The leading insights provided to the project are potential shortfalls of systems engineering related facilities and equipment, and potential problems with the project's ability to meet desired milestones (Roedler et al., 2010).

Near Term: As an initial step in adapting the existing systems engineering (SE) leading indicators, the measurement specification can be augmented by adding model-based systems engineering implications to the Implementation Considerations within the Additional Information section of the measurement specification. Model-based programs necessitate personnel have (or have access to) computing “equipment,” including desktop/laptop computers or workstations with adequate performance, access to networks and/or intranet, data and model repositories, model libraries, computer services support, data/cloud storage, etc. Facilities may include the individual engineer’s workspace, as well as collaborative spaces. There is also a need to have access to the selected version of model-based toolset that is maintained. The facilities and equipment need to support any required upgrades of versions, which may have implications for the existing computing facilities. Another implication consideration is that facilities and equipment must accommodate any necessary collaboration with other internal groups and/or external organizations (such as a supplier or customer) as needed. The facilities and equipment must be adequate to support this. This includes necessary facilities and equipment to support tool interoperability, data/model exchange, version compatibility control, model sharing, model security, etc. Model-based programs need to have adequate budget allocated, as insufficient availability of the necessary facilities and equipment will have major impact on systems engineering effectiveness.

Longer Term: As we look to the future of digital engineering, the issue with using the existing Facilities and Equipment Availability leading indicator is that it takes a somewhat decoupled approach at these rather than as highly interconnected, as is the case for MBSE. In fact, with the transformation of traditional engineering to digital engineering, there is a need to look at this in context of the larger digital ecosystem. This includes interconnected digital environments that extend beyond the boundaries of the engineering organization. In the existing SE leading indicator guide published in 2010, the Facilities and Equipment Leading Indicator has relatively less substance than the other indicators given that it was not a major focus of the team. With digital engineering transformation, taking the perspective of the overall digital engineering ecosystems is necessary. The success of systems engineering on a program will be fully dependent upon the environment and infrastructure available to participate as part of the larger ecosystem. The supporting infrastructure required for digital engineering (Bone et al., 2018) necessitates that a new leading indicator be developed respective to the importance it has to system success and the dimensions and complexity of that infrastructure.

Leading Indicators Most Likely to Be Implemented with Direct Use of a Model-Based Toolset

The first subset of leading indicators, as shown in Table 2, are those that are most likely to be implemented with the direct use of the program’s MBSE toolset. In this case, the base measures as shown in the respective measurement specifications in the leading indicator guide (Roedler et al., 2010) are likely to be obtainable from the system model and composed into a leading indicator. Assuming an effective user interface and any required trend data, this provides the ability to obtain real-time leading indicator information to better inform and accelerate decisions based on this information.

Table 2. Leading Indicators Most Likely to Be Implemented With Direct Use of Model-Based Toolset
(Roedler et al., 2010)

Leading Indicator	Insight Provided	Model-Based Implications
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.	• See the section Requirements Trends Leading Indicator for a detailed discussion.
System Definition Change Backlog Trend	Change request backlog which, when excessive, could have adverse impact on the technical, cost, and schedule baselines.	• Model-based tools will enable collection and analysis of data • MBSE enables fixing defects earlier in time, where less effort is typically required. Accordingly, historical trends will vary from model-centric programs.
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.	• Similar to requirements trends; see the section Requirements Trends Leading Indicator for a detailed discussion.
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.	• Similar to requirements trends; see the section Requirements Trends Leading Indicator for a detailed discussion. • Model-based tools may accelerate the pace of validation so historical data trend data may not be as useful.
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.	• Similar to requirements trends; see the section Requirements Trends Leading Indicator for a detailed discussion. • Model-based tools may accelerate the pace of verification so historical data trend data may not be as useful.

Leading Indicators Most Likely to Be Partially Implemented with Use of a Model-Based Toolset

The second subset of leading indicators, as shown in Table 3, are those that are most likely to be partially implemented with the use of the program's model-based toolset. For example, technical performance risk information might be associated with the system model, but there may be other programmatic risk information that is tracked elsewhere. The extent to which the five leading indicators in this table are able to be generated from a model is dependent on what types of models the program uses, and how model-based toolsets are customized and extended.

