Data Traceability for Complex, Distributed Live, Virtual, Constructive Simulation Events

Eric Tollefson, Jonathan Andrews, Michael O'Connor
Trideum Corporation
Huntsville, AL
etollefson@trideum.com, jandrews@trideum.com, moconnor@trideum.com

Tilghman Turner
Redstone Test Center
Redstone Arsenal, AL
tilghman.a.turner.civ@army.mil

ABSTRACT

The primary purpose of conducting a test and evaluation event is to collect the data required to evaluate and assess an acquisition system. However, the complexity of a distributed live, virtual, constructive (LVC) simulation environment required to replicate multi-domain operations for an operational test event makes it difficult during both planning and execution to ensure that the data requirements are met. Even after a test team identifies its analytic requirements down to the specific data elements required to address them, it must then ensure that the test plan and environment capture those requirements and continue to do so as a wide variety of test personnel refine and modify the plan leading up to the event. During execution, the test team must be able to characterize the impacts of unplanned changes, such as weather events or instrumentation failures, on the data collection requirements to decide how to react.

In this paper, we describe a comprehensive methodology to maintain data traceability—the ability to associate multiple aspects of data context, lineage, and provenance throughout the test data lifecycle by means of relationships and interconnectedness. This methodology is nested in a larger Distributed LVC Event Process for reducing risk when building complex LVC simulation environments. We will depict the methods and tools we developed thus far to decompose the data context, which we define as the surrounding information and circumstances under which the data is collected, processed, and interpreted. We will then show how we establish the linkages between the many types of data context, e.g., analytic requirements, scenario elements, resources, and event scheduling, to ensure that the complex graph of relationships is captured. We will also describe the challenges and trade-offs associated with the application of our methodology, and our plans for future work.

ABOUT THE AUTHORS

Eric Tollefson, Ph.D., is a Senior Solutions Architect for Data and Analytics at Trideum Corporation. Prior to joining Trideum in 2022, he retired from the Army as a Colonel with over 28 years of service as an Operations Research/System Analyst and Infantry officer. His assignments included the U.S. Army Network Enterprise Technology Command, U.S. Cyber Command, U.S. Military Academy (USMA) Department of Systems Engineering; Training and Doctrine Command (TRADOC) Analysis Center; deployments to Iraq and Afghanistan; and company-grade assignments as an Infantry officer. Eric attended the Georgia Institute of Technology where he earned a Ph.D. in Industrial Engineering, MS in Statistics, and MS in Operations Research. He also earned a BS in Engineering Physics from USMA.

Jonathan Andrews is a Solutions Architect for Data and Analytics at Trideum Corporation. He obtained a BS and MS in Electrical Engineering from the University of Alabama in Huntsville (UAH). His MS thesis mathematically characterized dichotomous and centroid tracking behavior of RF missile seekers. Before joining Trideum Corporation, he worked as the Director of Data Science at CFD Research Corporation and as a Senior Engineer at Dynetics, Inc. in the RF and Electronic Warfare Solutions Department supporting the DIA Missile and Space Intelligence Center (MSIC). Previous experience includes high-fidelity radar modeling and simulation development with an emphasis on human operator-in-the-loop analysis supporting all international FVEY partners. These efforts helped modernize pilot training requirements based on current and next-generation surface-to-air missile threat capabilities.

Michael O'Connor is Chief Technologist at Trideum Corporation. Mr. O'Connor has more than 35 years of experience in Modeling and Simulation (M&S) and has been a key participant in developing distributed modeling and simulation standards, including IEEE 1278 and IEEE 1516. He has held many community positions, including Chairman of the Simulation Interoperability Standards Organization (SISO) Standards Activities Committee and Chairman of the SISO Executive Committee. He has served as the chair of the I/ITSEC Simulation Subcommittee, I/ITSEC Training Subcommittee, I/ITSEC Tutorial Subcommittee, and is currently the Deputy Program Chair for I/ITSEC. He has led the development of multiple simulations using DIS, HLA, and TENA. In addition, Mr. O'Connor has led the technical integration of large multi-architecture distributed, live, virtual, constructive (DLVC) events. He is a key member of a team that continues to develop a DLVC integration and execution process based on IEEE 1730. He holds a bachelor's degree in Computer Engineering from Auburn University and a Master of Science in Computer Science from the University of Alabama in Huntsville. Mr. O'Connor is a CMSP.

Tilghman Turner is the Live, Virtual, and Constructive Subject Matter Expert at the Redstone Test Center, Huntsville, AL, with more than 27 years of experience in Modeling and Simulation (M&S), Live Flight testing, and Hardware in the Loop (HWIL) and Distributed Testing.

Data Traceability for Complex, Distributed Live, Virtual, Constructive Simulation Events

Eric Tollefson, Jonathan Andrews, Michael O'Connor
Trideum Corporation
Huntsville, AL
etollefson@trideum.com, jandrews@trideum.com, moconnor@trideum.com

Tilghman Turner
Redstone Test Center
Redstone Arsenal, AL
tilghman.a.turner.civ@army.mil

INTRODUCTION

The primary goal of a test and evaluation (T&E) event is to collect data that will be analyzed to evaluate a system under test (SUT). In today's multi-domain operations (MDO) environment, the systems of systems within which the SUTs operate continue to grow in complexity; consequently, so do the live-virtual-constructive (LVC) simulation test events required to immerse a SUT in a realistic MDO environment. Such test events can generate a tremendous volume of intricately interrelated data which will be used to evaluate and assess the effectiveness, suitability, and survivability of the SUT. But how does the test team know if they have collected the right data?

