Multi-Relationship Evaluation Design: Modeling an Automatic Test Plan Generator

Brian A. Weiss
National Institute of Standards and Technology
100 Bureau Drive, MS 8230
Gaithersburg, Maryland 20899
+1 (301) 975-4373
brian.weiss@nist.gov

Linda C. Schmidt
University of Maryland
0162 Glenn L. Martin Hall, Building 088
College Park, Maryland 20742
+1 (301) 405-0417
lschmidt@umd.edu

ABSTRACT

Advanced and intelligent systems within the manufacturing, military, homeland security, and automotive fields are constantly emerging and progressing. Testing these technologies is crucial to (1) inform the technology developers of targeted areas for improvement, (2) capture end-user feedback, and (3) verify the degree of the technology's capabilities. Evaluation designers have put forth considerable effort in developing methods to speed test-plan generation. The Multi-Relationship Evaluation Design (MRED) methodology is being created to gather multiple inputs from several source categories and automatically output evaluation blueprints that identify the pertinent test-plan characteristics. MRED captures input from three categories including the evaluation stakeholders, the technology state, and the available resources. This information and the relationships among these inputs are combined as input into an algorithm that will yield specific test plan characteristics. This paper reviews the MRED methodology as it enters its final stages of development, including new discussion of the relationships among the various inputs and the chosen method of Evaluative Voting to capture *Stakeholder Preferences.* An example focusing on the design of test plans to evaluate a robotic arm is also presented to bring further clarity to the latest MRED developments.

Categories and Subject Descriptors

B.8.0 [**Performance of Systems**]: *measurement techniques, modeling techniques, performance attributes*

General Terms

Measurement, Performance, Design, Experimentation, Verification

Keywords

MRED, performance evaluation, model, test plan design

1. INTRODUCTION

Advanced and intelligent systems within the manufacturing, military, homeland security, and automotive industries are constantly emerging and progressing. Evaluating these technologies is vital to (1) inform the technology developers of targeted areas for improvement, (2) capture end-user feedback, and (3) verify the degree of the technology's capabilities. Evaluation events provide useful data that both update the state of the technology and support future testing. In this paper, the term *test* refers to a planned evaluation event or exercise focused on capturing data to generate performance metrics of a specific technology under scrutiny. Evaluation designers put forth extensive efforts in generating methods to speed the test-plan development process. These efforts are most apparent when designers must create comprehensive test plans to evaluate advanced and intelligent technologies.

The Multi-Relationship Evaluation Design (MRED) methodology will allow evaluation designers to hasten the test-plan development process. MRED gathers multiple inputs from several source categories and automatically outputs evaluation blueprints that identify pertinent test-plan characteristics. MRED captures input from three categories including the evaluation stakeholders, the technology state, and the available resources. This information and the relationships among these inputs are combined as input to an algorithm that will yield specific test plan characteristics.

This paper is organized as follows: Section 2 presents the overall MRED methodology; Section 3 discusses the preference capture method of 'Evaluative Voting' and how it will be implemented with MRED; Section 4 shows an example application of 'Evaluative Voting' integrated into MRED; and Section 5 concludes the discussion.

2. MULTI-RELATIONSHIP EVALUATION DESIGN (MRED) - METHODOLOGY

MRED's goal is to automatically produce evaluation test plans based upon multiple inputs [12]. MRED is an interactive algorithm that processes information from multiple input categories and outputs one or more evaluation blueprints including their constituent test plan elements (Figure 1). During this process MRED invokes the relationships among the inputs and the impacts the inputs have on the outputs. The overall methodology was proposed in [11], while the output blueprint evaluation elements were defined in [9] and [10]. The relationships between specific inputs and outputs were presented in [12] and [13]. This section briefly presents the MRED model inputs (including the *Technology State, Resources,* and *Stakeholder Preferences*) and outputs (including *Technology Test Levels, Metrics, Resources, Evaluation Scenarios,* and *Explicit Environmental Factors*). Greater detail can be found in the afore-mentioned references. The remainder of Section 2 gives an overview of MRED's process and presents new work characterizing the relationships among the various inputs.

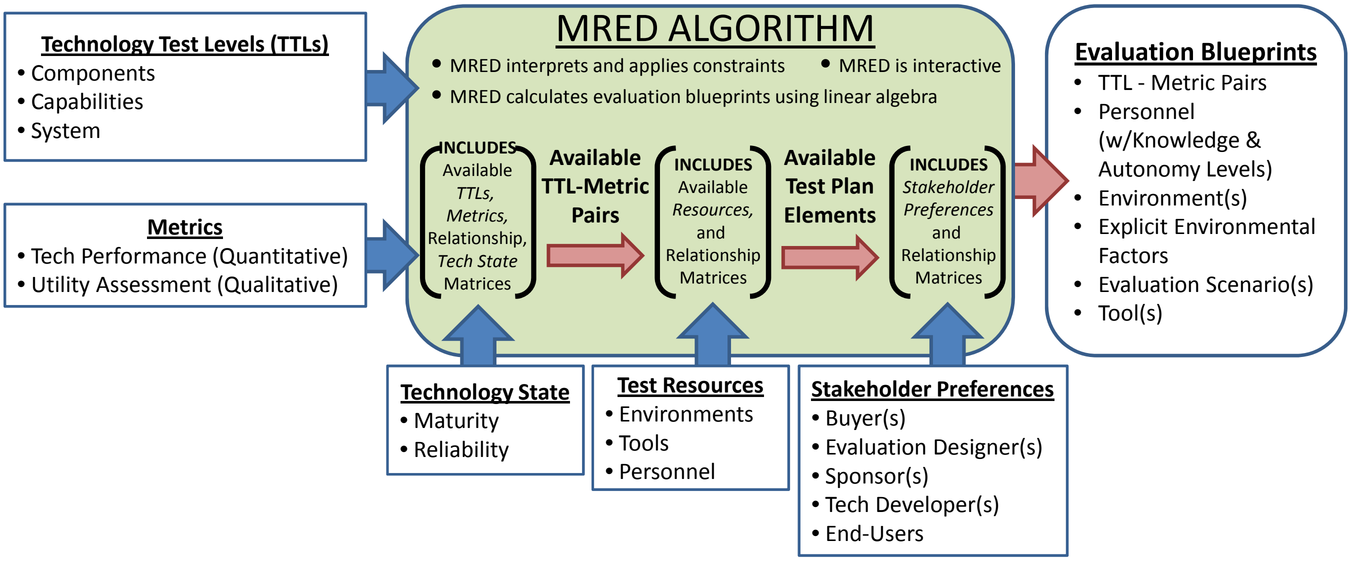

Technology Test Levels (TTLs)
• Components
• Capabilities
• System
Metrics
• Tech Performance (Quantitative)
• Utility Assessment (Qualitative)
MRED ALGORITHM
• MRED interprets and applies constraints
• MRED is interactive
• MRED calculates evaluation blueprints using linear algebra
INCLUDES Available TTLs, Metrics, Relationship, Tech State Matrices
Available TTL-Metric Pairs
INCLUDES Available Resources, and Relationship Matrices
Available Test Plan Elements
INCLUDES Stakeholder Preferences and Relationship Matrices
Evaluation Blueprints
• TTL - Metric Pairs
• Personnel (w/Knowledge & Autonomy Levels)
• Environment(s)
• Explicit Environmental Factors
• Evaluation Scenario(s)
• Tool(s)
Technology State
• Maturity
• Reliability
Test Resources
• Environments
• Tools
• Personnel
Stakeholder Preferences
• Buyer(s)
• Evaluation Designer(s)
• Sponsor(s)
• Tech Developer(s)
• End-Users

Figure 1. MRED Model with Input (TTLs, Metrics, Technology State, Test Resources, Stakeholder Preferences) and Output (Evaluation Blueprints)

2.1 Input

The most significant inputs into the MRED model are the *Technology Test Levels (TTLs)* and corresponding *Metrics*. *TTLs* are defined as the technology's constituent *Components* and *Capabilities* along with the *System* as a whole [9]. Specifically, they can be described as:

- *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* is made up of one or more *Capabilities*. A *Capability* is enabled by either a single *Component* or multiple *Components* working together.
- *System* – Group of cooperative or interdependent *Components* forming an integrated whole to accomplish a specific goal(s).