Table 3. Leading Indicators Most Likely to Be Partially Implemented With Use of Model-Based Toolset (Roedler et al., 2010)

Leading Indicator	**Insight Provided**	**Model-Based Implications**
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.	• Model-based tool sets provide opportunity to have risk associated with or directly included within models.
Risk Treatment Trends	Effectiveness of the SE organization in implementing risk mitigation activities. If SE is not retiring risk in a timely manner, additional resources can be allocated before additional problems are created.	• Model-based tool sets provide opportunity to have risk associated with or directly included within models.
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 the product design and/or project team's performance.	• Model-based approaches, methods, and tools will enhance technical performance measurement. • Ability to project planned value and predict variances may be improved, so tolerance bands may vary from traditional engineering.
Defect/Error Trends	Progress towards the creation of a product or the delivery of a service that meets the quality expectations of its recipient. Understanding the proportion of defects being found and opportunities for finding defects at each stage of the development process of a product or the execution of a service.	• With model-based approach errors and defects may be found earlier in time; software can automate finding and fixing some defects. • Necessitates defining an alternative to "defects per page." • Historical defect discovery profiles from traditional engineering will likely not be suitable; defects models and discovery profiles will need to be developed as experience grows
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.	• Models may become tracked work products in model-centric programs; criteria would need to be developed. • Models may influence the approval rate of system work products.

Leading Indicators Less Likely to Be Implemented with Use of Model-Based Toolset

The third subset of leading indicators, as shown in Table 4, are those that are less likely to be implemented with the use of a program's model-based toolset. Presently, these leading indicators would likely be tracked in a separate technical management tool or tracking system. Model toolset experts view it as possible to extend model-based toolsets to include any programmatic and process models in a model-centric environment. So, while at present there are likely to be few programs that have implemented this, the likelihood will increase over time as model-based environments evolve.

Table 4. Leading Indicators Less Likely to Be Implemented with Use of Model-Based Toolset (Roedler et al., 2010)

Leading Indicator	Insight Provided	Model-Based Implications
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.	• Increased use of models may enhance ability to assess potential impacts.
Review Action Closure Trends	Responsiveness of the organization in closing post-review actions. Adverse trends could forecast potential technical, cost, and schedule baseline issues.	• Model-centric programs are likely to have more continuous action item review than traditional. • Technical-related action items may be directly linked to models.
Systems Engineering Staffing & Skills Trends	Quantity and quality of SE personnel assigned, the skill and seniority mix, and the time phasing of their application throughout the project life cycle.	• Model-based approaches, methods, and tools require additional staffing and skills, possibly at different points in program. • Insufficient model-based staffing/skills have impact on cost, schedule, and quality.
Process Compliance Trends	Quality and consistency of the project defined SE process as documented in SEP/SEMP. Poor/inconsistent SE processes and/or failure to adhere to SEP/SEMP, increase project risk.	• Model-based programs will be using newer and/or developing processes integrated with toolsets. • Compliance deviations and comments recorded within the model enable automated compliance measurement. • Process compliance measurement needs to accommodate modifications to process given learning on program and/or other programs.
Facility and Equipment Availability Trends	Availability of non-personnel resources (infrastructure, capital assets, etc.) needed throughout the project life cycle.	• See the section Facilities and Equipment Trend Leading Indicators for a detailed discussion.
System Affordability Trends	Progress toward a system that is affordable for the stakeholders. Understanding the balance between performance, cost, and schedule and the associated confidence or risk.	• Assessing affordability under the digital engineering paradigm is likely to require different approach. • Lacking historical data, model-based programs need to develop, approach, and adjust measurement of this.
Architecture Trends	Maturity of an organization with regards to implementation and deployment of an architecture process that is based on an accepted set of industry standards and guidelines.	• Model-based approaches/tools will have influence on assessing maturity. • Programs should tailor base measures as needed to reflect advantages of model-based approaches/tools.
Schedule and Cost Pressure	Impact of schedule and cost challenges on carrying out a project.	• Minimal historical data available for the digital engineering situation, and setting notional values for thresholds may be challenging.