Early in planning, the test analysts create a detailed analysis plan that includes the individual data elements that must be collected. They then pass those requirements to operational planning personnel to develop the scenarios that will generate the necessary conditions for the evaluation. But those who design the scenarios are usually different from those who implement them in simulation. Other personnel, such as instrumentation specialists, data collection managers, and test monitoring personnel own parts of the planning process as well. Having multiple specialists involved in the development of an extremely complex test plan makes it difficult to ensure that the originally identified data requirements propagate throughout the plan and retain their original intent. Each handoff between test team members requires that the data collection requirements are both passed on to and understood by the recipients, who have varied levels of expertise and knowledge of the overarching test objectives. We call this *forward* data traceability. Similarly, if a downstream planner or unforeseen incident leads to deviations from the plan, the test team needs mechanisms in place to determine whether and how those changes affect the original data collection requirements. We call this *backward* data traceability. For example, a simulation modeler may have to alter the given scenario due to limitations with the software, thereby changing the conditions originally intended to address a specific evaluation issue. During execution, the test team must be able to understand the impact on the data collection plan of unexpected issues or interruptions, such as instrumentation failures or disruptive weather events, in order to decide whether to extend or repeat a portion of the test. These issues may be manageable for simple events, but are extraordinarily difficult for integrated, distributed LVC events representing complex MDO environments. There are currently no methodologies that fully address this concern.

Failure to ascertain whether a test plan comprehensively addresses all evaluation issues—or to assess the impact of subsequent modifications on test effectiveness—can have significant and wide-ranging implications. First and foremost, the test could simply fail to produce the necessary data, invalidating the event and potentially requiring additional resources and time to make up for the shortfall, which in turn could delay the acquisition program. During test execution, incomplete information could lead to suboptimal decisions such as extending the test event unnecessarily or failing to extend it when required. During planning, insufficient information could result in inefficient resource allocation, unnecessary rework, and extended planning timelines.

We propose that a comprehensive methodology for maintaining data traceability—one that verifies the validity, completeness, and lineage of data at every step of the event planning and execution process—will ensure that the right data is collected during complex MDO T&E events. In this paper, we lay out such a methodology, starting with a discussion of the distributed LVC event process that frames our effort, as well as an overview of key data concepts used throughout the paper. The main portion of the paper explains our proposed methodology to date along with some elucidating examples. The final sections describe key challenges we encountered and outline future steps for building and refining the methodology. Although our work is in direct support of T&E, we will also intersperse suggestions on how it can be extended to training.

Distributed LVC Planning Process

Our work is nested within the Distributed LVC Event Process for integrating and executing large, distributed LVC events (Zinser et al., 2024). That process is based on the Distributed Simulation Engineering and Execution Process (DSEEP) standard (IEEE, 2010). The implementation has nine steps (adding two additional steps to DSEEP) which are divided into 27 activities, and includes a detailed set of processes, templates, and guidance on how to perform the activities. The goal of the process is to mitigate substantial integration and event validity risks by producing a verified distributed LVC environment in which to conduct an event. The process was developed to support distributed LVC T&E, but can apply as well to research and development, training, and experimentation.

While the current Distributed LVC Event Process addresses analytic and data collection activities and tasks, those aspects are not as well developed as many of the other activities and outputs. The process also falls short in ensuring the traceability of data requirements throughout the steps and activities. Thus, even if the resulting environment is verified and validated, it is still possible to conduct an event that fails to collect the right data. To address those gaps, we began conducting a series of working group meetings composed of T&E, LVC simulation, and data subject matter experts. Our work began in August 2023 and remains ongoing. This paper describes our methodology and progress to date, as well as the road ahead.

Key Data Concepts

The primary goal of our effort is to develop methods and tools that assist the test team in ensuring that it collects the right *test data*, that is, data produced and collected during the execution of a test event to address one or more evaluation issues. That data will either be manual data generated by a test team member, observer, or participant based on observation or opinion, or instrumentation data that is collected digitally by a test instrument or system. For test data, we are concerned with both data provenance and data lineage. There is a surprising lack of consistency regarding the definitions of and relationships between these two concepts, so we define them here. *Data provenance* is the record of a dataset's origin, including where it came from, how it was created, who generated it, and any modifications it underwent throughout its lifecycle (Mucci, 2024). *Data lineage* traces how the data under analysis is acquired, transformed, and used over time, including how new datasets are derived, where the data moves, how it changes, and how it ultimately supports analytics or decision-making (DAMA International, 2021). Thus, data provenance deals primarily with the generation and modification of the data at the point of origin, whereas data lineage addresses the transformations, movement, and use of the data throughout its lifecycle.

Central to our effort is the concept of *data context*—the surrounding information and circumstances under which data is collected, processed, and interpreted (Pappu, 2024). Data context encompasses the details and conditions that provide a framework for understanding the test data, including the source of the data, the purpose of its collection, the time and location of collection, and any significant variables or factors that may affect its meaning and relevance. It includes information about the evaluation issues to be addressed, the data requirements, the scenarios within which the data is collected, the execution event schedule, etc.

The ultimate goal is to maintain *data traceability*, which we define as the ability to associate multiple aspects of data context, lineage, and provenance throughout the test data lifecycle by means of relationships and interconnectedness. Data traceability provides the test team the means to