Pertinent *Metrics* are also input for each *TTL*. *Metrics* fall into one of two groups:

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

Technology State features another set of inputs: *Maturity* and *Reliability* of the individual *TTLs*. In the context of MRED, they are defined as:

- *Maturity* – The state of development of individual *Components*, *Capabilities*, and the *System*. *Maturity of a* technology's *Components* must be provided by the *Technology Developer(s),* whereas *Maturity* of *Capabilities* and the *System* could either be provided by the *Technology Developer(s)* or calculated by MRED given *Component Maturity* and the *Component – Capability* matrix (presented in Section 2.4). *Maturity* information provided by the *Technology Developer(s)* is either classified as *Fully-Developed, Functional,* or *Non-Functional*.
- *Reliability* – The probability that a specific *Component, Capability,* or the *System* (as a whole) will continue to function under certain conditions for a certain time. Similar to *Maturity*, *Component Reliability* must be directly provided by the *Technology Developer(s)* or by the *Evaluation Designer(s)* from prior test efforts. *Reliability* of specific *Capabilities* and the *System* can either be obtained from the *Technology Developer(s)*, the *Evaluation Designer(s)* (also from prior testing), or through MRED calculations using *Component Reliability* and the *Component – Capability* relationship matrix. The nature of the specific *Reliability* measure is dependent upon the technology in question.

Further details on *Technology State* including *Reliability* and *Maturity* can be found in [13].

Test Resources represents the availability of the viable *Environments*, *Personnel*, and *Tools* for data collection and analysis. Discussion of these inputs is presented in [9] and [10].

The last significant input category is that of the *Stakeholder Preferences*. Initially presented in [12], this includes the preferences from five specific individuals (or groups) presented in Table 1. *Stakeholder preferences* are captured with respect to *TTL-Metric* pairs[1], *Environments, Tools, Personnel, Explicit Environmental Factors,* and *Evaluation Scenarios* [10] [11].

Note that colors are used in tables throughout this document to assist the reader in distinguishing data among the rows and columns. Colors do not indicate information of greater or lesser importance.

[1] *TTL-Metric* pairs are specific *Technology Test Levels* and *Metrics* that are coupled together. Multiple *TTLs* can be coupled with the same *Metrics* and vice-versa.

Table 1 – Stakeholders [12]

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

2.2 Output Elements

MRED is designed to automatically output sets of evaluation blueprints complete with specified elements (Figure 1). Each set of blueprints will include one (or more) *TTL-Metric* pairs, an *Environment* for testing, *Tools* to support the capture of data to generate the necessary *Metrics*, *Personnel* including those that will interact with the technology and those that will execute the evaluation, *Knowledge* and *Autonomy Levels* dictating what specific *Personnel* can and cannot do during the evaluation [12], *Evaluation Scenarios* describing the types of exercises that will occur [10], and *Explicit Environmental Factors* which provide guidance as to the level of *Feature Complexity* and *Feature Density* within the *Environment* [10].

2.3 MRED Process

MRED generates the most preferred evaluation blueprints by using an interactive process between:

- Interacting with the *MRED Operator* to collect the necessary information and *Stakeholder Preferences* and
- Processing the collected information and preferences by calculating pertinent *Technology State* information, assessing the feasibility of blueprint elements, generating potential blueprints, and scoring the feasible blueprints.

This multi-step process shown in Figure 2 is summarized below. The term *MRED Operator* is defined as the individual that inputs data, information, and preferences into MRED. This is usually the *Evaluation Designer* or another facilitator who is guiding the blueprint generation process.

1. *MRED Operator* inputs the technology's *TTLs* and corresponding *Metrics* that are considered for testing.
2. *MRED Operator* defines the *Components-Capabilities* and *Metrics-TTLs* relationship matrices.
3. *MRED Operator* inputs *Component Tech. State* data.
4. *MRED* calculates the *Technology State* data for the *Capabilities* and the *System*.
5. *MRED* eliminates *TTLs* and *Metrics* based upon the *Technology State* data input in 3 and calculated in 4.
6. *MRED Operator* inputs the *Available Resources* including *Environments*, *Tools*, and *Personnel*.
7. *MRED Operator* defines the *TTLs-Environment* and *Metrics-Tools* relationship matrices.
8. *MRED* eliminates *TTLs, Metrics, Environments, Tools,* and *Personnel*.
9. *MRED* captures *Stakeholder Preferences* as to which *TTL-Metric* pairs should be tested.
10. *MRED* scores and groups the pairs based upon the *Stakeholder Preferences*
11. *MRED* eliminates low scoring *TTL-Metric* pairs.
12. *MRED* captures *Stakeholder Preferences* as to which *Personnel* should evaluate the remaining candidate *TTL-Metric pairs.*
13. Step 12 is sequentially repeated with the remaining blueprint elements until *MRED* outputs the most preferred blueprints.

The noted relationship matrices are elaborated upon in Section 2.4 while the overall process will be formalized in future work.

