ABSTRACT

Title of Document: MULTI-RELATIONSHIP EVALUATION DESIGN (MRED): AN INTERACTIVE TEST PLAN DESIGNER FOR ADVANCED AND EMERGING TECHNOLOGIES

Brian Adam Weiss, Doctor of Philosophy in Mechanical Engineering, 2012

Directed By: Dr. Linda Schmidt, Associate Professor, Department of Mechanical Engineering

Ground-breaking technologies are developed for use across a broad range of domains such as manufacturing, military, homeland security and automotive industries. These advanced technologies often include intelligent systems or robotic elements. Evaluations are a critical step in the development of these advanced systems. Evaluation events inform the technology developers of specific needs for enhancement, capture end-user feedback, and verify the extent of the technology's functions. Test exercises are an opportunity to showcase the technology's current abilities and limitations and provide data for future test efforts. The objective of this research is to develop the Multi-Relationship Evaluation Design (MRED) methodology, an interactive test plan blueprint generator. MRED collects multiple

inputs, processes them interactively with a test designer and outputs evaluation blueprints, specifying key test-plan characteristics. Drawing from the Systems Engineering Paradigm, MRED models a process that had not been modeled before. The MRED model is consistent with the experience of evaluation designers. This method also captures and handles stakeholder preferences so that they can be accommodated in a meaningful way. The result is the MRED methodology that combines practical evaluation design experience with mathematical methods proven in the literature.

MULTI-RELATIONSHIP EVALUATION DESIGN (MRED): AN INTERACTIVE TEST PLAN DESIGNER FOR ADVANCED AND EMERGING TECHNOLOGIES

By

Brian Adam Weiss

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in
Mechanical Engineering
2012

Advisory Committee:
Associate Professor Linda Schmidt, Chair
Professor Satyandra Gupta
Professor Peter Sandborn
Associate Professor Jeffrey Herrmann
Professor Bilal Ayyub, Dean’s Representative

To My Dear Wife, Ilyssa

Acknowledgements

Many individuals supported me during the course of this five year research effort and I am deeply indebted to them. First, I would like to express my gratitude to my advisor, Dr. Linda Schmidt. I have been fortunate to share many wonderful moments with her during this journey including numerous brainstorming sessions and countless online meetings. She always knew when to pull me back in from tangential discussions or give me pep talks when the challenges began to pile up. She perfectly blended scientific focus with honest guidance as she shephered me through the research. I could not have done this without her. Special thanks to my dissertation committee members; Dr. Jeffrey Herrmann, Dr. Satyandra Gupta, Dr. Peter Sandborn, and Dr. Bilal Ayyub. Their words of encouragement and scholarly advice were most appreciated.

I would also like to recognize several of my colleagues at the National Institute of Standards and Technology. Craig Schlenoff has been an invaluable friend and colleague for the past eight years as we have worked together to evaluate numerous advanced and intelligent technologies for several government agencies. Harry Scott has been an sound mentor, both offering constructive feedback on my numerous MRED publications and supporting my professional growth and development. The efforts of Dr. Stephen Balakirsky and Brian Antonishek are greatly appreciated. Their reviews of my seven relevant publications were marked by insight and attention to detail.

I'd like to acknowledge my closets friends and family who helped me through this journey. Jonathan Manset has been a sounding board, well before I began this

research endeavor. His candid advice has always been welcome and he has played a significant part in keeping me grounded these past five years. My parents, Layne & Harlan, have become some of the best listeners in my life. They always provided sound counsel and wisdom when listening to the trials of my research. I am forever grateful to them.

Lastly, I am eternally in-debted to my devoted wife, Ilyssa. Her unwavering support is deeply treasured and gave me the constant push to make it through this journey. She sacrificed both time and energy to help me through the challenging times and she has always been my biggest fan. With her by my side, I proudly conclude this chapter of my life and look forward to the next.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Ground-breaking technologies are developed for use across a broad range of domains such as manufacturing, military, homeland security and automotive manufacturing. These advanced technologies often include intelligent systems or robotic elements. Evaluations are a critical step in the development of these advanced systems. Evaluation events inform the technology developers of specific needs for enhancement, capture end-user feedback, and verify the extent of the technology's functions. Test exercises are an opportunity to showcase the technology's current abilities and limitations and provide data for future test efforts. Many researchers have documented the necessity of evaluation regimens and how they guide Artificial Intelligence (AI) system research and development (Cohen and Howe, 2008; Gao and Tsoukalas, 2002). Leedom states that scientific rigor in test methods is vital to moving forward in the development of intelligent systems (Leedom, 2003). In concert with the need for scientific rigor, Hubey points out that a general scientific foundation is the basis for measuring complex phenomena and intelligence (Hubey, 2001).