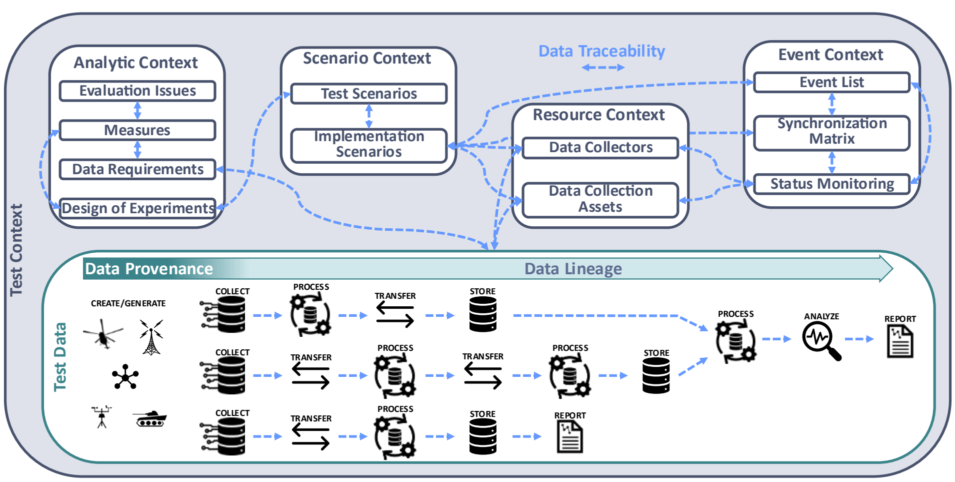

Figure 1. Key Data Concept Relationships

understand how the different aspects of the test plan relate to and affect each other. Figure 1 graphically depicts the relationships between these concepts. The interrelated components of analytic, scenario, resource, and event context trace to and directly influence the generation, collection, and understanding of the test data.

METHODOLOGY

We discuss the proposed methodology and tools in two parts. In the first part, we describe the analysis planning process, which decomposes the high-level test objectives into detailed data elements and experimental design components to capture the analytic context. The second part focuses on scenario development, describing the design of test scenarios and their implementation to capture the scenario (or mission) context. Analysis planning and scenario development are much broader processes than what we will discuss in this paper. We focus specifically on the aspects of those processes necessary to ensure data traceability.

Analysis Planning

The analysis planning methodology begins with the receipt of the test objectives and yields two primary sets of products—the analytic framework and the design of experiments.

Analytic Framework

The analytic framework, which captures the analytic requirements for the test, starts with the overarching test objectives and follows a series of decompositions that ultimately result in a comprehensive set of data elements to be collected during the event. Fortunately, many methodologies for accomplishing this already exist. For example, the U.S. Army Test and Evaluation Command (ATEC) captures these decompositions in products called the Data Source Matrix and Pattern of Analysis (Department of the Army [DA], 2003). Similarly, the U.S. Navy Operational Test and Evaluation Force (OPTEVFOR) planning handbook (2020) and test design checklist (2021) describe a series of products (e.g., the Measure Matrix, Traceability Matrix, Critical Tasks and Measures Table, Measures-to-Data Requirements Matrix, and Conditions-to-Data Requirements Matrix (CDRM)) which capture similar information. The Air Force Operational Test and Evaluation Center (AFOTEC) also employs a well-established methodology, formalized in its Operational Test and Evaluation (OT&E) Construct, which decomposes test objectives into critical operational issues (COIs), and further into measures of effectiveness (MOEs), measures of performance (MOPs), and measures of suitability (MOSs), each ultimately defined by one or more data elements (Alvarado & Bradley, 2024). This structured decomposition supports traceability from mission objectives to specific, measurable criteria. Notably, the origins of this approach date back to a 1974 Air Force report that sought to standardize management and analytical procedures for operational test design, explicitly emphasizing standardized data elements and MOEs (Braddock & McDonald, 1974). These examples, and other similar methodologies, are tailored to the organizations and events they support. In developing our methodology, our intent was to generalize enough to meet the minimum requirements for ensuring data traceability while remaining sufficiently flexible to apply across a wide range of testing, experimentation, and training applications. The Army, Navy, and Air Force methods described above are consistent with what we have developed. Table 1 shows the four key components of the analytic framework.

Table 1. Key Components of the Analytic Framework

Component	Description
Evaluation issues	Overarching questions and issues designed to ensure a comprehensive plan for addressing a system's operational effectiveness, suitability, and survivability (DA, 2003). Includes or can be derived from Key Performance Parameters, Key System Attributes, Additional Performance Attributes, Critical Technical Parameters, Critical Operational Issues and Criteria, etc.
Measures	Information that infers the condition, state, or existence of something, and provides a reliable means to ascertain performance or effectiveness (Joint Chiefs of Staff, 2020). Includes MOEs, MOPs, MOSs, Operational Measures, Technical Measures, etc.
Data requirements	Quantitative or qualitative piece of information that is relevant to the determination or categorization of one or more measures (DA, 2003).
Data elements	Lowest level of information collected and only require direct observation, timing, or recording by one person (or piece of instrumentation) at a single location at a single time (DA, 2003).

To develop the analytic framework, the test team first identifies the evaluation issues that support the test objectives. Each of those evaluation issues is then decomposed into its supporting measures. For smaller events, the test team may not have, or need to identify, evaluation issues but may instead go straight to measure identification. For each measure, the team identifies the general data requirements for addressing the measure and, subsequently, the specific data elements necessary. In some cases, the team may prefer to directly identify data elements without explicitly identifying or recording data requirements. Figure 2 shows the hierarchical structure and relationships among the analytic framework components, where the dashed-line boxes indicate optional components.

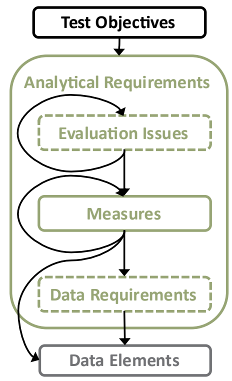

Figure 2. Analytic Framework Components

Critical to data traceability is identifying *conditional measures*—measures that are dependent on a specific set of conditions occurring during test execution. For example, evaluating a direct-fire weapon system's effectiveness may require target presentations at pre-defined ranges, exposures, or visibility conditions. Other measures, such as those addressed by servicemember-completed surveys may not depend on scenario conditions. In the subsequent scenario development phase, the test team will trace conditional measures to the parts of the scenarios that address them. The team should also account for data that must be collected but may not directly support a measure. For example, an external stakeholder may levy additional data requirements on the test teams that do not specifically address an evaluation issue but instead may be 'targets of opportunity.' To enable data traceability, all components of the analytic framework should use a unique numbering schema (e.g., MOE 2.3.2) to facilitate the establishment of relationships with other parts of the test plan. Although the analytic framework is developed early in the planning process, the test team will continue to update and refine it throughout planning.