Composability of Leading Indictor Measurement Data

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. It also characterizes stability and completeness of system requirements which 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 constructs and digital artifacts. With modeling languages (e.g., SysML, LML) there are requirements diagrams, use case diagrams, activity diagrams, state machine diagrams, parametric diagrams, and others. With the advantages of model-based approaches, a leading indicator used to assess progress of system definition that uses only requirements would be a limited indicator. In this case, one would want to consider progress of systems definition to include composition of measurement information from system diagrams of all relevant types.

Composability concerns selection of elements that can logically and reasonably be assembled. In context of this research, the focus is on composability of base measures extracted from a digital system model or digital process model used to produce a leading indicator. An initial step is to consider the existing 18 leading indicators. 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 explore the future of leading indicators for digital engineering program decision-making.

Illustrative Case Discussion

An illustrative case has been used in the research to explore how digital engineering is expected to modify and/or enable the leading indicators most likely to be implemented with direct use of model-based toolsets. These five leading indicators (see Table 2) all relate to aspects of requirements management. 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 the chosen model-based tools used. For the purposes of this illustrative case, Innoslate® was 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). LML provides an organized and structured terminology for systems engineering, enterprise-defined extensions, and includes features that support the earliest concept stage throughout the life cycle to disposal (Dam, 2019, pp. 6, 10). 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 *Action* entity in the LML ontology is the primary building block for functional models. Similarly, the *Asset* entity 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.

The other key basic concept in functional models is *Input/Output*, which represents the flow of information 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.

Relationships are used to make connections among entities. Decomposition is denoted with the *decomposed by/decomposes* relationships. A functional model Action can be allocated to a physical model Asset using the "*performed by/performs*" relationship. 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. Standard entity attributes are defined in the LML ontology along with standard relationships and the entity types they connect (LML Steering Committee, 2015).

Tracking the trends needed for the leading indicators (Table 2) requires taking snapshots of metrics values at intervals over time as the program proceeds. If integrated into an LML model, this program management data would be stored as objects in the database to facilitate integrative analysis with other program data.

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) computed 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 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 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 Innoslate 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.

Requirements Validation Trends and Requirements Verification Trends

Systems engineering practice involves beginning requirements validation and verification early in the project as requirements are found and entered into the database. 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. The leading indicators for requirements verification and requirements validation could be improved by adding a measure for progress on developing test plans to complement the metric for successful completion of validation and verification testing. Product validation and verification also needs to be considered holistically as well as individually by requirement. The model can be used with simulation tools to predict the behavior of the whole system or subsystems.

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. This provides more transparency to 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 find patterns that could be used to judge the effectiveness of the engineering on an ongoing program. For this reason, the knowledge and recent advancements in visual analytics may offer significant support in processing and displaying measurement data.

Vitiello and Kalawsky (2012) state the "guiding process in visual analytics is a synergy between interactive visualization and automated analysis of the data." They discuss an approach that integrates a visual analytic-based workflow to 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.

Interactive Dashboards

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 6–7 years (Dwyer, 2020). Leading indicators can help stakeholders see how a project or program is progressing throughout the life cycle 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. 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. Dashboards are often created as views into program data that has been extracted and loaded into a data warehouse. 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.

Selby (2009) argues that interactive dashboards facilitate effective management. 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) extend Selby and propose a framework for guiding leading indicator development and usage. Recent work by Thiruvathukal et al. (2018) shows the potential for using open source software repositories in the development of software metrics dashboards. Nadj (2020) addresses how interactive dashboards help managers in gaining and maintaining situational awareness to understand the context of metrics.

Discussion and Future Directions

Potential impact of adapted and extended leading indicators is twofold: (1) to continue to provide visibility into the future state through use of leading indicators in model-centric programs; and (2) to enhance insights provided by the leading indicators related to digital models and artifacts that enrich the systems development practice. The current set of leading indicators is predicated on use of document-driven processes and major milestone reviews. The transformation to digital-model based engineering will result in the use of digital artifacts, with more frequent review of the expected system performance (Parrot & Weiland, 2017). One of the expected outcomes of digital engineering is to move away from milestone design reviews to more continuous reviews given access to the maturing system model. Leading indicators can be very supportive of this goal (Orlowski, 2017; Orlowski et al., 2015). An open question is how trend information for models and digital artifacts (e.g., SysML diagrams) could be used in a similar manner to predict when the model is in a state where a review activity is most useful.