Many of these complex technologies are being developed by or for the government. It is typical for the government to fund these development programs on multi-year schedules under open competition. Such programs differ from product development efforts in that the government structures its programs into multiple phases. Each development phase typically consists of one or more formal evaluation events assessing technologies developed by one or more companies. Technology developers are awarded contracts based upon their responses to government-issued

Broad Agency Announcements (BAAs). Likewise, continuing funding may also be influenced by the outcome of evaluation events.

The BAA not only describes the technology the government is seeking, it also provides some go/no-go evaluation metrics and evaluation criteria that must be met for advancement within the program. The soliciting government agency either designs and implements the appropriate testing of the developmental technology or contracts to an independent third party (usually another government agency) to spearhead the test efforts. These program solicitations reflect the importance placed upon the technology evaluations even prior to contract awards. It is evident that the evaluation design process has a significant impact on these types of efforts.

1.1. Need for Test and Evaluation

Human-robot interaction (HRI) or human computer interaction (HCI) is common among emerging and advanced technologies (Dautenhahn, 2007). This field also includes augmented reality as a method to enhance a technology operator's perception (Green et al., 2008). Researchers have put forth considerable effort to devise metrics to adequately evaluate the quality of human-robot interactions including those technologies that are capable of variable autonomy (Billman and Steinberg, 2007; Olsen and Goodrich, 2003). Scholtz defines five different interaction roles among humans and intelligent systems (Scholtz, 2002). The human operator may be controlling all system functions, observing the technology's behavior, or exerting various levels of control in between these two extremes. No matter the level of autonomy of these emerging and intelligent systems, there is always a human-in-the-loop.

Autonomous ground and air vehicle technologies are intelligent systems that have evolved significantly over the past decades. These systems feature complex subsystems and numerous capabilities including intelligent control architectures, automated positioning and mapping systems. Advancements are constantly being made in both the systems' capabilities and the human-robot interfaces. These technologies have motivated the generation and execution of extensive test exercises, most of which are conducted before the final systems are deployed (Albus et al., 2006; Albus, 2002; Bostelman et al., 2006; Lacaze et al., 2002; Scrapper et al., 2008).

Another area of advanced technology is that of small unmanned robotic systems. Researchers and technology developers spend considerable time in developing such a system's constituent components and capabilities prior to the system being fully-constructed and fielded. One robotic capability that has been heavily invested in is mobility. Zhang et al. developed a robotic platform to enable different types of locomotive apparatus to be modularly connected to isolate mobility (Zhang et al., 2002). The intent of that effort is to provide a standard platform allowing researchers to pair various modes of locomotion (wheels, tracks, etc.) to better evaluate mobility. Yue et al. also focused their research on mobility by designing and analyzing retractable-claw wheels for small ground robotics (Yue et al., 2010).

This class of small robots includes both military and domestically deployed systems. Often, these systems are tested by the technology developers, themselves. For example, Aoyama et al. have produced specific test apparatus and scenarios to evaluate the mine detection robot they developed (Aoyama et al., 2007). They focused on measuring technical performance of the mine detection system in both a controlled environment where terrain variables could be manipulated and in a more operationally-relevant environment. Likewise, Jian-Jun et al. created unique test plans to measure the quantitative performance of the semi-autonomous SUPER-PLUS Explosive Ordnance Disposal (EOD) robot system (Jian-Jun et al., 2007). Frost et al. took a broader approach in measuring the performance of the PackBot man-portable unmanned ground vehicle they created (Frost et al., 2002). The PackBot, designed to support military and Urban Search and Rescue (US&R) operations, underwent specific tests at a controlled facility to extensively evaluate its mobility and durability. The PackBot-specific tests output pass/fail criteria over specific challenges as opposed to yielding more quantitative data. Additionally, operational testing was conducted on the PackBot in an actual environment when it was deployed at the World Trade Center shortly after 9/11. Evaluation personnel collected situational awareness and communications performance data at this event. Zhang et al. created some very basic tests to get some initial performance feedback on its reconfigurable US&R robot (Zhang et al., 2006). Their tests focused on kinematic analysis and locomotion since their primary contribution was the system's reconfigurability. Their test design was focused on capturing technical performance of physical implementations without considering the HRI element. Nourbakhsh et al. take a different approach by devising an architecture to test US&R implementations across a combination of real-world and simulation-based testing (Nourbakhsh et al., 2005). Their architecture is built upon the notion of transforming a physical robot into a robot agent to interact within a multi-agent system. This enables robotic agents to be tested in a simulated US&R environment at a fraction of the cost compared to physical testing (Balakirsky et al., 2006). These non-standardized tests are highlighted by a lack of consistency and uniformity.