Design of Experiments (DOE)

Broadly speaking, DOE is a systematic, efficient method that enables analysts to study the relationship between multiple input variables and key output variables (JMP Statistical Discovery, LLC, 2025). The type we address in this subsection is a formal DOE—one that is developed through the application of statistical methods—although the methods discussed can apply to informal DOE as well. Not all events require or include a formal DOE. Even when they do, the DOE may only apply to parts of the event, rather than defining the conditions for the entire event. The test team may also (or exclusively) identify conditions that must occur but do so outside a formal DOE construct. Our methodology captures such requirements both through the identification of conditional measures as previously discussed and during the scenario development process described in subsequent sections.

There are numerous resources for selecting and creating experimental designs. Our methodology only specifies that the team identifies and consults the sources of information that will drive the design, and identifies the key components of the design:

- Factors – input variables;
- Conditions – levels for each factor;
- Controls – whether the factor is systematically varied, held constant, or recorded but not controlled; and
- Responses – output variables.

Ultimately the test team should have a list of unique combinations of factors and conditions that will be executed during the test, which is essentially the characterization of the experimental design itself. Each of those factor combinations will have a unique identifier for establishing relationships with other parts of the test plan, particularly the test scenario. Figure 3 shows the relevant aspects of the DOE that should be captured, as well as the data traceability connections with the analytic framework. As with the analytic framework, the test team will continue to update and refine the experimental design products throughout the test planning process.

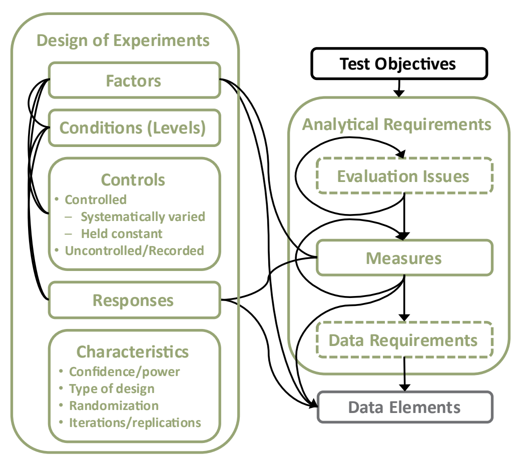

Figure 3. DOE Traceability to the Analytic Framework

These analysis planning processes can be extended to training as well. For example, the Army uses training and evaluation outlines (T&EOs) to specify performance and proficiency standards for collective and individual tasks

(DA, 2021). Within the T&EO is the Objective Task Evaluation Criteria Matrix which stipulates the required operational environment factors similar to a DOE. The Performance Steps and Measures section of the T&EO captures the performance measures that must be observed. Our methodology adds the rigor of formally specifying the data elements required to address the performance measures as well as the event design and conditions that must occur relative to the measures.

Scenario Development

The process of developing scenarios from the data traceability perspective is not as well documented or consistent as analysis planning. Analysis planning is focused on collecting the data necessary to address the overarching evaluation issues and therefore naturally lends itself to establishing data traceability. But while scenarios should be, and are, created with data requirements in mind, the resulting products are insufficient to establish full data traceability. For example, Army DA Pamphlet 73-1 describes the development of an event dendritic, which is "a hierarchical decomposition of system functions into data required for analysis and evaluation" which is a decomposition "by the sequence of events performed" (2003, p. 63). It also later describes test event development in terms similar to an experimental design (2003). As currently implemented, though, those products leave significant gaps between the test's operational conditions and data requirements. The Navy OPTEVFOR guidance goes even further in its description of the CDRM and Traceability Matrix, describing the linkage of measures to tasks and subtasks, conditions to subtasks, and conditions to data requirements. Those products are deliberate about establishing data traceability, but do not address some of the unique challenges of the LVC simulation environment. In LVC environments, the team must maintain data traceability from the scenario planners to the live component planners and simulation modelers, neither of whom likely have sufficient insight into the data requirements that drove the scenarios. The greater the number of personnel involved in test planning, the more likely it is that someone violates analytic or planning requirements of which they are not aware. Further, Army test planners, our primary audience, mainly develop scenarios via military symbology on a map that may or may not include a decomposition of tasks and events.

To address these gaps, we set out to create a methodology for representing scenario (or mission) context information that facilitates test planning and data traceability. We distinguish between three types of scenarios:

- *Operational Scenario* – A scenario or scenario component developed by a Service or Joint authority for acquisition-related decisions and activities. These are not developed by the test team but guide the development of the scenarios used during the test events.
- *Test Scenario* – A collection of missions, tasks, and events derived from the Operational Scenarios that provides the operational, or mission, context for the test event. The Test Scenario represents the instantiation of the Operational Scenarios on the actual terrain of the test with an explicit timing component.
- *Implemented Scenario* – The realization of the Test Scenario in the LVC test environment. The Implemented Scenario is the actual LVC representation of scenario entities, activities, effects, and conditions.

To enable the overarching goals of our effort, a scenario development methodology must satisfy a few key requirements. First, the tools and techniques must facilitate the association of scenario elements with other aspects of the test event plan to enable the traceability of analytic requirements both upstream and downstream in the planning process, as well as to support the linkage of scenario context information to the collected test data for subsequent analysis. Second, the methods must allow the user to separate the scenario information into distinct parts as necessary for operational or data management requirements (i.e., decomposability). Third, the methods should be general enough to accommodate various lexicons and taxonomies used for scenario development and representation across different types of test, training, and operational processes. Finally, the methods must be user-friendly, that is, not onerous to use; flexible enough to allow trade-offs between the level of detail provided and the time and effort expended; and compatible with current methods of developing Test Scenarios.