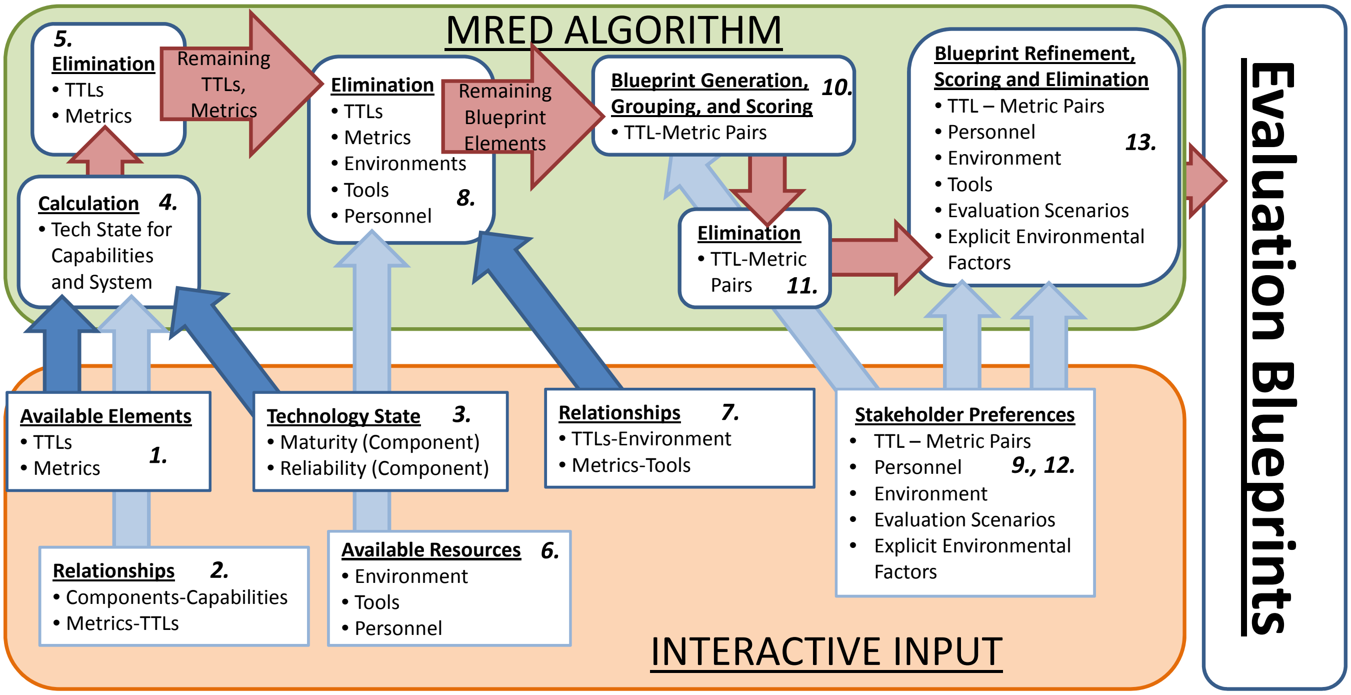


Figure 2. MRED Process Flow Diagram

2.4 Key MRED Relationships

MRED exploits the numerous relationships that exist among the various inputs. Two types of relationships are: (1) physical (the two *Components* of an engine and a transmission work to affect the vehicle's *Capability* of acceleration); and, (2) performance-based (the *Reliability* of the vehicle's acceleration is a function of the *Reliability* of the vehicle's engine and transmission). Since each technology being considered for evaluation is unique, these relationships must be defined by the *MRED Operator* with input from other *Stakeholders*. These relationships (or lack thereof) are critical to MRED's success whereby they are integrated with the inputs defined in Section 2.1. Each set of relationships is represented by one or more matrices within the MRED Algorithm. This section will present these specific relationships.

An example robotic arm will be used to clearly illustrate the relationships as they are defined below. The example robotic arm, shown in Figure 3, is illustrated as a *System* with seven *Components* (C_1, C_2, C_4, and C_6 are revolute joints; C_3 and C_5 are prismatic joints; C_7 is a gripper). These seven *Components* function to provide seven *Capabilities* (P_1, P_2, and P_3 are translation in X, Y, and Z of the end-effector; P_4, P_5, and P_6 are roll, pitch, and yaw of the end-effector; and P_7 is grasping).

MRED interacts with the *Operator* to obtain many of the relationships discussed throughout this section. This is important to note considering that relationships are technology specific and have the potential to change as a technology evolves to its final iteration. MRED's design also contains natural constraints and intrinsic relationships. These are discussed where present.

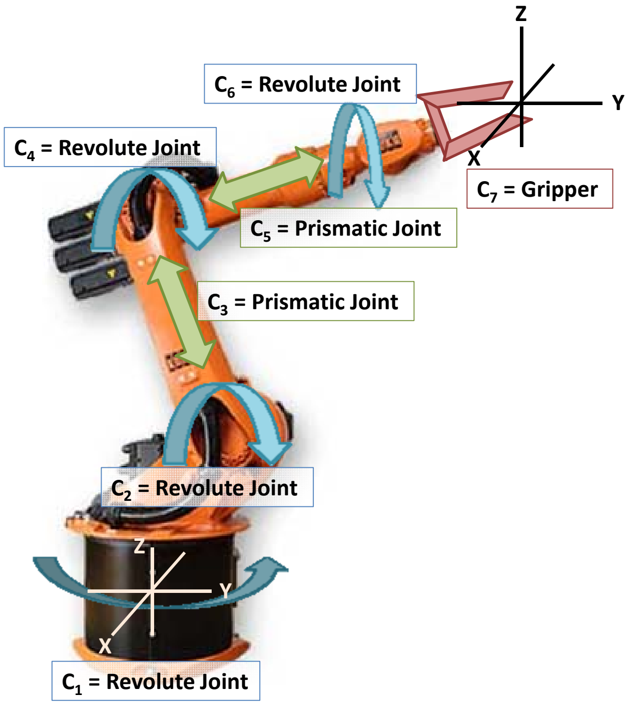


Figure 3. Robotic Arm[2] Example

The first relationship defined in MRED is that between the *Components* and *Capabilities*. This relationship exists because *Capabilities* are only produced through the function of one or more *Components*. This relationship is similar to that between Functional Requirements and Design Parameters as defined by Suh in his theory of Axiomatic Design [7]. An example of this binary matrix is shown in Table 2. In the *Components – Capabilities* Matrix, a "1" cell indicates that the corresponding *Component* contributes to (influences) the corresponding *Capability*. A "0" indicates that no such relationship exists between the *Component* and *Capability*.

Table 2 - Example *Components – Capabilities* Relationship Matrix for Robotic Arm

	CAPABILITIES						
COMPONENTS	X (P_1)	Y (P_2)	Z (P_3)	Roll (P_4)	Pitch (P_5)	Yaw (P_6)	Grasp (P_7)
Rev 1 (C_1)	**1**	**1**	**0**	**0**	**0**	**1**	**0**
Rev 2 (C_2)	**1**	**1**	**1**	**1**	**1**	**0**	**0**
Pris 1 (C_3)	**1**	**1**	**1**	**0**	**0**	**0**	**0**
Rev 3 (C_4)	**1**	**1**	**1**	**1**	**1**	**0**	**0**
Pris 2 (C_5)	**1**	**1**	**1**	**0**	**0**	**0**	**0**
Rev 4 (C_6)	**0**	**0**	**0**	**1**	**1**	**1**	**0**
Gripper (C_7)	**0**	**0**	**0**	**0**	**0**	**0**	**1**

The *Components – Capabilities* relationship is critical when MRED defines the *Maturity* and *Reliability* for *Capabilities* and the *System*. If these *Maturities* and *Reliabilities* are not provided by the *Technology Developer(s)* or *Evaluation Designer(s)* at these *Technology Test Levels*, then they must be calculated given the *Maturity* and *Reliability* of the *Components* along with the *Component* and *Capability* relationship matrix. If unknown, MRED calculates the *System Maturity* and *Reliability* matrices based upon the *Maturities* and *Reliabilities* for the various *Capabilities*. The *Maturities* and *Reliabilities* for each *Component*, *Capability*, and the *System* must be above certain thresholds in order for a specific *TTL* to be considered further for evaluation. If these thresholds are not met, then MRED eliminates these *TTLs* from further testing consideration.

The second set of relationships captured by MRED is that between the *Metrics* and *Technology Test Levels*. This relationship is documented in two matrices; one binary matrix whose columns display all of the *Technology Test Levels* with the rows indicating potential *Technical Performance Metrics* (an example is presented in Table 3)*;* the second binary matrix's columns present the *Capability* and *System Technology Test Levels* with the corresponding *Utility Assessment Metrics* highlighted in the matrix's rows. Note that one of the MRED constraints is that *Utility Assessment Metrics* can only be captured for *Capabilities* and the *System* while *Technical Performance Metrics* can be captured for all three *TTL* groups [9].