In an example of evaluation testing coordination, there is a standard array of tests aimed at evaluating Urban Search and Rescue (US&R) and bomb disposal robots across a range of operational scenarios (Jacoff and Messina, 2007c; Jacoff et al., 2003; Messina, 2009; Messina and Jacoff, 2007). The array includes test suites created to assess different system capabilities including mapping, mobility, communications, directed perception, grasping dexterity, visual acuity, etc (Jacoff and Messina, 2007b; Remley et al., 2007; Scrapper et al., 2009). Tests used to assess US&R robot mobility are random stepfield pallet arrangements designed to challenge the systems as they attempt to traverse varying terrains (Jacoff et al., 2008). Stepfield pallets are designed to represent a complex terrain or debris that is describable, reproducible, and repeatable for evaluating robotic technologies. The stepfield pallet layouts are used to both directly test a robot's mobility and as a secondary test to see how they impact system performance while the robot is attempting to complete another task (such as grasp an object). The test specifications are complete with variables (including human operators, stepfield arrangements, etc.) which determine the test conditions. Typically, the conditions change as the evaluation goals evolve. Standardized testing is being emphasized in this area based upon a combination of need and an emergence of resources.

Researchers have recognized the necessity and value of collaborating in the development of standard US&R robot test methods and bolster this initiative through joint robot response exercises (Jacoff and Messina, 2007a). These events have strengthened communication between test designers, technology developers and robot end-users by immersing them in both controlled test methods and operational scenarios. Everyone gains from participating; test designers learn what was successful and what needs improvement in their test methods; technology developers better understand the operational environments and the needs of the end-users; and end-users can gain a greater understanding of what the robots are capable in addition to contributing to the test method designs.

Competitions have grown as a means of strengthening technology development and evaluating system performance. Researchers at the National Institute of Standards and Technology (NIST) have administered the Rescue Robot Competition at the American Association for Artificial Intelligence's (AAAI) conference from 2000 through 2004 and the RoboCup Rescue Robot League from 2002 to present (2012) as a means to design and refine the physical test methods while getting valuable performance data from its competitors (Jacoff et al., 2000; Jacoff and Tadokoro, 2005; Jacoff et al., 2003). RoboCup also saw the birth of the RoboCup Rescue Virtual League that enabled researchers to deploy their virtual robots in simulated US&R environments to score their overall performance including their ability to autonomously generate maps (Balaguer et al., 2009). These events have fostered international collaboration in developing both state-of-the-art implementations that can be ported to operational scenarios and refining performance metrics to appropriately score the systems. These competitions are classified as ranked events with objective scoring (Yanco, 2001). As a practical matter, competitions act as a means of de-facto testing for the participants.

It is evident that both government and private institutions have devoted a substantial amount of resources into the research and development of methods and frameworks to effectively, efficiently and thoroughly evaluate the performance of maturing and intelligent systems. Most of these test design frameworks have been sufficient to evaluate their specific technologies and attain project-specific goals. Yet, no single framework has been identified as being suitable to appraise quantitative and qualitative performance across a range of technologies, incorporating both human-controlled and autonomous capabilities.

The rapid emergence of so many unique, advanced and intelligent systems and the greater need to get technologies out to the end-users motivates efficient testing to speed the pace of development and validate the final implementations. Unfortunately, efficient test planning is a laborious and challenging process due to system complexity. Another obstacle the evaluation designers face is that these test planning activities are done manually. This often leads to time-consuming re-design activities as additional information is obtained. This additional information can take the form of evolving stakeholder preferences which can greatly impact the direction of an evaluation. It is prudent for the evaluation designer to produce a plan that satisfies all of the key stakeholders based upon their stated preferences.

1.2. Test Planning Compared to Testing in Product Development

A test plan is defined as a "document detailing a systematic approach to testing a system such as a machine or software" (Test Plan, 2011). Given this definition and drawing upon prior evaluation design experience (Schlenoff et al., 2007; Schlenoff et al., 2010; Weiss & Schlenoff, 2011; Weiss et al., 2006), test planning includes the the design of evaluation blueprints. A blueprint is defined as the specifications created as the result of test planning that specify the key characteristics of a test event. These blueprints lay out the strategy by which a technology will be tested. These blueprints turn into test plans when evaluation designers add detailed test specifications. Blueprints provide the evaluation stakeholders with the key feasible and desirable test characteristics. Test plans contain the detailed workflow that can list out evaluation procedures, personnel responsibilities and even logistics (e.g. specific placement of evaluation props, execution time of evaluation, etc.).