Test Scenario

The Test Scenario is the fundamental element of scenario planning. Test events typically involve multiple Test Scenarios to collect data on the SUT under varied conditions and mission types. Each Test Scenario is decomposed into *frames*, which are graphical representations of time-based subsets of the overarching Test Scenario. They are typically shown as military symbology on a map or satellite image. Each frame represents a set of mission-related activities that occur at a point in time, with the time intervals between frames either remaining fixed or, more likely, varying in duration. A sequence of frames communicates details that the planners need to implement the scenario in the LVC environment, as well as the traceability information needed to ensure that the Implemented Scenario satisfies analytical and other planning requirements. Figure 4 shows a notional example of a frame sequence for a Test Scenario.

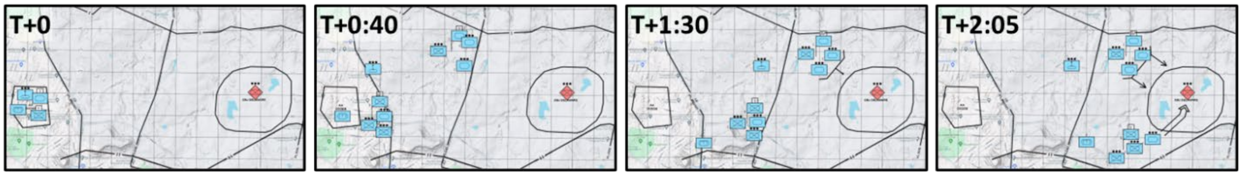

Figure 4. Example Test Scenario Frame Sequence

In a given test event, there will likely be multiple Test Scenarios. Each Test Scenario has a unique identifier to distinguish it from the others (e.g., Test Mission 6), along with any pertinent, descriptive metadata (e.g., battalion air assault, pilot run vs. record run, etc.). The scenario team then develops a set of frames for each Test Scenario, uniquely identifying each by relative time (e.g., H+1:25, T+0:45, etc.) since actual execution times will not be known at this point. In some cases, it will be difficult to assign times to frames, such as when the transition between frames is based on conditional events (e.g., a unit crossing a phase line). However, the test planners must use their judgment to estimate a time because synchronization across LVC components requires it. In those cases, the transitional conditions should be recorded in the metadata for the frames in question. The team will represent the scenario via military icons/graphics on a map or imagery, identifying both battlefield entities and required battlefield effects (e.g., electromagnetic warfare (EW), cyber).

The ‘art’ of frame development is deciding how many frames to create. Increasing the amount of time between frames eases the burden on the initial scenario planning team but requires the LVC implementation team to make more assumptions about what happens between frames. Thus, the number of frames is inversely proportional to the amount of free play allowed in both the live and simulated components of the test. It is also important to recognize that frame development necessarily involves more than just military planners and should include analysts, technologists, and other stakeholders who have insights into the data requirements as well as the downstream needs of the other planners.

Once the test team has created the frame sequence, each frame is further decomposed into *sub-frames*, which are subsets of a frame identified for traceability purposes. Sub-frames serve three purposes. The primary purpose is to establish traceability of the scenario and its components to test requirements and other parts of the test plan. Sub-frames can also capture critical scenario elements that require separate tracking or control during the event. Finally, they provide the details necessary for test planners and participants to carry out their roles, for example, for LVC modelers to implement the scenarios in simulation or for live participants to prepare tactical plans, operations orders (OPORDs), and other planning artifacts. See Figure 5 for an example.

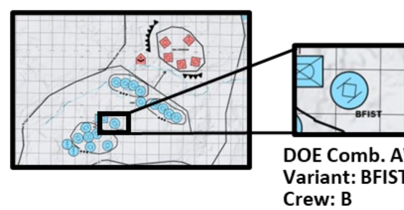

Figure 5. Example of a Test Scenario Sub-frame

Each sub-frame contains a unique set of battlefield entities/effects that traces to one or more test requirements or test plan components and must adhere to a set of conventions. First, all sub-frames must be location-based, although locations in a single sub-frame need not be contiguous (e.g., threat radar and launcher). Effects could be modeled via orientation (e.g., radio frequency (RF) emitters), area/volume (e.g., chemical plumes), or through affected entities (e.g., cyber attacks). Second, the set of all entities and effects in the frame constitutes a sub-frame. Third, entities and effects may be part of multiple sub-frames within the same frame; in other words, sub-frames may overlap.

For each sub-frame, the test team creates a unique identifier, which is a combination of the frame identifier and a unique sub-frame designator. Sub-frame designators are not necessarily unique from frame to frame, and, if creatively used, can make the decomposition more intuitive and easier to reference. For example, standard designators based on area, theme, or purpose that are consistent across frames and Test Scenarios, such as labeling DOE combinations (e.g., DOE_4-1), can make it easier to verify traceability and coverage across the scenario. The test team should then record data traceability information, that is, the test requirement or test plan component to which the sub-frame traces, as well as any conditions that must occur to satisfy an analytic or other requirement, and any other pertinent information that is not obvious from the graphics (e.g., off-frame scenario elements). The team will label the sub-frames with the unique identifiers graphically on the frame and use tables to record the linkages to the DOE, measures, and other planning elements.

Traceability of the sub-frames to other planning components takes three forms. First are linkages to DOE combinations. Not all events will have a formal DOE, but for those that do, every DOE combination must link directly to one or more sub-frames. That mapping is many-to-many. A DOE combination may be iterated multiple times during a single Test Scenario, or it may endure over multiple frames (and thus sub-frames). Similarly, a single sub-frame may address multiple DOE combinations (particularly for situations where there are multiple DOEs in play simultaneously). For situations where the timing of DOE occurrences is ambiguous and difficult to assign to a time frame, the test team must still estimate the time for those occurrences to adhere to the rule that frames be time-based. The second form of traceability is between sub-frames and measures. Recall that the team identifies conditional measures during analysis planning. The purpose of any scenario is to create the conditions necessary to collect the required data. Thus, the team will use sub-frames to capture where such conditions occur and link those back to the supported measures. Note that DOE is also about creating conditions, so these first two steps are not independent. This second step should capture the required conditions not accounted for by the DOE. As with DOE, the linkages between measures and sub-frames are many-to-many. The third form of traceability is to external requirements (e.g., Operational Scenario products). Such traceability may not be critical from a data perspective but is necessary to ensure the scenarios also comply with external guidance for validation purposes. Figure 6 shows an example of a sub-frame decomposition with traceability information.

