*ITEA Journal* 2011; 32: 479–488


# Multi-Relationship Evaluation Design: Formalizing Evaluation-Design Input and Output Blueprint Elements for Testing Developing Intelligent Systems

**Brian A. Weiss**
National Institute of Standards and Technology, Gaithersburg, Maryland
**Linda C. Schmidt, Ph.D.**
University of Maryland, College Park, Maryland

*Intelligent technologies within the military, law-enforcement, and homeland-security fields are continuously evolving. Testing these technologies is crucial to (a) inform the technology developers of specific aspects for enhancement, (b) request end-user feedback, and (c) verify the degree of the technology's capabilities. Test exercises provide valuable data that both update the state of the technology and present information to the evaluation design team to aid further testing. Evaluation designers have exerted substantial effort in creating methodologies to streamline the test-plan development process. This is particularly evident when producing comprehensive test plans. The Multi-Relationship Evaluation Design (MRED) methodology is being developed to collect input from several source categories and automatically output evaluation blueprints that identify pertinent test characteristics. MRED captures input from three specific categories: personnel stakeholders, the technology state, and the available resources. This information and the relationships among these inputs are merged to feed an algorithm that will output specific test-plan elements. This article will propose a model of developing a technology's state and its influence on the MRED-output. MRED defines the input technology-state category to include the maturity, reliability, and repeatability of a technology under test. The states of these three characteristics evolve as a technology is developed from the conceptual stage to a fully functional system. Likewise, test characteristics evolve to capture the most pertinent data to enhance this development process. In order to ensure that the appropriate test designs are generated, it is critical to understand the relationships between these input and output elements. These relationships are also described in this article. Future efforts will describe and formalize the entire MRED model as relationships are further investigated between all of the inputs and the test-plan output elements.*

**Key words:** Appropriate data; components; input elements; intelligent systems; maturity; metrics; output elements; reliability repeatability; test planning.

Intelligent technologies are continuously being developed for use in military domains, law-enforcement situations, and first-response incidents. These technologies are distinguished by their interactions with human operators and/or robotic elements to achieve specific goals. Assessing these technologies is crucial to update the system creators during the development process and validate the performance of the final systems (Weiss et al. 2010).

Most intelligent technologies are designed by or for the government. It is common for the government to fund these developmental programs on multiyear schedules. These programs are distinct from commercial product-development efforts in that the government organizes its programs in several phases. Each phase usually consists of one or more prescribed test events evaluating technologies created by one or more development teams. It is common for the technology-development and evaluation-design processes to be entwined.

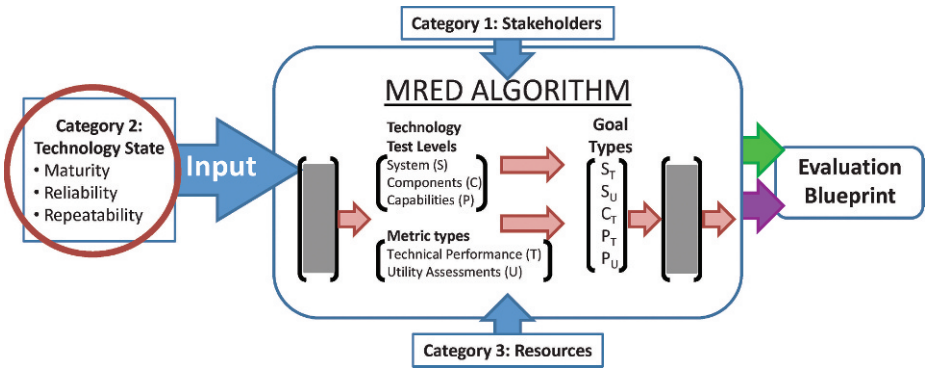


*Figure 1. MRED model including inputs and outputs.*

Both private and government organizations have expended a considerable amount of effort on the research and development of methods and frameworks to effectively and thoroughly evaluate the capabilities of intelligent technologies. Many of these customized test-design methods have been adequate to evaluate precise technologies and accomplish project-specific objectives. No single method has been recognized as being capable of evaluating quantitative and qualitative performance across a range of prototype and physical technologies, encompassing both human-controlled and autonomous capabilities. Test design can be an arduous and challenging process due to technology complexity. Evaluation designers also face another obstacle in that the test-planning activities are prepared manually, where modifications to the unknown and known information may require them to redesign their test exercises. Many of these test methodologies have been presented in prior work (Weiss et al. 2010; Weiss and Schmidt 2010a; Weiss and Schmidt 2010b). The authors have designed the Multi-Relationship Evaluation Design (MRED) methodology to address these shortcomings. Specifically, the MRED methodology is being created to take multiple inputs from numerous input source categories and automatically output evaluation test plans (also called blueprints).