Interim Research Findings

Interim outcomes of the research include knowledge for augmenting measurement specifications of existing systems engineering leading indicators to describe model-based implications, and illustrating what is possible for enhanced measurement through direct use of model-based toolsets. Another interim outcome is initial investigation of how visual analytics and interactive dashboards may be used to provide leading indicator information that provides a deeper understanding for program leaders. This research has identified considerations and additional interpretation guidance to augment existing systems engineering leading indicator measurement specifications. The results are aimed at assisting systems engineering organizations that have been using leading indicators as they transition to model-based systems engineering.

This research has also included initial investigation for how 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.

Related Research and Initiatives

Interim and final research outcomes are shared with several working groups as input to future research on digital engineering metrics and revision of the leading indicators guide. The DoD Systems Engineering Research Center (SERC) performed a research activity in 2020 that investigated digital engineering transformation metrics (McDermott et al., 2020). Metrics categories were derived as a general categorization, including adoption, user experience, velocity/agility, quality, and knowledge transfer. Literature review and a survey were performed, resulting in a set of top metrics categories related to the benefit of digital engineering. Although the focus of the leading indicators and digital engineering metrics are somewhat different, their relationship is worth considering as potential additional leading indicators are developed.

The outcomes of the SERC investigation, other ongoing measurement related efforts, and foundational work by PSM are employed in a broader Aerospace & Defense Digital Engineering Metrics Initiative. This initiative brings together a diverse team including industry associations, government agencies, FFRDCs, UARCs, academia, and industry (including AIA, NDIA, INCOSE, OUSD R&E, SEI, SERC, the Aerospace Corporation, PSM, MIT, and several companies). The effort aims to define an industry consensus measurement framework and candidate performance indicators for digital engineering, and to align measures with business information needs for project execution and organizational performance improvement.

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 metric data 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 need to be examined in regard to 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.

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 life span. Two aspects for future investigation are techniques emerging from visual analytics and 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 further elicit ideas from the systems community on program level indicators and enterprise-level indicators. Desirable research is 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.

References

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.

Dam, S. (2019). *Real MBSE: Model-based systems engineering (MBSE) using LML and Innoslate*.

Dam, S. (2020). *How to perform a model-based review* (White paper). https://www.innoslate.com/resource/model-based-review-whitepaper/

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

Dwyer, M. (2020). *Understanding acquisition speed for the Defense Department's costliest and most complex programs* (CSIS-AM-20-159). CSIS. https://dair.nps.edu/handle/123456789/4288

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

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

Gerst, K. J., & Rhodes, D. H. (2010). Strengthening 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. *Procedia Computer Science, 28*, 71–80.

LML Steering Committee. (2015). *Lifecycle modeling language (LML) specification*. https://lifecyclemodeling.org/wp-content/uploads/2021/01/LML_Specification_1_1.pdf

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* (Technical report SERC-2020-SR-001). Systems Engineering Research Center (SERC).

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.

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.

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

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*. http://www.psmsc.com

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

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.*

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.

Roedler, G. J., & Rhodes, D. H. (2007). *Systems engineering leading indicators guide: Version 1.0.* Massachusetts Institute of Technology.

Roedler, G. J., Rhodes, D. H., Schimmoler, H., & Jones, C. (2010). *Systems engineering leading indicators guide: Version 2.* Massachusetts Institute of Technology.

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

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

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. https://doi.org/10.1016/j.procs.2013.01.006

Systems Engineering Research Center. (2020). http://sercuarc.org

Thiruvathukal, G. K., Hayward, N. J., & Läufer, K. (2018). *Metrics dashboard: A hosted platform for software quality metrics*. https://arxiv.org/pdf/1804.02053.pdf

Thomas, J. (2007). *Visual analytics: An agenda in response to DHS mission needs*. National Visualization and Analytics Center.

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

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), 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), 529–554.

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

www.acquisitionresearch.net