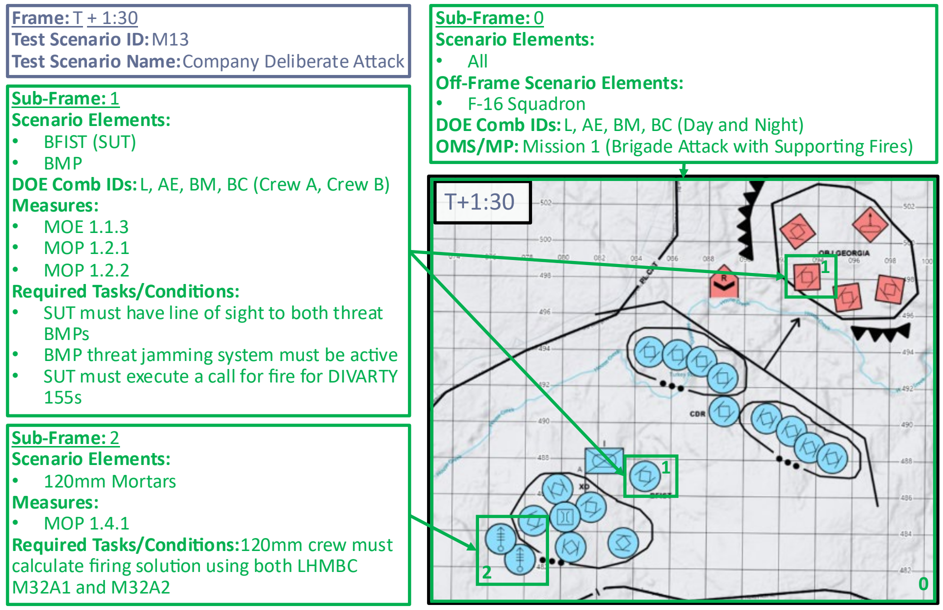

Figure 6. Example Sub-frame Decomposition

The last concept for Test Scenarios is that of *layers* – subsets of components within a frame that facilitate visualization of related information. Layers are independent of sub-frames. The team may use them, for example, to show blue forces (BLUFOR) versus opposing forces (OPFOR), fires overlays, etc. Within software solutions, layers may be implemented as filters based on properties and attributes assigned when the entities and effects are created. Layers serve an important visualization function to facilitate awareness and understanding. See Figure 7 for an example of layers.

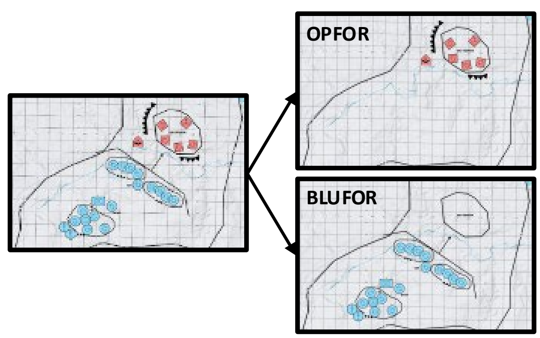

Figure 7. Example of Test Scenario Layers

To summarize, the primary components of the Test Scenario are frames, sub-frames, and layers. Frames capture the sequence of mission-based activities over the course of the scenario. Sub-frames provide a means of linking subsets of the scenario to other planning elements, as well as to provide additional detail necessary for planning. Layers filter frame information visually to facilitate situational awareness. Thus, frames are for synchronization; sub-frames are for traceability; and layers are for visualization.

Implemented Scenarios

The primary objectives of Implemented Scenarios are to identify how the entities, effects, and conditions from the Test Scenario are implemented via LVC components; facilitate the verification that the conditions required by the Test Scenario are achieved by the LVC environment; and provide traceability between the LVC components and both downstream and upstream planning products and processes. There is a one-to-one mapping between Implemented Scenarios and Test Scenarios. The Implemented Scenario consists of three primary products: the LVC Implementation Tables, the Conditions Implementation Table, and the LVC implementations themselves.

Table 2. Example LVC Implementation Table (Scenario Elements)

LVC Implementation Table Test Scenario M24							
Scenario Elements	Subsystem	Unique ID	LVC Representation	Representation Level	Member Application	Application Instance	Applicable Frames
A/1-333 IN							
CDR		A43	Live	Entity			All
XO		A17	Live	Entity			All
1SG		A02	Live	Entity			All
1/A/1-333 IN		A46, A19, A44, A22	Live	Entity			All
2/A/1-333 IN							
M2A4		A32	Live (SUT)	Entity			
M2A3		A14 856931	Live Constructive	Entity	OneSAF	OTC XLCC	T+0 - T+2:30 T+2:30 - T+4:20
M2A3	Movement Gunner Comms	476201 N/A N/A	Virtual	Entity	DriveTrainer VirtT CommSim	ATC ATC ATC	All
M2A3		A09 488010	Live Constructive	Entity	OneSAF	OTC XLCC	T+0 - T+2:30 T+2:30 - T+4:20
3/A/1-333 IN		A35, A61, A53, A12	Live	Entity			All
B/1-333 IN		846263	Constructive	Platoon	OneSAF	ATC	All

The LVC Implementation Tables are four tables that identify how the entities and effects are implemented within LVC components. The process of developing the tables starts with identifying all scenario elements in the Test Scenario that will be represented, including systems and subsystems, as well as any key effects, and designating them with a unique identifier. For each system and relevant subsystem, the test team determines whether it will be represented as live, virtual, constructive, or digital twin, keeping in mind that its representation may change over the duration of the scenario. Time is recorded via reference to the applicable frames. The team also determines the level of aggregation at which the system must be represented (e.g., entity, platoon, company, etc.). Implementers then identify the application(s) required for the representation (e.g., simulation, stimulation, etc.), as well as the instance, version, and terrain database used. The resulting set of tables consists of the scenario elements (shown in Table 2), scenario effects, application versions, and terrain databases.