Technology-state characteristics are challenging to capture in any test-design process, and modeling their influence on test plans is critical to MRED's success. In this article, the MRED model will be presented; detailed definitions and relevant relationships of the technology-state input category will be discussed; the output test-plan elements of technology test levels, metrics, and test environments will be defined and their constraints presented; the technology state's influence on determining the technology test levels and test environments will be discussed; and an example of this cause-and-effect relationship will be highlighted in example test plans for a robot arm.

## Multi-Relationship Evaluation Design (MRED)

MRED's objective is to automatically generate evaluation test plans based upon multiple inputs (Weiss and Schmidt 2011). The MRED methodology will take information from three input categories and output one or more evaluation blueprints complete with their own specific test-plan elements. MRED will also characterize the relationships among inputs and the influences inputs have on outputs.

The MRED methodology model describes the important design inputs into the planner and the output evaluation blueprint (Weiss et al. 2010; Weiss and Schmidt 2010a; Weiss and Schmidt 2010b). *Figure 1* presents the overall MRED model. The MRED algorithm will operate on the categories' inputs to generate appropriate evaluation blueprints. The technology-state input category is a main focus of this article. Likewise, the output evaluation blueprint elements of technology test levels, metric types, and goal types are discussed in detail, so they are centrally highlighted, as well.

### *Input categories*

*Stakeholders.* Stakeholders are classified into six categories of parties interested in a technology's evaluation. Members of these categories have their own motivation for the test-plan results of a technology's performance. Their individual motivations will reflect personal uncertainties manifesting in test-design preferences. The six stakeholder categories are buyers, users and potential users, evaluation designers, evaluators, sponsors and funding sources, and technology developers. These categories are listed in *Table 1* (see Weiss and Schmidt 2011 for more detail).

*Technology state.* Three factors are selected to describe the technology's anticipated state of development at the time of its test. These factors are presented in *Table 2* and discussed in greater detail later.

Table 1. Stakeholders.

| Stakeholder group | Who they are |
|---|---|
| *Buyers* | Stakeholders purchasing the technology |
| *Users and Potential Users* | Stakeholders that will be or are already using the technology |
| *Evaluation Designers* | Stakeholders creating the test plans by determining MRED inputs |
| *Evaluators* | Stakeholders implementing the evaluation test plans |
| *Sponsors and Funding Sources* | Stakeholders paying for the technology development and/or evaluation |
| *Technology Developers* | Stakeholders designing and building the technology |

*Resources.* The final input group is comprised of specific types of material, manpower, and technology to be included in the testing exercise. Resource availability (or lack thereof) and limitations can have a significant impact on the final evaluation design. These categories are shown in *Table 3*.

## Output elements

This section presents the output evaluation-blueprint elements that have been specified to date. Technology test levels and test environments are briefly described here and elaborated upon in greater detail in the following sections.

*Technology test levels.* A system (often called a "technology") is made up of constituent components representing a physical hierarchy or set of levels. Likewise, the system's overall performance is made up of constituent capabilities representing a functional hierarchy or set of levels. There are several terms related to these technology test levels:

- *System:* a group of cooperative or interdependent components forming an integrated whole to accomplish a specific goal;
- *Component:* an essential part or feature of a system that contributes to the system's ability to accomplish a goal; and
- *Capability:* a specific ability of a technology, enabled by either a single component or multiple components working together. A system is made up of one or more capabilities.

*Test environments.* The setting in which the evaluation occurs, the test environment, can influence the behavior of the personnel and limit the ability to test technology at certain levels of maturity. MRED defines three distinct environments:

- *Lab:* a controlled environment where test variables and parameters can be isolated and manipulated to determine how they impact system performance and/or the users' perception of the technology's utility;
- *Simulated:* an environment outside of the lab that is less controlled and limits the evaluation team's ability to control influencing variables and parameters, since it tests the technology in a more realistic venue; and
- *Actual:* the domain of operations in which the system is designed to be used. The evaluation team is limited in the data they can collect, since they cannot control environmental variables.

*Other blueprint elements.* Metrics and goal types are two of the remaining outputs from the MRED methodology.

*Metrics*, or measures, are performance indicators that can be observed, examined, detected, or perceived either manually or automatically. In turn, metrics are the result of the analysis of one or more output measures (Weiss et al. 2010). Specifically, there are two types of metrics:

- *Technical performance:* metrics related to quantitative factors (such as accuracy, precision, time, distance, etc.); and
- *Utility assessments:* metrics related to the qualitative factors that express the condition or status of being useful and usable to the target user population.[1]

*Goal types* are a dependent variable determined by combinations of technology test levels and desired metrics. There are five goal types that are output from the MRED framework, listed in *Table 4*.

It is important to note that utility assessments cannot be captured in component evaluations. This is

Table 2. Technology state factors.

| Factor | Definition |
|---|---|
| *Maturity* | Technology's state or quality of being fully developed |
| *Reliability* | Technology's ability to perform a required function under stated conditions for a specified period of time |
| *Repeatability* | Technology's ability to yield the same or comparable results as in previous test(s) |

Table 3. Resources of testing and analysis.

| Resource | Description |
|---|---|
| *Personnel* | Individuals that will use the technology, those that will indirectly interact with the technology, those that will collect data during the test, and those that will analyze the data following the test(s) |
| *Test Environment* | The physical venue, supporting infrastructure, artifacts, and props that will support the test(s) |
| *Data-Collection Tools* | The tools, equipment, and technology that will collect quantitative and/or qualitative data during the test(s) |
| *Data-Analysis Tools* | The tools, equipment, and technology capable of producing the necessary metrics from the collected evaluation data |

because components are defined as parts that technology users are unable to engage or interact with during realistic operations. The remaining output evaluation-blueprint elements are presented in *Table 5*.

## Input category: technology-state factors

The technology-state factors are described by three elements: *maturity, reliability, and repeatability*. These three factors must be known (as much as possible) and understood with respect to a given technology to design an effective test plan for that specific technology. A technology's design and construction include that of its components. As components are integrated together, they enable specific capabilities. Some of the technology's capabilities may be operational before the entire system is fully functional. Throughout the technology's development cycle, its maturity, reliability, and repeatability are constantly evolving. For instance, if several components have a nonfunctional maturity, then they cannot be tested. But if the components are functional, yet not fully functional, then it is likely that limited testing can occur.

### Component and capability relationships

All intelligent systems are composed of components that are integrated to enable a system to perform one or more capabilities. For example, suppose the system to be tested is an intelligent Cartesian robotic arm (these types of control movements are similar to those of a human using multiple joints in harmony to reach for a cup). This specific example features an arm that is composed of six joints (a combination of revolute and prismatic joints) and an end-effector gripper. The entire assembled arm is considered the system. Further, each of the six joints and the gripper are considered components. The capabilities in this instance would be the *x*-, *y*-, and *z*-translations of the gripper; the roll, pitch, and yaw of the gripper; and the grasping of the gripper.

Table 4. Goal types.

| |
|---|
| *Component-Level Testing—Technical Performance* |
| *Capability-Level Testing—Technical Performance* |
| *System-Level Testing—Technical Performance* |
| *Capability-Level Testing—Utility Assessment* |
| *System-Level Testing—Utility Assessment* |

A distinction critical to this work is that technology end users interact with capabilities, not components. This means that the users are focused on the success of the motions (*x*, *y*, *z*, roll, pitch, yaw, and grasping) of the robotic arm, which are its capabilities—not on any of the components (i.e., prismatic joints, revolute joints, and gripper). To simplify the presentation of this example, the links between the joints and other common elements (drive motors, base, etc.) of the robotic arm are not considered.

### Maturity

Maturity must be input into MRED for a technology test level to be considered for testing. The maturity level can be for the system (i.e., the overall technology) and for each individual capability and component that are to be tested. At any time during development, the maturity of the system, its components, and its capabilities will fall into one of the following classes:

- *Nonfunctional:* The technology test level being tested has yet to be developed or is in the process of being developed, so that it is not functional and therefore cannot be tested.
- *Functional:* The technology test level being tested is developed to the point of being functional, yet is not complete (still requires additional development).
- *Fully developed:* The technology test level is developed to the point of being functional and complete.

Maturity data are gathered from the technology developers. These stakeholders are in the best position to provide these data, since they are most familiar with the technology and are likely to have the most up-to-date information.