Table 3 - Example *Metrics (Technical Performance) - TTL* Relationship Matrix for Robotic Arm

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	P_1	P_2	P_3	P_4	P_5	P_6	P_7	System (S)
Max Force	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
Max Linear Velocity	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
Max Torque	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**
Max Angular Velocity	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**
Range of Motion	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Max Lift Capacity	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**
Speed	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**
Force	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**

[2] Robot arm image courtesy of www.robots.com

The goal of establishing the *TTL – Metric* relationship matrices is to indicate which *Metrics* can be obtained from testing the various *TTLs*. MRED utilizes the data within these relationship matrices numerous times throughout the test plan generation process. In addition, MRED uses this matrix numerous times to eliminate either *TTLs* or *Metrics* if the other is eliminated in a prior step (presented in Section 2.3). For example, if a *TTL* is eliminated because its *Maturity* and *Reliability* do not meet the designated threshold, then MRED would eliminate any *Metrics* that solely correspond to this *TTL* which would only be shown in the *Metrics – TTL* relationship matrix.

The third set of binary relationship matrices captured in MRED are the *TTL – Environment* matrices. The three specific *TTL – Environment* matrices are: 1) *Components and Capabilities* (rows) – *Lab Environment* (columns), 2) *Components*, *Capabilities* and *System* (rows) – *Simulated Environment* (columns), and 3) *Capabilities* and *System* (rows) – *Actual Environment* (columns). The necessity of these three matrices is brought upon by MRED's constraints that only *Components* and *Capabilities* can be tested in *Lab Environments* and only *Capabilities* and the *System* can be tested in *Actual Environments* [10] [11]. A *TTL – Environment* relationship matrix is presented using the robotic arm example in Table 4.

Table 4 – Example *Components* and *Capabilities* - *Environment* (*Lab*) Matrix for Robotic Arm

		LAB ENVIRONMENTS		
		ABC Controls Lab	ABC Robotics Lab	DEF Force/Torque Lab
COMPONENTS and CAPABILITIES	C_1	1	0	1
	C_2	1	0	1
	C_3	1	0	1
	C_4	1	0	1
	C_5	1	0	1
	C_6	1	0	1
	C_7	1	0	0
	P_1	0	1	0
	P_2	0	1	0
	P_3	0	1	0
	P_4	0	1	0
	P_5	0	1	0
	P_6	0	1	0
	P_7	0	1	0

The goal of establishing the *TTL – Environment* matrices is to indicate which of the candidate *TTLs* could be tested in the various environments. Figure 4 presents a screen capture from the interactive MRED interface (created in Matlab[3]) that enables the *Evaluation Designer* to indicate the available *Environments* (shown in the top half of the figure) and specify the *TTL – Environment* relationship matrices (in the bottom half of the figure). If there are no candidate *Environments* available to test a specific *TTL*, then MRED eliminates this *TTL* from further testing consideration. If MRED eliminates *TTLs* at this stage because there are no viable *Environments* that exist, then MRED checks the *Metrics – TTL* relationship matrix and eliminates those *Metrics* that only correspond to the eliminated *TTL(s)*.

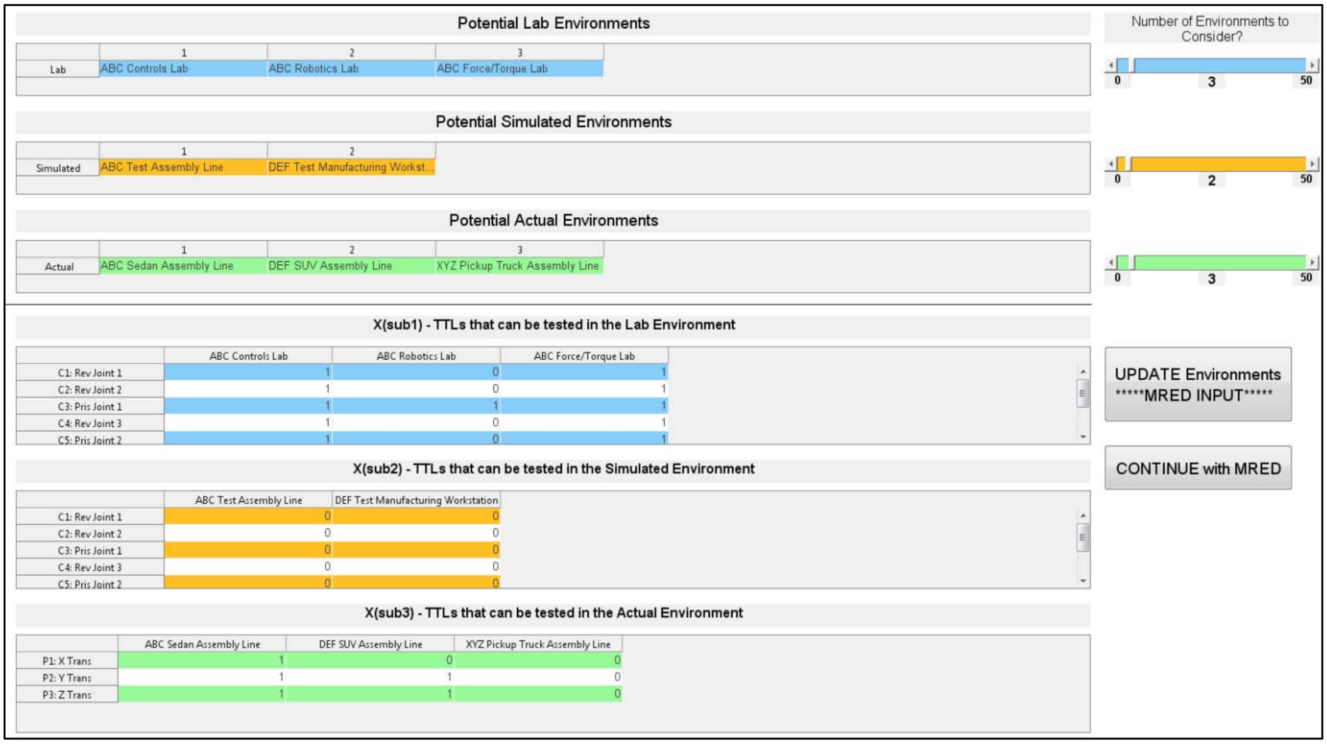
The fourth set of binary relationship matrices captured in MRED are the *Metric – Tools* matrices. The two relationship matrices in this category are: 1) *Technical Performance Metrics – Tools* and 2) *Utility Assessment Metrics – Tools*. The first matrix only includes those data collection and analysis tools that support the generation of *Technical Performance Metrics* while the second includes those tools that support the production of *Utility Assessment Metrics*.

Table 5 – Example *Technical Performance Metrics – Tools* Matrix for Robotic Arm

		TOOLS		
		Tension Sensor	Dynamometer	LADAR
TECHNICAL PERFORMANCE METRICS	Max Force	1	1	0
	Max Linear Velocity	0	0	1
	Max Torque	0	1	0
	Range of Motion	0	0	1
	Max Lift Capacity	1	0	0
	Speed	0	0	1
	Force	1	1	0

The benefit of these matrices is that they indicate if any *Tools* are unnecessary (in that they do not support any of the *Metrics*) and/or if *Metrics* cannot be obtained (if the appropriate *Tools* are unavailable). Similar to the *Environment – TTL* relationship matrices, the *Metric – Tools* matrices are used to eliminate *Metrics* if there are no candidate *Tools* available to capture the required data. If MRED eliminates *Metrics* due to a lack of *Tools,* then MRED checks the *Metrics – TTLs* relationship matrix and eliminates those *TTL(s)* that only correspond to the eliminated *Metric(s)*. MRED accesses the data in this set of matrices several times throughout the test plan generation process.