The Conditions Implementation Table is a complement to the LVC Implementation Tables that associates key conditions identified in Test Scenario sub-frames with any specific conditional requirements that must be achieved to successfully collect the required data. It requires information from the LVC Implementation Tables, which therefore must be done either first or at the same time. Recall that conditions may or may not be specified by DOE requirements. The test team will begin by identifying and fully describing the conditions that must occur. Those descriptions should come directly from the Test Scenario development process. Then, for each set of conditions, the team will capture:

- the timing by identifying the relevant Test Scenario sub-frame(s) during which they will occur;
- the entities and effects that must be involved in achieving the required conditions; and,
- how the conditions are implemented by scenario element, e.g., specifying entity orientation, speed, or activity, or identifying systems or subsystems that must be activated or deactivated.

See Table 3 for an example of the Conditions Implementation Table.

Table 3. Example Conditions Implementation Table

Conditions Implementation Table								
Frame/ Sub-frame	Conditions	Scenario Elements	Subsystem	Unique ID	LVC Rep	Rep Level	Member App	Implementation
T + 2:30/D4A	- SUT must have line of sight to both threat BMPs - BMP threat jamming system must be active - SUT must execute a call for fire for DIVARTY 155s	2/A/1-333 IN						
		M2A4		A32	Live (SUT)	Entity		- Orients NE towards BMPs at T+2:30 - Executes CFF on BMPs
		Threat BMP #1		Z632	Live	Entity	VizMod	Jamming system remains activated
		Threat BMP #2		Z618	Live	Entity	VizMod	

The last component of the Implemented Scenario includes the LVC implementations themselves. These are the actual instantiations of the scenario in the simulations, stimulations, and other software required for event execution, saved in whatever format is required for the software. The creation of the Implemented Scenario is complex and likely involves interplay between the planners of the Test Scenario and implementers and modelers building the Implemented Scenario. The latter may discover that some aspects of the Test Scenario cannot be modeled in simulation as described and may need to make changes. If those changes break a traceability linkage between the Test Scenario sub-frames and the Implemented Scenario, the team must assess the impact of the changes and potentially go back and update the frames/sub-frames or other parts of the plan. The LVC implementations also include the unit plans (e.g., OPORDs) that drive the live and virtual portions of the scenario. Live and virtual component planners should be involved early when possible, and the test team must account for the likelihood that the live and virtual products may lack the detail that the constructive components will have.

The scenario development methods discussed for both Test and Implemented Scenarios could also be used for training. Using Army training doctrine again as an example, the T&EOs explicitly define the conditions under which the training must be conducted and the required LVC training environment. The use of Test Scenario frames in combination with Implementation Scenario tables could help synchronize the LVC training environment components. Sub-frames would be valuable for ensuring the necessary conditions occur and that specific elements of the scenario are traced back to the performance measures that must be evaluated, along with the required data elements. While the intent of training differs from that of testing, there is significant overlap in planning and execution, particularly for complex LVC environments, where our methodology would benefit both types of events.

CHALLENGES

Of the many challenges we faced in trying to develop a general methodology for maintaining data traceability in the complex MDO LVC modeling environment, there are two that we describe here.

First is satisfying a self-imposed requirement to develop a methodology that does not need specialized software to implement. We wanted our methods and tools to be widely applicable and usable in a variety of connected and disconnected environments and not constrained to software implementations. However, the complexity of the environment makes this difficult. The nature of data traceability and the required linkages across test planning products and processes lend themselves to relational or graph database implementations to maintain those relationships. Without such databases, teams must use unique identifiers and tables to crosswalk the various planning components, which is tedious. Nonetheless, our requirement was that it be achievable, not necessarily easy. We discuss that implementation challenge next.

The second challenge is the inherent complexity of implementing the methodology. As just discussed, the creation of the products can be time-consuming and tedious, particularly if done manually and without automation. The larger the event and number of participating personnel, systems, and LVC implementations, the more complex the relationships and thus the effort required to capture them. Ultimately if we create a methodology that is too cumbersome, no one will use it. We approached this challenge by providing flexibility that allows the user to make trade-offs between the effort rendered and value gained. For example, in creating Test Scenarios, the user can choose to create fewer frames, which in turn may increase the risk to the synchronization of the LVC components of the Implemented Scenario. Or, if DOE is the test team's primary concern, they may choose to link only DOE combinations at the sub-frame level and not data elements or other sets of conditions. Such a decision greatly simplifies the sub-frame method, but it comes at a cost. The reduced traceability significantly decreases the test team's ability to characterize the impact of instrumentation failures to test data requirements. Such decisions are the purview of the test team lead. Our methodology provides that flexibility without sacrificing the rigor required to fully capture data traceability in a complex MDO LVC environment.