### Reliability

Like maturity, reliability is defined for the system and the individual components and capabilities that are to be tested. Reliability is the probability that a portion of the items will survive under certain conditions for a certain time. Reliability will be represented as either "No Data" (if data have never been collected) or a

*Table 5. Other evaluation blueprint elements.*

| Other element | Definition |
|---|---|
| ***Personnel—Evaluation Members*** | Various individuals and groups are required to perform an effective evaluation. They are classified into two categories: primary (direct interaction) technology users and secondary (indirect interaction or evaluation support). The primary technology users are defined as Tech Users. These individuals directly interact with the technology during the evaluation. Secondary personnel are those that indirectly interact with the technology during the evaluation. This includes Team Members and Participants. Both primary and secondary personnel are discussed in greater detail in the following sections as their selection relates back to the Stakeholders' preferences. |
| ***Evaluation Scenarios*** | The Evaluation Scenarios govern exactly what the technology users will encounter during the test and the challenges within the identified Test Environments. Three types of Evaluation Scenarios are Technology-based, Task/Activity-based, and Environment-based. |
| ***Explicit Environmental Factors*** | The Explicit Environmental Factors are characteristics within the environment that impact the technology and therefore influence the outcome of the evaluation. These factors pertain to the overall physical space, which is composed of Participants, structures, and any integrated props and artifacts. These factors are broken down into two characteristics, Feature Density and Feature Complexity. Together, these two elements determine the Overall Complexity of the environment. |
| ***Data Collection Methods*** | Data Collection Methods are used to capture experimental and ground truth data, depending upon the technology being evaluated and the specified Test Environment. No matter the type of tools used, Data Collection Methods are characterized by factors that influence the techniques being employed. |
| ***Personnel—Evaluators*** | There are three classes of evaluation personnel that are necessary to ensure that the evaluation proceeds according to plan and that the necessary data are captured to evaluate a technology's performance. They fall into the three classes of Evaluators: Data Collectors, Evaluators: Test Executors, and Evaluators: Safety Officers. |
| ***Data Analysis Methods*** | The Data Analysis Methods blueprint element will be a dependent variable that is specified based upon other blueprint elements, including Data Collection Methods and Metrics. These methods are specific to the technology under test and the available resources, and are therefore not specified in greater detail. |

numerical value ranging from 0% to 100%. Reliability data are collected from an independent third party, which could be the evaluators or evaluation designers.

Depending upon the prior test data that are known and provided, reliability data will be either directly assessed from quantitative data or extracted from qualitative data. For example, quantitative reliability data can be captured from technical performance evaluations relating to either system- or component-level tests. These are usually represented as a percentage. Qualitative reliability data are captured from utility assessment evaluations completed for either system- or capability-level tests. These data are usually represented on a scale signifying an average perception from test subjects. An example qualitative scale would be 1 = very unreliable, 2 = unreliable, 3 = marginally reliable, 4 = reliable, 5 = very reliable. It would be the evaluation designers' responsibility to correlate the qualitative reliability data to the numerical range of 0% to 100%.

It is important to note that reliability of a system cannot simply be strictly calculated by using component or capability reliability test data. This statement is justified by the following principles:

- "The sum is greater than the parts." Just because components and/or capabilities perform at various reliabilities when individually tested does not mean that they will perform at an aggregated reliability when the entire system is tested.
- "The parts can be greater than the sum." A test subject may have a stronger opinion of a technology in tests that allow a focus on specific capabilities, as compared to tests where the subject is forced to select among or operate multiple capabilities within a system. For example, a test subject could be easily overwhelmed when provided multiple capabilities to employ, as compared to being given a single capability to use.
- "Tests are unique." Component and capability tests, which isolate individual elements, are typically unique in comparison to system tests, where multiple components and capabilities are tested in parallel.

### Repeatability

Repeatability is defined as a technology's ability to yield the same or comparable results as those in previous tests. A technology's repeatability can be presented similarly to its reliability: Repeatability can be represented as either "No Data" or a range from 0 to 100%. These data are also gathered by an independent third party.

Repeatability conveys different information from reliability and is measured differently. This is seen in that reliability data can be obtained from a single data set, whereas repeatability must be obtained across multiple data sets. The evaluation designer must consider the scope of the technology and tests when

*Table 6. Robotic arm components and capabilities.*

| | Capabilities | | | | | | |
|---|---|---|---|---|---|---|---|
| | Translation | | | Rotation | | | |
| Components | *X* ($P_1$) | *Y* ($P_2$) | *Z* ($P_3$) | Roll ($P_4$) | Pitch ($P_5$) | Yaw ($P_6$) | Grasping ($P_7$) |
| Revolute Joint 1 ($C_1$) | X | X | | | | X | |
| Revolute Joint 2 ($C_2$) | | X | X | | X | | |
| Prismatic Joint 1 ($C_3$) | X | X | X | | | | |
| Revolute Joint 3 ($C_4$) | X | | X | X | | | |
| Prismatic Joint 2 ($C_5$) | X | X | X | | | | |
| Revolute Joint 4 ($C_6$) | | | | X | X | X | |
| Gripper ($C_7$) | | | | | | | X |

determining how many data sets are necessary to adequately state the reliability and repeatability of a component, capability, or entire system. Note that repeatability can be measured for almost any type of metric. The following example highlights maturity and reliability. Repeatability will be addressed in future work.

## Robotic-arm example

The technology-state factors of maturity and reliability are highlighted in the following example featuring the robotic arm introduced previously. The robotic arm to be tested is comprised of the seven primary components (represented as $C_1$ through $C_7$) that produce seven capabilities (represented as $P_1$ through $P_7$), whose relationships are shown in *Table 6*. This matrix can be interpreted in several ways. Individual components can be examined to see which capabilities they contribute. In this case, Revolute Joint 2 ($C_2$) contributes to the capabilities of *y*-translation ($P_2$), *z*-translation ($P_3$), and pitch ($P_5$). Each column of the matrix displays the components necessary to produce a specific capability. For example, yaw ($P_6$) is controlled by Revolute Joint 1 ($C_1$) and Revolute Joint 4 ($C_6$).

Suppose that the seven components of the robotic arm have the various levels of maturity at time *t*, as according to *Table 7*. Note that the maturity levels of these components would be supplied by the technology developers.

This table is split into different regions depending upon the state of the corresponding components and their relationships to the capabilities (capability maturity is dependent upon component maturity).

- Nonfunctional: Grasping, $P_7$, is a nonfunctional capability because its lone component, $C_7$, is nonfunctional.
- Nonfunctional to functional: The rotation motions ($P_4$, $P_5$, and $P_6$) may fall anywhere in the range of nonfunctional to functional capabilities. This is because at least one contributing component, $C_6$, is nonfunctional, while the other contributing components—$C_1$, $C_2$, and $C_4$—are either functional or fully developed. The specific levels of maturity in this instance would be based upon additional queries by MRED of the technology developer.
- Functional: Translations in *x*-, *y*-, and *z*-directions ($P_1$, $P_2$, and $P_3$) are functional capabilities, since their constituent components are either functional ($C_3$, $C_4$, and $C_5$) or fully developed ($C_1$ and $C_2$).
- Fully developed: A capability falling into this category would be impacted by components that

*Table 7. Influence of component maturity on capability maturity at a given time.*

| | | CAPABILITIES | | | | | | |
|---|---|---|---|---|---|---|---|---|
| COMPONENT MATURITY | COMPONENTS | Translation | | | Rotation | | | Grasping |
| | | X ($P_1$) | Y ($P_2$) | Z ($P_3$) | Roll ($P_4$) | Pitch ($P_5$) | Yaw ($P_6$) | ($P_7$) |
| Fully-Developed (FD) | Revolute Joint 1 ($C_1$) | X | X | | | | X | |
| Fully-Developed (FD) | Revolute Joint 2 ($C_2$) | | X | X | | X | | |
| Functional (FN) | Prismatic Joint 1 ($C_3$) | X | X | X | | | | |
| Functional (FN) | Revolute Joint 3 ($C_4$) | X | | X | X | | | |
| Functional (FN) | Prismatic Joint 2 ($C_5$) | X | X | X | | | | |
| Non-Functional (NF) | Revolute Joint 4 ($C_6$) | | | | X | X | X | |
| Non-Functional (NF) | Gripper ($C_7$) | | | | | | | X |
| | CAPABILITY MATURITY | FN | FN | FN | NF to FN | NF to FN | NF to FN | Non-Functional |

*Table 8. Influence of component reliability on capability reliability.*

| COMPONENT RELIABILITY | COMPONENTS | CAPABILITIES | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Translation | | | Rotation | | | Grasping |
| | | X ($P_1$) | Y ($P_2$) | Z ($P_3$) | Roll ($P_4$) | Pitch ($P_5$) | Yaw ($P_6$) | ($P_7$) |
| 99% | Revolute Joint 1 ($C_1$) | X | X | | | | X | |
| 98% | Revolute Joint 2 ($C_2$) | | X | X | | X | | |
| 72% | Prismatic Joint 1 ($C_3$) | X | X | X | | | | |
| 65% | Revolute Joint 3 ($C_4$) | X | | X | X | | | |
| 51% | Prismatic Joint 2 ($C_5$) | X | X | X | | | | |
| 3% | Revolute Joint 4 ($C_6$) | | | | X | X | X | |
| *No Data* | Gripper ($C_7$) | | | | | | | X |
| | CAPABILITY RELIABILITY | 23.6% | 35.6% | 23.4% | 1.95% | 2.94% | 2.97% | *No Data* |

are all fully developed. This example does not contain any capabilities in this category, although there are several components that are fully developed. This is because no single capability is solely influenced by these fully developed components.

Since the system is the sum of its components and capabilities, it is plausible that the system's maturity could range from nonfunctional to functional. The extent of its functionality would also be ascertained from direct queries to the technology developer.

*Table 8* provides an example of component reliability influencing capability reliability. These data assume that capability reliability cannot be measured directly and that it is the product of the reliabilities of those components that influence that specific capability. This means that the reliability of $P_1$ = Reliability($C_1$) × Reliability($C_3$) × Reliability($C_4$) × Reliability($C_5$). The reliabilities of the remaining capabilities would be calculated similarly. If the reliability of a specific capability is available from direct measurement, it is possible this value could differ from that derived by traditional means of calculating system reliability.

Based upon the example information provided in *Table 8*, it is not practical to test any capabilities that are reliant upon component $C_6$, because this component's reliability is so low (indicated by the stated maturity of nonfunctional), as seen in *Table 7*. Some of the capability reliabilities may appear low in this example, yet these could be reasonable data for those technologies that are undergoing constant development.

## Output elements

A majority of the output elements presented in *Figure 1* are influenced by the technology-state factors. A glimpse of this is seen in the earlier section with respect to maturity of the technology test levels. This section will take a deeper look at the relationships among three of the output elements that are impacted by this input category. Specifically, technology test levels and the test environment will be discussed with respect to their influences on one another, and the following section will examine the relationships between them and the technology-state factors. It is important to note that the technology-state factors influence more output elements than these three that are highlighted. Conversely, these two output elements are influenced by more than just the technology-state factors. *Table 9* presents a portion of the overall input-category/output-element relationship matrix.

The input/output relationships presented in this article are highlighted in green in *Table 9*; those highlighted in red are presented extensively elsewhere (Weiss and Schmidt 2011). The remaining relationships will be discussed in future work.

## Technology-state factor influence on technology test levels, metrics, and test environments

The technology-state factors impact the available technology test levels and test environments. Evidence of this is seen in the robot-arm example. Given the maturity of the components, capabilities, and system stated in *Table 7*, it is important to identify those technology test levels that can be tested and those that cannot. The maturity data presented in *Table 7* are reorganized in *Figure 2*. The relationships illustrate that the system's maturity is dependent upon the capabilities' maturity, which in turn is dependent upon the components' maturity.

The information provided in *Figure 2* enables the generation of *Figure 3*, which highlights the varying levels of testing that could be performed on the technology test levels. The availability of technology test-level elements for testing is a single example of the numerous evaluation-blueprint characteristics that MRED would output. This example only shows the influence of maturity data. In reality, the reliability and

*Table 9. A portion of the overall input-category/output-element relationship matrix.*

| | | OUTPUT | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | INPUT | GOAL TYPES | | EVALUATION PERSONNEL | | | TEST ENVIRONMENT | EVALUATION SCENARIOS |
| | | Technology Levels | Metric Types | Tech Users | Team Members | Participants | | |
| CATEGORY 1: STAKEHOLDERS | Buyers | | X | X | X | X | X | X |
| | User, Potential User | | | X | X | X | X | X |
| | Evaluation Designer | X | X | X | X | X | X | X |
| | Evaluation Executor | | | X | X | X | X | X |
| | Sponsor/ Funding Source | X | X | X | X | X | X | X |
| | Technology Developer | X | X | X | X | X | X | X |
| | | | | | | | | |
| CATEGORY 2: TECHNOLOGY STATE | Maturity | X | X | X | | | X | X |
| | Reliability | X | X | X | | | X | X |
| | Repeatability | X | X | | | | X | X |

repeatability data, coupled with stakeholder preferences (e.g., stakeholders only want to test those individual components whose reliability is >70%), have the potential to further delineate which technology test levels should be tested and which should not for a given evaluation.

Relationships involving goal types (combination of technology test levels and metrics) have also been discussed in prior work (Weiss et al. 2010). In summary, the more advanced a technology, the more likely it is capable of operating in an actual environment. Using the robot-arm example, basic tests (at a minimum) should be performed on the individual components to attain a measure of confidence that they will behave as intended when integrated with each other to produce various capabilities and, ultimately, form the entire system. Premature integration can lead to catastrophic failure of multiple components, resulting in unnecessary financial and time loss. It is probable that component testing would take place in a controlled lab environment where a specific input can be produced and component-specific output data are measured. It is not practical (or plausible) to isolate and test an individual joint in a factory setting (i.e., simulated environment) or on a busy assembly line (i.e., actual environment). Advanced testing of the entire system can be performed when the technology is more fully developed. However, it is virtually impossible to isolate a component during system-level testing.

Based upon the information provided in *Figure 3*, it is reasonable to state that MRED would output test plans that call for testing in the lab and/or simulated environment. The actual environment would be a premature test venue, given that the system and several components are nonfunctional at this time. The simulated environment could be a reasonable option, given that several components are either fully developed or fully functional. The lab environment would be a preferred venue to examine individual capabilities and components to isolate specific behaviors and control specific test variables. Of course, stakeholder preferences (discussed in Weiss and Schmidt 2011) and resources (to be presented at a later date) influence the selection of the environment(s).

## Conclusions and future work

The simple robot-arm example illustrates MRED's broad potential to be applied to the evaluation design of complex commercial systems. MRED's development

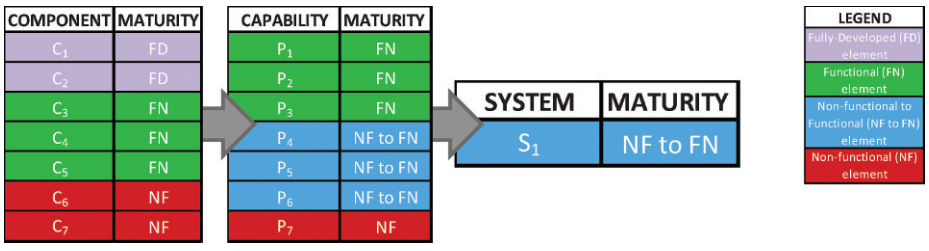


*Figure 2. Maturity of the robotic-arm technology test levels.*

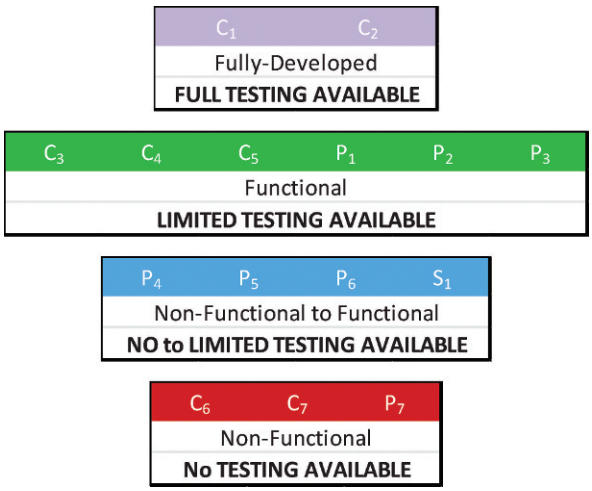


*Figure 3. Technology test levels available for testing.*

has also been supported by other test efforts, including those sponsored by the government. The National Institute of Standards and Technology and members of the Army Research Laboratory's Collaborative Technology Alliance have collaborated to design and execute evaluations to test multiple pedestrian-tracking algorithms (Bodt et al. 2009). The joint team worked together from 2007 through 2010 to plan and implement numerous test events. This work was used as an example in earlier reporting on the development of MRED (Weiss et al. 2010; Weiss and Schmidt 2010a; Weiss and Schmidt 2010b). The pedestrian-tracking example will continue to be explored using MRED. Upcoming efforts will formalize the relationships between input categories and output evaluation elements. It is anticipated that the expansion of the model shown in *Figure 1* coupled with the input/output relationships shown in *Table 9* will yield a mathematical formalization. This formalization will leverage principles from linear algebra and matrix manipulation to support the development of MRED's driving algorithm.

MRED continues to be defined by detailing the input technology-state factors and their influence on the evaluation-blueprint characteristics of technology test levels and test environment. The robot-arm example will be used to further elaborate upon the metrics and evaluation scenarios along with other MRED output-blueprint elements. Likewise, the input resources category will be explored to see its impact on test blueprints once these data are subsumed into MRED. Further investigation will continue in examining the input categories and output-blueprint elements to build upon the discussed relationships. Ultimately, MRED's model will be solidified and its algorithm defined so that test plans can be generated, given the necessary input data. This will enable evaluation designers, sponsors, etc., to quickly change their evaluation direction and/or test goals in the face of changing requirements. The rapid emergence of advanced and intelligent systems justifies methodologies such as MRED. It is envisioned that this automated test-planning methodology will improve the pace of development and delivery of intelligent systems. ❑

*Brian A. Weiss has been a mechanical engineer at the National Institute of Standards and Technology in Maryland since 2002. His focus is the development and implementation of performance metrics to quantify technical performance and assess end-user utility of intelligent systems throughout various stages of development. He has a bachelor of science in mechanical engineering from the University of Maryland and a professional master of engineering from the University of Maryland, and is working towards his doctor of philosophy in mechanical engineering at the University of Maryland. E-mail: brian.weiss@nist.gov*

*Dr. Linda C. Schmidt is an associate professor at the University of Maryland. She holds a doctorate in mechanical engineering from Carnegie Mellon University and bachelor of science and master of science degrees in industrial engineering from Iowa State University. She is active in teaching design research in theory and practice. She has also coauthored texts on engineering decision making and product development. E-mail: lschmidt@umd.edu*

## Endnotes

[1]Utility is defined as the status of being useful and usable to the technology user and is not meant in the economic sense.

## References

Bodt, B., R. Camden, H. Scott, A. S. Jacoff, T. Hong, T. Chang, R. Norcross, T. Downs, and A. Virts. 2009. Performance measurements for evaluating static and dynamic multiple human detection and tracking systems in unstructured environments. In *Proceedings of the 2009 Performance Metrics for Intelligent Systems (PerMIS) Workshop*, September 21–23, 2009, Gaithersburg, MD, eds. Madhaven, R. & Messina, E., 166–173. New York, NY: ACM.

Weiss, B. A., and L. C. Schmidt. 2010a. The Multi-Relationship Evaluation Design framework: Creating evaluation blueprints to assess advanced and intelligent technologies. In *Proceedings of the 2010 Performance Metrics for Intelligent Systems (PerMIS) Workshop*. September 28–30, 2010, Baltimore, MD, eds. Messina, M. and Tunstel, E., New York, NY: ACM.

Weiss, B. A., and L. C. Schmidt. 2010b. The Multi-Relationship Evaluation Design framework: Producing evaluation blueprints to test emerging, advanced, and intelligent systems. In *Proceedings of the 2010 International Test and Evaluation Association (ITEA) Annual Symposium*. September 13–16, 2010, Glendale, AZ.

Weiss, B. A., and L. C. Schmidt. 2011. Multi-Relationship Evaluation Design: Formalizing test plan input and output elements for evaluating developing intelligent systems. Forthcoming. In *Proceedings of the ASME 2011 International Design Engineering Technical Conferences (IDETC)—23rd International Conference on Design Theory and Methodology (DTM)*, August 28–31, 2011, Washington, D.C. New York, NY: ASME.

Weiss, B. A., L. C. Schmidt, H. Scott, and C. I. Schlenoff. 2010. The Multi-Relationship Evaluation Design framework: Designing testing plans to comprehensively assess advanced and intelligent technologies. In *Proceedings of the ASME 2010 International Design Engineering Technical Conferences (IDETC)—22nd International Conference on Design Theory and Methodology (DTM)*, August 15–18, 2010, 603–616, Montreal, Quebec, Canada. New York, NY: ASME.

## Acknowledgments

The authors would like to acknowledge Harry Scott and Craig Schlenoff from the National Institute of Standards and Technology's Intelligent Systems Division for their continued support.