[3] Certain commercial companies, products and software are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the companies, products, and software identified are necessarily the best available for the purpose.



Potential Lab Environments

	1	2	3
Lab	ABC Controls Lab	ABC Robotics Lab	ABC Force/Torque Lab

Number of Environments to Consider? 3

Potential Simulated Environments

	1	2
Simulated	ABC Test Assembly Line	DEF Test Manufacturing Workst...

Number of Environments to Consider? 2

Potential Actual Environments

	1	2	3
Actual	ABC Sedan Assembly Line	DEF SUV Assembly Line	XYZ Pickup Truck Assembly Line

Number of Environments to Consider? 3

X(sub1) - TTLs that can be tested in the Lab Environment

	ABC Controls Lab	ABC Robotics Lab	ABC Force/Torque Lab
C1: Rev Joint 1	1	0	1
C2: Rev Joint 2	1	0	1
C3: Pris Joint 1	1	1	1
C4: Rev Joint 3	1	0	1
C5: Pris Joint 2	1	0	1

X(sub2) - TTLs that can be tested in the Simulated Environment

	ABC Test Assembly Line	DEF Test Manufacturing Workstation
C1: Rev Joint 1	0	0
C2: Rev Joint 2	0	0
C3: Pris Joint 1	0	0
C4: Rev Joint 3	0	0
C5: Pris Joint 2	0	0

X(sub3) - TTLs that can be tested in the Actual Environment

	ABC Sedan Assembly Line	DEF SUV Assembly Line	XYZ Pickup Truck Assembly Line
P1: X Trans	1	0	0
P2: Y Trans	1	1	0
P3: Z Trans	1	1	0

Figure 4. Example Interactive MRED Screen from Matlab Presenting the Available Environments and corresponding *TTL – Environment* relationship matrices

Additional relationship matrices, including those relating *Personnel* and *Environments* are still being finalized and will be discussed in future work.

The last set of inputs into MRED comes from the *Stakeholders* in the form of *Stakeholder Preferences*. MRED presents the list of the candidate blueprint elements to the *Stakeholders* based upon those elements that are still available after *Maturities* and *Reliabilities* are calculated, relationships are defined, and available *Resources* are input. It is critical that their subjective preferences are appropriately captured and reflected in MRED. If not, the output evaluation blueprints will not accurately reflect the wishes of the *Stakeholders*. Preference is the topic of the next section.

3. PREFERENCE CAPTURE

The MRED inputs shown in Figure 1 are objective with the exception of the *Stakeholder Preferences*. These subjective preferences are supported by each *Stakeholder's* knowledge of the facts. Providing preferences to ultimately select evaluation blueprints is different than what is encountered in product development. Each class of *Stakeholders* could potentially select entirely unique blueprints with very different test plan elements. This is not the case in product development where preferences provided on constituent attributes (product size, weight, etc.) all contribute to the same overriding goal of profit for the business. In product development, the decision-makers are usually all employees of the same entity. In the typical development of an evaluation, input from different *Stakeholders* (often with competing interests) is collected and valued.

Accurately capturing and representing the preferences of the various stakeholders is critical to MRED's success. The *Stakeholder Preferences* are central to further reducing the set of candidate *TTLs* and *Metrics* down to those that are most valuable for testing at the present time. Likewise, these preferences also play a crucial role in determining what *Environment(s)* the *TTLs* should be tested, what type of *Evaluation Scenarios* will be used, and who (*Personnel*) will be using the technology during the evaluation exercises. Further, analyzing the preferences from multiple stakeholders to select the most preferred options is another key step within MRED. This step reflects that of group decision-making.

This section will present background on several preference capture and group decision-making methods, introduce the preference capture method of 'Evaluative Voting' that MRED is adopting, and discuss how it will be implemented into MRED's algorithm.

3.1 Background

Preference capture is a topic that has been studied for decades by researchers in many fields including economics and engineering design. Preference can be defined as *the power, right, or opportunity of choosing*[4] and as a *positive regard for something*[5]. In turn, preference capture is the act of obtaining an individual's or group's desires on one or more options. Each proposed preference capture method attempts to find out what an individual or group really wants. Many group decision-making methods have been produced and refined over many years of study. There are numerous challenges to effectively capturing group preferences including [8]:

- Delineating between weak and strong preferences for alternatives
- Comparing preferences between group members if there is minimal to no overlap on preferences of discrete alternatives

[4] http://www.merriam-webster.com/dictionary/preference

[5] http://www.merriam-webster.com/thesaurus/preference

- Weighting the importance of the attributes to one another that compose the alternatives
- Weighting the importance of each group member's preferences to one another
- Competing objectives or priorities held by different group members (this raises issues of fairness or equitable distribution if members do not share a common objective) so a Pareto Optimal frontier cannot be defined [8]
- Lack of a method for aggregating individual rankings "that does not directly or indirectly include interpersonal comparisons of preference" which does not resolve Arrow's Impossibility Theorem [8]

One method of preference capture and group decision-making is the Borda count, which is often referred to as a voting method [1] [2] [6]. The Borda count was developed as a method to allow a group of individuals to rank order candidates and select the 'most preferred' candidate among the members. This method is implemented by first asking the voters to individually rank the *n* candidates from 1 to *n* with the candidate being ranked number 1 the most preferred and the candidate being ranked *n* being the least preferred. If a voter chooses not to rank one of the candidates (whether they are indifferent or don't have enough information), then this candidate is ranked last (so multiple candidates could be ranked last). The Borda Count then turns the individual rankings into scores by giving *n* points to the candidate ranked 1st, *n-1* points to the candidate ranked 2nd, etc. Voter's scores for each candidate are added together and the candidate that receives the highest score is considered the winner (or 'most preferred'). This is a simple method to implement.

There are several drawbacks to this method that eliminated it from consideration with MRED. In general, the Borda Count satisfies Arrow's first four axioms yet violates Arrow's fifth axiom, *Independence of irrelevant alternatives*[6] [2]. Specifically, it is susceptible to agenda manipulation [1] in that it does not account for majority preferences at all. This method is strictly ordinal and it does not enable MRED *Stakeholders* to delineate the distance between adjacently-ranked alternatives. In this sense, a candidate that a *Stakeholder* is indifferent on would be scored the same as a candidate the *Stakeholder* finds least appealing (last).

Pairwise comparison is another method of preference capture and can be used to achieve a group decision when combined with other methods [2]. Pairwise comparison is predicated upon all alternatives being compared one-to-one. Although this method has been proven effective in some applications, it is not practical for integration with MRED. Specifically, the vast number of alternatives to be compared during the various steps of the *Stakeholder Preference* capture process would result in an extremely time-consuming process. It's possible that *Stakeholders* would have to compare over 20 alternatives which would require nearly 200 pairwise comparisons. Further, Arrow's Impossibility Theorem restricts aggregation of pairwise comparison [3].

There are many other methods available to capture individual preferences and produce a group decision. One such category includes methods in the area of Multi-Attribute Decision-Making (MADM) and Multi-Criteria Decision-Making (MCDM) [5] [14]. These methods have been proven beneficial when a selection must be made among various alternatives where each alternative is valued against one or more attributes.

This category of methods does not appear to be suitable for use with MRED. One important reason is MADM would require all possible blueprints to be input as the list of alternatives. This would potentially lead to a combinatory explosion of blueprints. If this were done for the robotic arm example introduced in Section 2.4, then it is likely hundreds, if not thousands of blueprints would have to be considered. This example includes up to:

- 15 *TTLs* (7 *Components*, 7 *Capabilities*, and 1 *System*)
- 6 *Metrics* (on average and including both *Technical Performance* and *Utility Assessment Metrics*)
- 5 *Environments* (on average, across the *Lab*, *Simulated* and *Actual Environments)*
- 3 types of *Technology Users* (part of the *Personnel* input)
- 3 types of *Evaluation Scenarios*
- And consideration to additional *Personnel* and *Explicit Environmental Factors*.

The above information would yield approximately 4050 sets of blueprints (15 x 6 x 5 x 3 x 3). Only by stepping through MRED, would one know exactly how many blueprints are being considered since *TTLs* and/or *Metrics* can be grouped together, test plan elements could be eliminated based upon *Maturity* and *Reliability*, etc.

Another reason that MADM is not suitable for integration with MRED is because the blueprints and diversity among *Stakeholder Preferences* is too complex to produce an objective function. The objective function is determined from the output of the tests since there's no way to indicate a preference rating in MADM.

Asking each *Stakeholder* to rate all of these blueprints would be tremendously time-consuming, especially considering that not all *Stakeholders* will care to test every *TTL*, generate every potential *Metric*, etc.

Realizing that MRED has the potential to generate an unnecessary and excessive amount of blueprints, it is important to identify a method that will capture the *Stakeholders' Preferences* in an inexpensive and timely manner, along with the ability to eliminate undesirable test plan elements prior to final blueprint selection to further streamline the process.

3.2 Evaluative Voting

MRED will leverage the method of Evaluative Voting to enable *Stakeholder Preference* capture on an independent cardinal scale [4]. Evaluative Voting is a method where voters (*Stakeholders*) score each alternative on an integer scale to signify their preference for, neutral, or against testing a particular *TTL-metric* pair. Using Hillinger's [4] general election *EV-3* scale (-1,0,1), a *Stakeholder* would give each alternative a score of '-1' (against the alternative), '0' (neutral stance), or '1' (for the alternative). An initial example of applying the *EV-3* scale to MRED would be asking the *Stakeholders* to score each of the available *TTLs* in regarding their agreement to the statement of "This *TTL* should be evaluated." A *Stakeholder* would vote '-1'to indicate they are against testing a *TTL* (they disagree with the statement); '0' to indicate they are indifferent as to if the *TTL* should be tested; or '1' to indicate they believe the *TTL* should be tested (they agree with the statement). The EV method provides a score of '0' if a voter decides not to cast their vote regarding a specific candidate. In the case of MRED, if a *Stakeholder* chooses not to vote on a specific element (due to a lack of information), the vote remains at the default of 'NV' to indicate they are recusing themselves from

[6] *Independence of irrelevant alternatives* (IIA) is defined as: If the aggregate ranking would choose A over B when C is not considered, then it will not choose B over A when C is considered.

scoring that specific element. This is different from the originally-defined EV method in that MRED does not average in a score of '0.' However, MRED does average in a score of '0' if a *Stakeholder* actively scores a specific element as neutral. The rationale behind this decision is that neutral preferences have a mathematical impact on the overall scores, where their lack of inclusion can present misleading data.

There are numerous benefits to integrating Evaluating Voting with MRED to capture *Stakeholder Preferences* [4]. They are:

- Enables the aggregation of judgments on a cardinal scale
- Avoids highly scoring a minority candidate which could occur with the Borda Count, Plurality Voting, and other voting methods
- Simple to implement and for the *Stakeholders* to understand
- Method is comparable to other judgments expressed on cardinal scales such as grades (given in schools, universities, etc.) which are often aggregated through averaging
- Successfully implemented using scales larger than (-1,0,1)
- Accounts for a *Stakeholder* that chooses not to vote on a specific element in such a manner that does not incorrectly inflate or deflate an element's score

Hillinger recommends using the *EV-3* scale (-1,0,1) for general elections (selection of a single candidate) and the *EV-5* scale (-2,-1,0,1,2) for expert decisions. A German political survey institute adopted an 11-point scale (-5,-4,-3,-2,-1,0,1,2,3,4,5) when asking survey respondents to rate their satisfaction with politicians. The University of Michigan Survey Research Center used a much larger integer scale (0 to 100) to capture voters' perceptions of candidates [4]. The 11-point scale (also known as the Forschungsgruppe Wahlen scale after the German institute that devised this scale) is selected for use with MRED. This decision is made based upon the amount of *TTL-metric pairs* that *Stakeholder Preferences* would be solicited and that multiple elements will be selected for consideration (while the lowest scoring elements will be eliminated from further consideration),

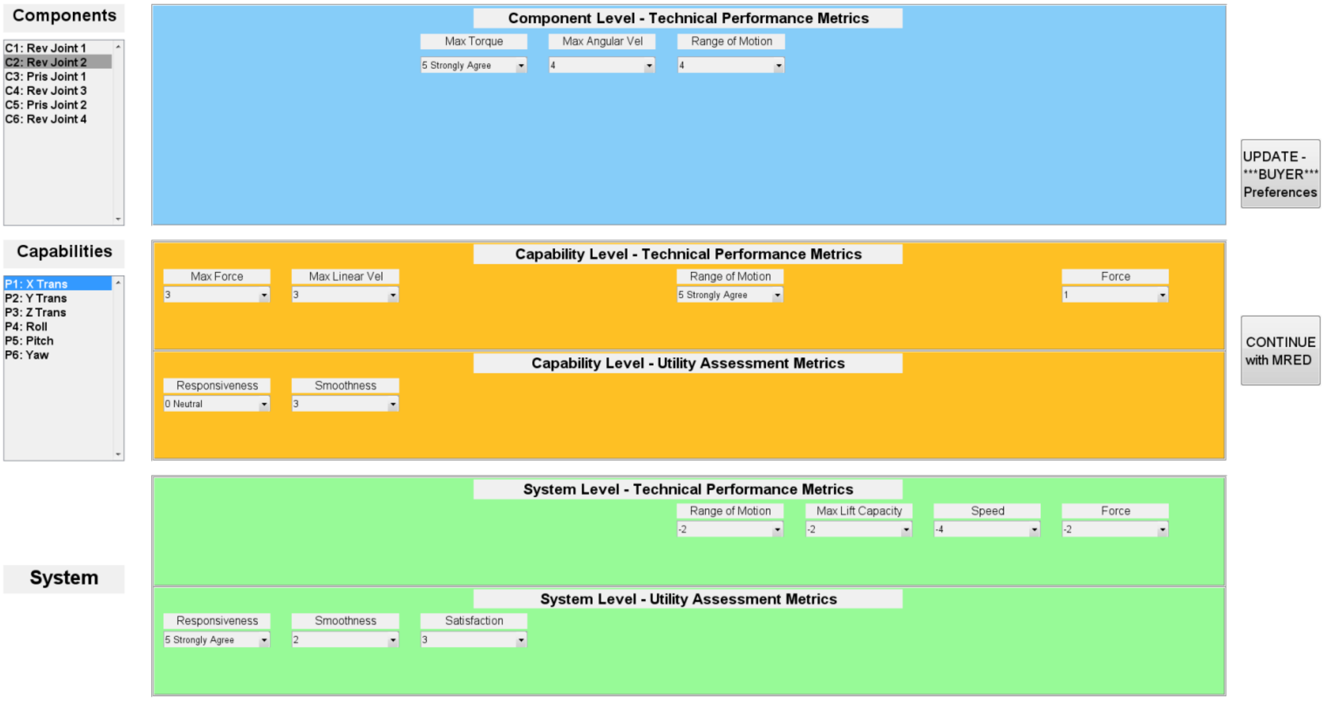
The following section will discuss how Evaluative Voting will be integrated with MRED to ultimately output preferred blueprints given *Stakeholder Preferences*.