FUTURE WORK

Although we have made great progress, our methodology is incomplete. There are two main areas that we are currently addressing. The first is data collection asset (DCA) planning. DCA planning involves not only arraying, configuring, and orienting DCAs to collect the required data, but also establishing traceability of those assets to the scenarios and analytic requirements. The former ensures that the data collection plan accounts for all data requirements and scenario conditions. The latter ensures that the test team can rapidly understand the impact of any changes to the DCA plan during planning or disruptions to it during event execution. This is particularly difficult during execution. Consider a scenario where a DCA fails during test execution. The following questions need to be answered to make a fully informed decision:

- Is this DCA currently collecting required data? If not, will it be needed to collect data before it can be fixed?
- Is there another DCA also collecting the required data?
- Did the time of failure overlap the occurrence of scenario conditions for which data must be collected?
- Will the event generate enough data without the missing data from the failure?
- Is the missing data significant enough to justify altering the event (e.g., restarting or extending the event)?

The methodology must fulfill the information requirements necessary to answer those questions. The DCA methodology must also include the means to record and track data provenance and lineage from data collection to analysis and evaluation.

The second area we intend to address is event scheduling. The purpose is to establish traceability between the Test Scenario which is based on relative time (e.g., H+1:30 or T+0:45); the schedule of events which is planned using date-time groups (e.g., 100945 SEP 26); and real-time event monitoring which captures when events actually occur. The test team must be able to correlate all these times to understand what is happening versus what was planned and to understand impacts of disruptions, such as weather delays, to the event.

Once we have a methodology that addresses data traceability throughout the event design, plan, prepare, execute, and assess phases, we will conduct a culminating table-top exercise (TTX) to put the methodology through its paces using a notional operational test event. We will leverage the TTX process to determine the sufficiency of our methodology and tools, identify gaps and challenges, and prioritize future work. We plan to complete the methodology by June of 2026 with the TTX following soon thereafter.

In parallel with our development of the methodology, we are building automated tools and enhancing existing ones to incorporate the methods we have created and to ease the burden of implementing them. We also plan to extend our methodology and tools beyond test to other domains such as training and experimentation. While there is nothing inherent in the processes we have developed that would prevent their application to those domains, we recognize that they would benefit significantly from a dedicated effort to tailor them for such applications.

CONCLUSION

The primary objective of a T&E event is to gather data to evaluate a SUT. Guaranteeing that the right data is collected within complex and sophisticated MDO LVC test environments is becoming increasingly difficult. Challenges arise from not only the intricate nature of the implementations, but also the involvement of numerous planning specialists who must be aware of, and pass on, data requirements. Deviations in planning or unforeseen issues, such as instrumentation failures or adverse weather, require mechanisms to evaluate their impact on data collection and determine if changes are necessary. Failure to effectively manage these aspects can lead to inadequate data, necessitating additional resources and potentially delaying acquisition programs.

Our contribution is a comprehensive methodology for maintaining data traceability throughout LVC event planning and execution. Our approach provides a structure for what are often unstructured planning products, introducing the necessary level of detail to define and maintain the connections between planning elements. For analytic requirements, this entails decomposing test objectives into individual data and experimental design elements. For scenario development, this involves organizing Test Scenarios into a sequence of frames, which are then further decomposed into sub-frames that establish the detailed linkages to other parts of the test plan. Implemented Scenarios then capture the actual implementation details of the Test Scenarios in the LVC environment.

Our approach is not without its challenges. The effort required to manage inherently complex relationships can be tedious and time-consuming, particularly without specialized software. Nonetheless, the flexibility offered by our methodology allows for significant adaptability, enabling users to balance effort and value effectively. Future work will focus on integrating DCA planning and event scheduling into the methodology, culminating in a TTX to evaluate our methods and tools and identify areas of improvement. We also plan to extend beyond T&E applications to training and experimentation.

REFERENCES

Alvarado, J. L., Jr., & Bradley, T. H. (2024). *A case study-based assessment of a model-driven testing methodology for applicability and metrics of model reuse*. International Test and Evaluation Association (ITEA) Journal, 45(4). https://itea.org/journals/volume-45-4/a-case-study-based-assessment.

Braddock, Dunn & McDonald, Inc. (1974). *Standard procedures for Air Force operational test and evaluation*. Defense Technical Information Center. https://apps.dtic.mil/sti/tr/pdf/ADB000365.pdf.

DAMA International. (2021). *DAMA-DMBOK: Data management body of knowledge*. Bradley Beach, NJ: Technics Publications.

Department of the Army. (2003). *Test and evaluation in support of systems acquisition (DA Pam 73-1)*. https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN40764-PAM_73-1-001-WEB-2.pdf.

Department of the Army. (2021). *Training (FM 7-0)*. https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN35076-FM_7-0-000-WEB-1.pdf.

Institute of Electrical and Electronics Engineers (IEEE). (2010). *IEEE recommended practice for distributed simulation engineering and execution process (DSEEP)*. IEEE Standard 1730-2010. https://standards.ieee.org/ieee/1730/4280/.

JMP Statistical Discovery, LLC. (2025). *Design of experiments*. https://www.jmp.com/en/statistics-knowledge-portal/what-is-design-of-experiments.

Joint Chiefs of Staff. (2020). *Joint planning (JP 5-0)*. https://jdeis.js.mil/jdeis/new_pubs/jp5_0ch1.pdf (access required).

Mucci, T. (2024). *What is data provenance?* IBM. https://www.ibm.com/think/topics/data-provenance.

Operational Test and Evaluation Force. (2020). *Operational test planning handbook*. https://www.cotf.navy.mil/wp-content/uploads/2023/02/Operational-Test-Planning-Handbook_v1_11MAY2020.pdf (access required).

Operational Test and Evaluation Force. (2021). *Mission-based test design checklist*. https://www.optevfor.navy.mil/Portals/103/MBTD-IEF-Checklist-4_0.pdf (access required).

Pappu, N. (2024). *Understand data context: Enhancing value and usability*. Zendata. https://www.zendata.dev/post/understand-data-context-enhancing-value-and-usability.

Zinser, R., LeSueur, K., Turner, T., Furr, J., Goerger, S., & Lerz, E. (2024). *A process for distributed LVC integration and execution*. 2024 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC).