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
100 Bureau Drive MS 8230
Gaithersburg, Maryland 20899 USA
brian.weiss@nist.gov
Phone: 301.975.4373
Fax: 301.990.9688

Linda C. Schmidt
University of Maryland
0162 Glenn L. Martin Hall, Building 088
College Park, Maryland 20742-3035
lschmidt@umd.edu
Phone: 301.405.0417
Fax: 301.314.9477

Abstract

Intelligent technologies within the military, law enforcement, and homeland security fields are continually evolving. Testing these technologies is crucial to (1) inform the technology developers of specific aspects for enhancement, (2) request end-user feedback, and (3) verify 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 including personnel stakeholders, the technology state, and the available resources. This information and the relationships among these inputs are merged to feed an algorithm to output specific test plan elements. This paper 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 conditions 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 will also be described in this paper. 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.

1. Introduction

Intelligent technologies are continually 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 autonomous 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 [2].

Many intelligent technologies are designed by or for the government. It is common for the government to fund these developmental programs on multi-year 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 to evaluate technologies created by one or more development teams. It is common for the technology development and evaluation design processes to be entwined.

Both private and government organizations have expended a considerable amount of effort into 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 to evaluate 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 re-design their test exercises. Many of these test methodologies have been presented in prior work [2] [3] [4]. 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. This paper will discuss the following: the MRED model; detailed definitions and relevant relationships of the technology state input category; the output test plan elements of technology test levels, metrics, and test environments along with their constraints; the technology state's influence on determining the technology test levels and test environments; and an example of this cause and effect relationship will be highlighted in example test plans for a robot arm.

2. Multi-Relationship Evaluation Design (MRED)

MRED's objective is to automatically generate evaluation test plans based upon multiple inputs [5]. 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 that inputs have on outputs.

The MRED methodology model describes the important design inputs into the planner and the output evaluation blueprint [2] [3] [4]. 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 paper. 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.

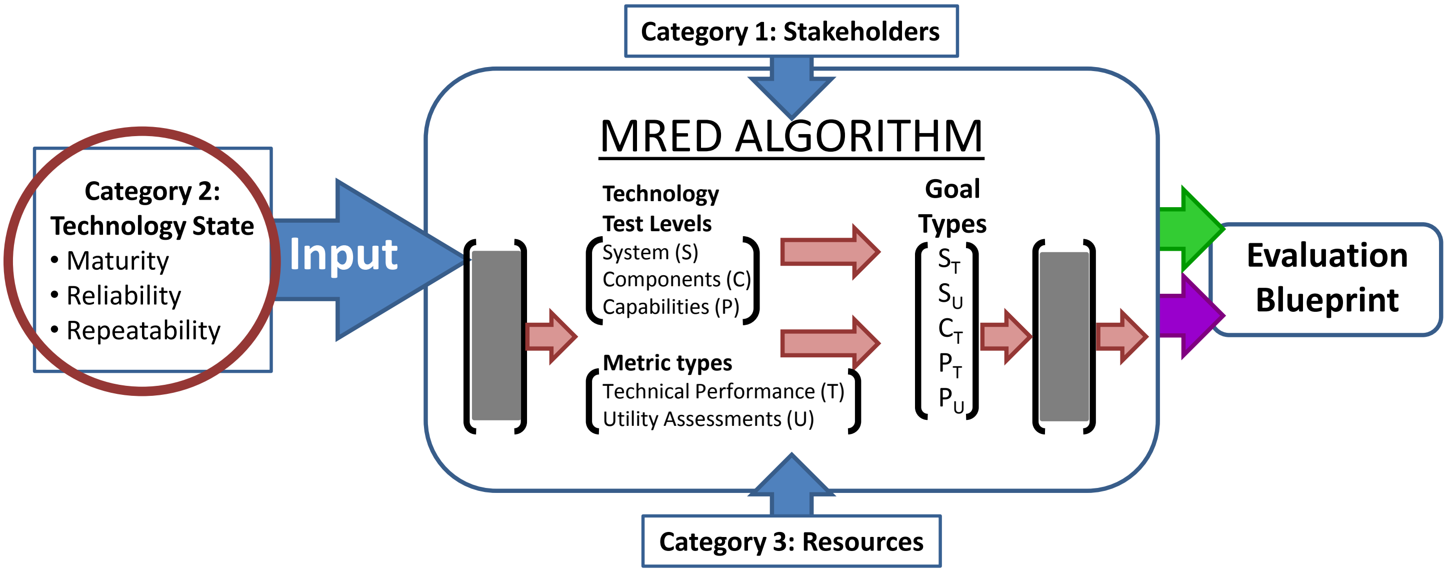


Figure 1: MRED Model including Inputs and Outputs

2.1. Input Categories

2.1.1. Stakeholders

Stakeholders, as shown in the top of Figure 1, 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 [5] for more detail).

Table 1 - Stakeholders

STAKEHOLDER GROUPS	WHO THEY ARE...
Buyers	Stakeholder purchasing the technology
Users, Potential Users	Stakeholder that will be or are already using the technology
Evaluation Designers	Stakeholder creating the test plans by determining MRED inputs
Evaluators	Stakeholder implementing the evaluation test plans
Sponsors/Funding Sources	Stakeholder paying for the technology development and/or evaluation
Technology Developers	Stakeholder designing and building the technology

2.1.2. Technology State

As shown in Figure 1, 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 in Section 3.

Table 2 - Technology State Factors

FACTORS	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 in previous test(s).

2.1.3. 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 their limitations can have a significant impact on the final evaluation design. These categories are shown in Table 3.

Table 3 - Resources for Testing and Analysis

RESOURCES	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.

2.2. 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 below and elaborated upon in greater detail in the following sections.

2.2.1. 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* – Group of cooperative or interdependent *Components* forming an integrated whole to accomplish a specific goal.
- *Component* – Essential part or feature of a *System* that contributes to the System's ability to accomplish a goal(s).
- *Capability* – A specific ability of a technology. A *System* provides one or more *Capabilities*. A *Capability* is enabled by either a single *Component* or multiple *Components* working together.

2.2.2. 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* – 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* – 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.
- *Actual* – Domain of operations that the system is designed to be used. The evaluation team is limited in the data they can collect since they cannot control environmental variables.

2.2.3. Other Blueprint Elements

The elements below constitute the remaining outputs from the MRED methodology. Greater detail can be found in [2] [3] [4] [5].

Metrics: Measures are a performance indicator that can be observed, examined, detected, and/or perceived either manually or automatically. In turn, *Metrics* are the result of the analysis of one or more output measures [2]. Specifically, there are two types of *metrics* listed below:

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

Goal Types: 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.

Table 4 - Goal Types

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

It is important to note that *Utility Assessments* cannot be captured in *Component* evaluations. This is 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.

Table 5 - Other Evaluation Blueprint Elements

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 accordingly to plan and that the necessary data is 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.

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

3. 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 non-functional maturity, then they cannot be tested. Rather if the components are functional, yet not fully-functional, then it's likely that limited testing can occur.

3.1. 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 a human using multiple joints in harmony to reach for a cup). This specific example features an arm that it is composed of six joints (a combination of rotating revolute and actuating 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.

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, grasping) of the robotic arm which are its *Capabilities*, not any of the components (i.e., prismatic joints, revolute joints, 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.

3.2. Maturity

Maturity must be input into MRED for a *Technology Test Level* to be considered for testing. The *Maturity* level could 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:

1) Non-Functional – The *Technology Test Level* being tested has yet to be developed or is in the process of being developed where it's not functional and therefore cannot be tested.
2) Functional – The *Technology Test Level* being tested is developed to the point of being functional, yet is not complete (still requires additional development)
3) Fully-Developed – The *Technology Test Level* is developed to the point of being functional and complete

Maturity data is gathered from the *Technology Developers*. These *Stakeholders* are in the best position to provide this data since they are most familiar with the technology and are likely to have the most up-to-date information.

3.3. 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 either be represented as *No Data* (if data has never been collected) or have a numerical value ranging from 0% to 100%. *Reliability* data is collected from an independent, third party which could be the *Evaluators* or *Evaluation Designers*.

Depending upon the prior test data that is known and provided, *Reliability* data will either be 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. This is usually represented as a percentage. Qualitative *Reliability* data is captured from *Utility Assessment* evaluations completed for either *System* or *Capability* level tests. This data is usually represented on a scale signifying an average perception from test subjects. An example qualitative scale would be 1 – Very Unreliable, 2 – Unreliable, 3 – Marginal Reliability, 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 calculated by using *Component* or *Capability Reliability* test data. This statement is justified by:

- "The sum is greater than the parts" – If *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 them to focus on specific *Capabilities* as compared to tests where they are 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* with which to use.
- Tests are unique – *Component* and *Capability* tests are typically unique from *System* tests where multiple *Components* and *Capabilities* are tested in parallel in the latter as compared to isolating individual elements in the former.

3.4. Repeatability

Repeatability is defined as a technology's ability to yield the same or comparable results as those in previous test(s). A technology's *Repeatability* can also be presented similarly to *Reliability*. *Repeatability* can either be represented as *No Data* or range from 0 % to 100 %. This data is also gathered by an independent, third party.

Repeatability conveys different information than *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 their test(s) when determining how many data sets are necessary to adequately state the *Reliability* and *Repeatability* of a *Component*, *Capability*, or the 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.

3.5. Robotic Arm Example

The *Technology State Factors* of *Maturity* and *Reliability* are highlighted in the following example featuring the robotic arm introduced in Section 3.1. The robotic arm to be tested is comprised of the seven primary *Components* (represented as $C_{1\text{-}7}$) that produce seven *Capabilities* (represented as $P_{1\text{-}7}$) 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).

Table 6 - Robotic Arm Components and Capabilities

	CAPABILITIES						
	Translation			Rotation			Grasping (P_7)
COMPONENTS	X (P_1)	Y (P_2)	Z (P_3)	Roll (P_4)	Pitch (P_5)	Yaw (P_6)	
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**

Suppose that the seven *Components* of the robotic arm have the various levels of *Maturity* at time *t* according to Table 7. Note that the *Maturity* levels of these *Components* would be supplied by the *Technology Developers*.

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 7 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*).

- Non-Functional – Grasping, P_7, is a non-functional *Capability* because its lone *Component*, C_7, is non-functional.
- Non-Functional to Functional – The rotation motions (P_4, P_5, and P_6) may fall anywhere in the range of non-functional to functional *Capabilities*. This is because at least one contributing *Component*, C_6, is non-functional 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, C_5) or fully-developed (C_1, C_2)
- Fully-Developed – A *Capability* falling into this category would be impacted by *Components* that 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's plausible that the *System's Maturity* could range from non-functional 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*. This example data assumes 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, then it's possible this value could differ from that of traditional means of calculating system *Reliability*. For this example, the reliability of P_1 is assumed to be the product of the *Component* reliabilities for simplicity. It's reasonable that individual *Component* reliabilities could be weighted differently from one another which would impact the output *Capability* reliability.

Table 8 - Influence of Component Reliability on Capability Reliability

		CAPABILITIES						
		Translation			Rotation			Grasping
COMPONENT RELIABILITY	***COMPONENTS***	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*

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 non-functional as seen in Table 7). Some of the *Capability Reliabilities* may appear low in this example, yet this could be reasonable data for those technologies that are undergoing constant development. Note that colors are used in Table 8 to enable information to stand-out from that in adjacent cells; color has no other meaning.

4. 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 previous section with respect to *Maturity* on 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 below with respect to their influences on one another, while 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 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.

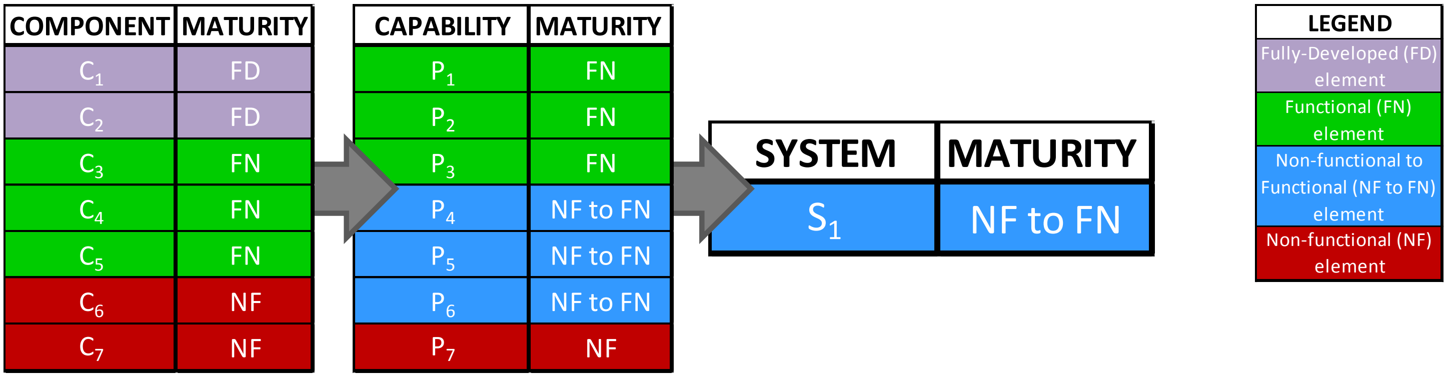
Table 9 - Portion of the Overall Input Category/Output Element Relationship Matrix

		OUTPUT						
		GOAL TYPES		EVALUATION PERSONNEL				
	INPUT	Technology Levels	Metric Types	Tech Users	Team Members	Participants	TEST ENVIRONMENT	EVALUATION SCENARIOS
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

The I/O relationships presented in this paper are highlighted in green in Table 9, while those highlighted red were presented extensively in [5]. The remaining relationships will be discussed in future work.

5. 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 the *System* stated in Table 7, it's important to identify those *Technology Test Levels* that can be tested and those that cannot. The *Maturity* data presented in Table 7 is reorganized in Figure 2 below. The relationships illustrate that the *System's Maturity* is dependent upon the *Capabilities' Maturity* which, in turn, is dependent upon the *Components' Maturity*.



COMPONENT	MATURITY
C_1	FD
C_2	FD
C_3	FN
C_4	FN
C_5	FN
C_6	NF
C_7	NF

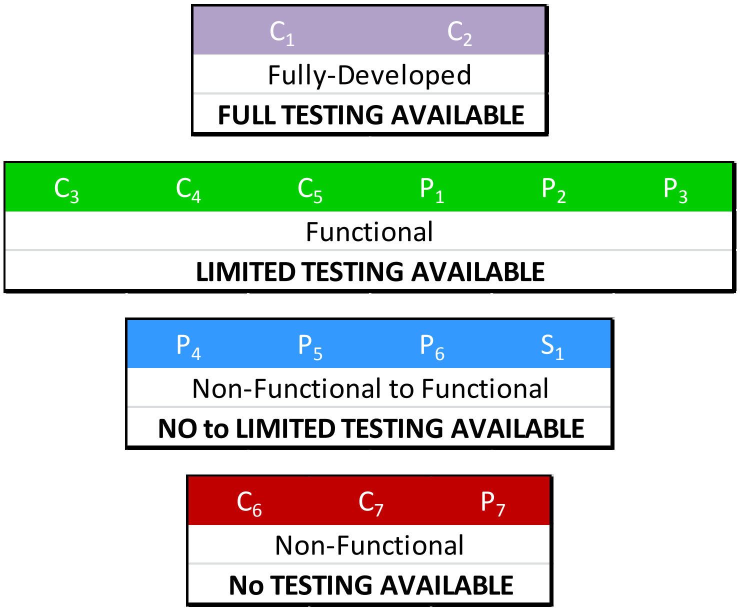
CAPABILITY	MATURITY
P_1	FN
P_2	FN
P_3	FN
P_4	NF to FN
P_5	NF to FN
P_6	NF to FN
P_7	NF

SYSTEM	MATURITY
S_1	NF to FN

LEGEND
Fully-Developed (FD) element
Functional (FN) element
Non-functional to Functional (NF to FN) element
Non-functional (NF) element

Figure 2: Maturity of the Robotic Arm Technology Test Levels

The information provided in Figure 2 enables the generation of Figure 3, shown below. Figure 3 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 *Repeatability* data, coupled with *Stakeholder* preferences (i.e., *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.



C_1	C_2
Fully-Developed	
FULL TESTING AVAILABLE	

C_3	C_4	C_5	P_1	P_2	P_3
Functional					
LIMITED TESTING AVAILABLE					

P_4	P_5	P_6	S_1
Non-Functional to Functional			
NO to LIMITED TESTING AVAILABLE			

C_6	C_7	P_7
Non-Functional		
No TESTING AVAILABLE		

Figure 3: Technology Test Levels Available for Testing

Relationships involving *Goal Types* (combination of *Technology Test Levels* and *Metrics*) have also been discussed in prior work [2]. 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 other 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 is measured. It's 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's 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 non-functional 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 [5]) and *Resources* (to be presented at a later date) influence the selection of the *Environment(s)*.

6. 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 has also been supported by other test efforts including those sponsored by the government. The National Institute of Standards and Technology (NIST) and members of the Army Research Laboratory's (ARL) Collaborative Technology Alliance (CTA) have collaborated to design and execute evaluations to test multiple pedestrian tracking algorithms [1]. The NIST/CTA 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 [2] [3] [4]. 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 this data is subsumed into MRED. Further investigation will continue to examine 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.

7. Acknowledgements

The authors would like to acknowledge Harry Scott and Craig Schlenoff from NIST's Intelligent Systems Division for their continued support.

References

[1] Bodt, B., Camden, R., Scott, H., Jacoff, A.S., Hong, T., Chang, T., Norcross, R., Downs, T., & Virts, A., 2009, "Performance Measurements for Evaluating Static and Dynamic Multiple Human Detection and Tracking Systems in Unstructured Environments," *Proceedings of the 2009 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[2] Weiss, B.A., Schmidt, L.C., Scott, H., and Schlenoff, C.I., 2010, "The Multi-Relationship Evaluation Design Framework: Designing Testing Plans to Comprehensively Assess Advanced and Intelligent Technologies," *Proceedings of the ASME 2010 International Design Engineering Technical Conferences (IDETC) – 22ND International Conference on Design Theory and Methodology (DTM).*

[3] Weiss, B.A. and Schmidt, L.C., 2010, “The Multi-Relationship Evaluation Design Framework: Creating Evaluation Blueprints to Assess Advanced and Intelligent Technologies,” *Proceedings of the 2010 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[4] Weiss, B.A. and Schmidt, L.C., 2010, “The Multi-Relationship Evaluation Design Framework: Producing Evaluation Blueprints to Test Emerging, Advanced, and Intelligent Systems,” *Proceedings of the 2010 International Test and Evaluation Association (ITEA) Annual Symposium*.

[5] Weiss, B.A. and Schmidt, L.C., 2011, “Multi-Relationship Evaluation Design: Formalizing Test Plan Input and Output Elements for Evaluating Developing Intelligent Systems,” To Appear: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences (IDETC) – 23RD International Conference on Design Theory and Methodology (DTM)*.