3.3 MRED Implementation

An iterative approach is used with respect to implementing Evaluative Voting with MRED. This iterative process consists of 1) capturing *Stakeholder Preferences* of a single set of test plan elements, 2) aggregating these cardinal scores whereby the weakest scoring elements are eliminated from further consideration, and 3) the remaining test plan elements are then considered with another set of test plan elements for further preference capture. This process is repeated until a series of candidate test plan elements is output. Figure 5 illustrates this approach with respect to the robotic arm example. This figure presents the Matlab MRED interface for capturing Stakeholder Preferences for the *Metric-TTL* pairs. The process begins with 1) all of the *Stakeholders* inputting their preferences for each *Metric-TTL* pair on the selected Evaluative Voting scale, 2) these preference scores being aggregated where those *Metric-TTL* pairs scoring lower than '0' (or another threshold set by the *Evaluation Designer*) being eliminated and 3) the remaining *Metric-TTL* pairs being passed through to the next set of test plan elements.

This approach offers numerous benefits in both capturing *Stakeholder Preferences* and using these preferences to both eliminate low-scoring blueprint elements and highlight high-scoring blueprint elements. This approach will be discussed further , followed by its advantages and disadvantages.

Implementing Collaborative Evaluative Voting (CEV) in MRED begins with having the *Stakeholders* score each of the available



Component Level - Technical Performance Metrics

Max Torque: 5 Strongly Agree
Max Angular Vel: 4
Range of Motion: 4

Capability Level - Technical Performance Metrics

Max Force: 3
Max Linear Vel: 3
Range of Motion: 5 Strongly Agree
Force: 1

Capability Level - Utility Assessment Metrics

Responsiveness: 0 Neutral
Smoothness: 3

System Level - Technical Performance Metrics

Range of Motion: -2
Max Lift Capacity: -2
Speed: -4
Force: -2

System Level - Utility Assessment Metrics

Responsiveness: 5 Strongly Agree
Smoothness: 2
Satisfaction: 3

Figure 5. Example Collaborative Implementation with respect to capturing Stakeholder Preferences of *Metric-TTL* pairs for the robotic arm

TTL-Metric pairs on the 11-point scale. The *Stakeholders Preference* scores for the *TTL-Metric* pairs are averaged and *TTL-Metric* pairs with an average score less than 0 are eliminated from further consideration. A negative average score indicates that the group's aggregate preference is to not evaluate this *TTL-Metric* pair. The only exception where a negatively scoring *TTL-Metric* pair could still be considered for further evaluation is if it's grouped with other *TTL-Metric* pairs (either of the same *TTL* or same *Metric*) that were scored above '0.'

MRED then requests *Stakeholder Preferences* on the possible *Personnel/TTL-Metric* pair combinations based upon the *TTL-Metric* pairs that scored above '0,' the available *Personnel*, and MRED's constraints on which *Personnel* can realistically interact and/or evaluate the different types of *TTL-Metric* pairs. The scores for each *Personnel/TTL-Metric* pair combination are averaged and those combinations scoring a '0' or lower are eliminated from further consideration. In some instances, it may be desired to set the elimination threshold to a higher value (e.g., '1' or '1.5'). This would be at the *MRED Operator's* discretion given the total amount of *TTL-Metric* pairs being considered, the number of pairs with positive averages, etc.

Once the *TTL-Metric* pairs are combined with additional blueprint elements, it's plausible that some of the *Stakeholders* may not have preferences regarding specific combinations. This situation is likely due to a *Stakeholder* being asked to rate a combination whose *TTL-Metric* pair the *Stakeholder* rated poorly or did not have an opinion. To counteract this situation, *Stakeholders* have the power to issue a 'NV' for an entire group of blueprint elements, in addition to individual elements.

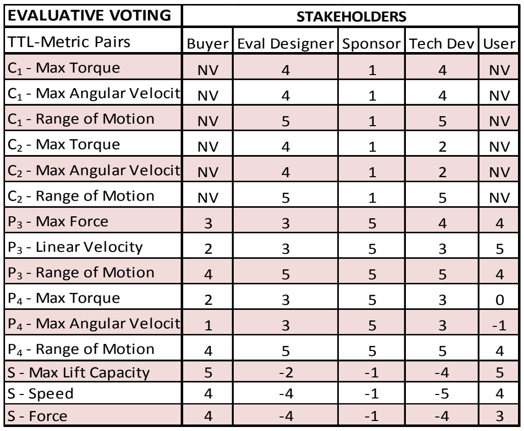
The CEV process of 1) preference capture, 2) averaging, and 3) elimination is repeated with *Personnel, Knowledge,* and *Autonomy Levels*, then *Environments*, followed by *Evaluation Scenarios* and finally *Explicit Environmental Factors*. The final output of this process is a series of blueprints ordered based upon those receiving the highest scores throughout the CEV process. This process is demonstrated in an example throughout the following section.

4. MRED EXAMPLE

The CEV process is demonstrated using the robot arm example presented in Section 2.4. A subset of the *TTL*s and *Metrics* are paired up according to the relationships presented in Table 3 where the *Stakeholders* provide their preferences to evaluate each *TTL-Metric* pair according to the Evaluative Voting process defined in Section 3.2. The *Stakeholder Preference* scores are presented in Table 6.

The reason a subset of the potential *TTL-Metric* pairs are used in this example is so that the process could be shown in detail. The full set of *TTL-Metric* pairs is easily scored, averaged, and processed in Matlab code that is being developed. The overall robotic example is not as large or complex (relatively speaking) as compared to other technologies. An autonomous ground vehicle would be an example of a more complicated technology for evaluation.

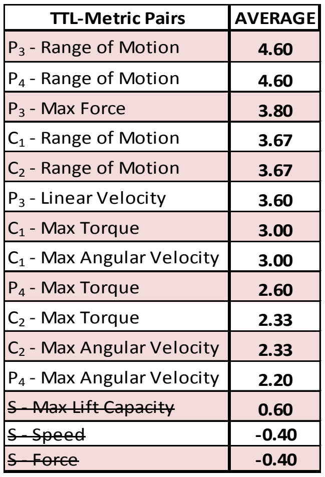
Table 6 - Evaluative Voting Scores for TTL-Metric Pairs for Robotic Arm



EVALUATIVE VOTING	STAKEHOLDERS				
TTL-Metric Pairs	Buyer	Eval Designer	Sponsor	Tech Dev	User
C_1 - Max Torque	NV	4	1	4	NV
C_1 - Max Angular Velocit	NV	4	1	4	NV
C_1 - Range of Motion	NV	5	1	5	NV
C_2 - Max Torque	NV	4	1	2	NV
C_2 - Max Angular Velocit	NV	4	1	2	NV
C_2 - Range of Motion	NV	5	1	5	NV
P_3 - Max Force	3	3	5	4	4
P_3 - Linear Velocity	2	3	5	3	5
P_3 - Range of Motion	4	5	5	5	4
P_4 - Max Torque	2	3	5	3	0
P_4 - Max Angular Velocit	1	3	5	3	-1
P_4 - Range of Motion	4	5	5	5	4
S - Max Lift Capacity	5	-2	-1	-4	5
S - Speed	4	-4	-1	-5	4
S - Force	4	-4	-1	-4	3

Recall that 'NV' indicates No Vote. This means that the average of the first *TTL-Metric* pair presented in Table 6 is (4+1+4)/3=3 since two *Stakeholders* cast an 'NV' score. In this example, if an 'NV' score was counted as '0' and averaged with the other scores, then this *TTL-Metric* pair average would be (0+4+1+4+0)/5=1.8. Removing 'NV' scores from the averages enables the *Stakeholders* to not impact the option to evaluate or not to evaluate a given *TTL-Metric* pair (at this step in the overall process) if they believe they are not equipped to make an informed decision. Table 7 presents the average scores from Table 6.

Table 7 - Evaluative Voting Averages for TTL-Metric Pairs for Robotic Arm



TTL-Metric Pairs	AVERAGE
P_3 - Range of Motion	**4.60**
P_4 - Range of Motion	**4.60**
P_3 - Max Force	**3.80**
C_1 - Range of Motion	**3.67**
C_2 - Range of Motion	**3.67**
P_3 - Linear Velocity	**3.60**
C_1 - Max Torque	**3.00**
C_1 - Max Angular Velocity	**3.00**
P_4 - Max Torque	**2.60**
C_2 - Max Torque	**2.33**
C_2 - Max Angular Velocity	**2.33**
P_4 - Max Angular Velocity	**2.20**
~~S - Max Lift Capacity~~	**0.60**
~~S - Speed~~	**-0.40**
~~S - Force~~	**-0.40**

The bottom three *TTL-Metric* pairs are removed from further consideration given the negative scores of the last two pairs and the range between these averages and the rest of the pairs.

It is not surprising to see the *System* excluded from consideration (shown in Table 7). This is likely early on in the development process and during the first round of evaluations where *Components* and *Capabilities* are still undergoing significant changes. Whether or not the *System* is ready for testing at this point is heavily dependent upon the type of *Technology*, the *Maturity* and *Reliability* of its constituent *TTLs*, etc.

The next step in the CEV process is to have each *Stakeholder* assign their preference scores for the possible *Personnel* that could use and/or evaluate each of the *TTL-Metric* pairs. Given that there are numerous *Personnel* options available for testing, the *Evaluation Designer* must consider the practicality of grouping pairs by *TTL* or pairs by *Metric*. In what manner they should be grouped (*TTL* vs. *Metric*) and even if they should be grouped at all is technology-specific and driven by the quantity of *TTL-Metric* pairs.

Groupings by *Technology Test Level (TTL)* are established in Table 8. This appears to be a logical decision considering that the 12 remaining *TTL-Metric* pairs are split among four unique *TTLs*. Note that Table 8 not only presents the individual pair averages within each group, it also shows the group average of these pairs and the pair average max within a group. Both of these values are important to consider when moving deeper into the CEV process so it's easily identifiable as to what groups, on the whole, are important to evaluate and which groups have the most critical elements.

Table 8 - TTL Groupings of Remaining TTL-Metric Pairs for Robotic Arm

TTL GROUPINGS		METRICS	Pair Averages	Group Average	Pair Average Max
	P_3	Range of Motion	4.60	4.00	4.60
		Max Force	3.80		
		Linear Velocity	3.60		
	P_4	Range of Motion	4.60	3.13	4.60
		Max Torque	2.60		
		Max Angular Velocity	2.20		
	C_1	Range of Motion	3.67	3.22	3.67
		Max Torque	3.00		
		Max Angular Velocity	3.00		
	C_2	Range of Motion	3.67	2.78	3.67
		Max Torque	2.33		
		Max Angular Velocity	2.33		

5. CONCLUSION

The definition of key relationships exploited within MRED and the integration of Collaborative Evaluating Voting (CEV) are fundamental pieces in the finalization of the MRED methodology. These relationship matrices, along with the *Evaluation Designer's* ability to set the specific relations, enable MRED to eliminate test plan elements based upon the availability of their relations. Likewise, CEV allows MRED to capture the *Stakeholder Preferences* of the various test plan elements which will ultimately lead to the generation of the most preferred sets of evaluation blueprints. The next efforts will finalize the MRED methodology to include scoring the output sets of evaluation blueprints so it's evident which are most preferred. MRED is proving to be an invaluable tool towards the generation and rapid re-iteration of evaluation blueprints to test complex, advanced, and intelligent systems.

6. ACKNOWLEDGMENTS

The authors would like to acknowledge the support of NIST's Intelligent Systems Division.

7. REFERENCES

[1] Dummett, M., 1998, "The Borda Count and Agenda Manipulation," *Social Choice and Welfare*. 15, 2 (1998), 289-296.

[2] Dym, C., Wood, W. and Scott, M., 2002, "Rank Ordering Engineering Designs: Pairwise Comparison Charts and Borda Counts," *Research in Engineering Design*. 13 (2002), 236-242.

[3] Geanakoplos, J., 2005, "Three brief proofs of Arrow's Impossibility Theorem," *Economic Theory*. 26 (2005), 211-215.

[4] Hillinger, Claude, 2004, "Voting and the Cardinal Aggregation of Judgments," Munich Discussion Paper 2004-9 (May 2004), http://epub.ub.uni-muenchen.de/353/.

[5] Olcer, A. and Odabasi, A., 2005, "A new fuzzy multiple attribute group decision making methodology and its application to propulsion/maneuvering system selection problem," *European Journal of Operational Research*. 166 (2005), 93-114.

[6] Saari, D., 2006, "Which is Better: the Condorcet or Borda Winner?" *Social Choice and Welfare*. 26, 1 (January 2006), 107-129.

[7] Suh, N.P., 1998, "Axiomatic Design Theory for Systems," *Research in Engineering Design*. 10,4 (December 1998), 189-209.

[8] Thurston, D.L., 2001, "Real and Misconceived Limitations to Decision Based Design with Utility Analysis," *Journal of Mechanical Design*. 123 (June 2001), 176-182.

[9] 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)*.

[10] 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*.

[11] Weiss, B.A. and Schmidt, L.C., 2011, "The Multi-Relationship Evaluation Design Framework: Producing Evaluation Blueprints to Test Emerging, Advanced, and Intelligent Systems," *ITEA Journal*. 32, 2 (June 2011), 191-200.

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

[13] Weiss, B.A. and Schmidt, L.C., 2011, "Multi-Relationship Evaluation Design: Formalizing Test Plan Input and Output Blueprint Elements for Testing Developing Intelligent Systems," *ITEA Journal*. 32, 4 (December 2011), 479-488.

[14] Xu, Z., 2007, "Multiple-Attribute Group Decision Making with Different Formats of Preference Information on Attributes," *IEEE Transactions on Systems, Man, and Cybernetics*. 37, 6 (December 2007), 1500-1511.