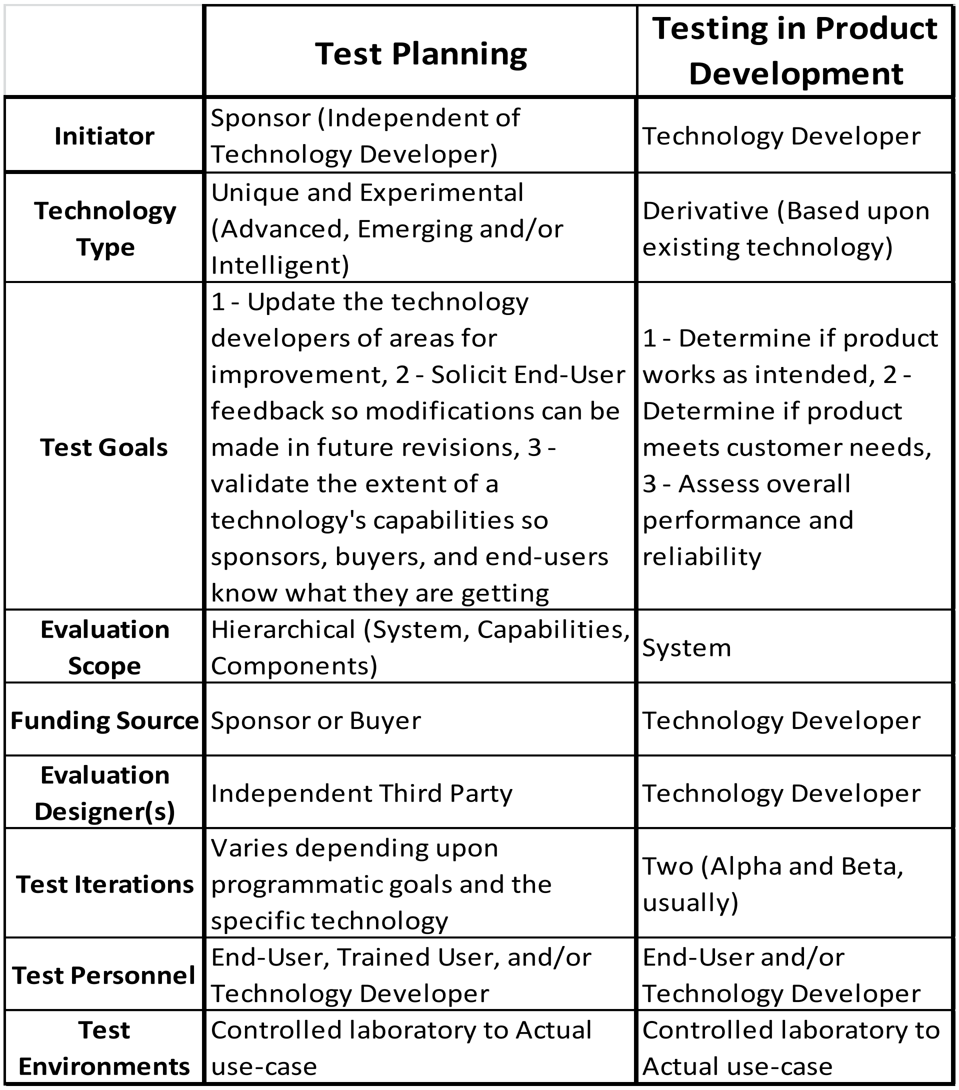
Test planning is usually initiated by an individual's or group's desire to understand the performance of a specific technology. Performance is then interpreted and translated into specific metrics of the technology. Those tasked with designing the evaluation are then responsible for understanding the purpose of the technology, who the eventual users will be or who the current users are, what type of training they need to adequately use the technology, the typical operating environments (along with those within which the technology could reasonably be tested), and the tools necessary to capture the desired performance metrics. Test planning is a non-trivial exercise that can be both time-consuming and resource-intensive. The ultimate goal of test planning is to produce a one or more evaluation blueprints aiming to evaluate specific the technology, its constituent components and/or capabilities by capturing data to generate targeted quantitative and/or qualitative measures.

Test planning should not be confused with the testing conducted during new product development. New product development is defined as the "complete process of bringing a new product to market" (New Product Development, 2012). Dieter and Schmidt define the product development process as a six step process which includes a "Testing and Refinement" phase (Dieter and Schmidt, 2009). This phase is composed of alpha and beta testing. Alpha testing evaluates prototypes that are made to the exact dimensions and specifications of the design, yet are not necessarily manufactured using the processes or tooling that are planned for mass production. The ultimate goal of alpha testing is to determine if the product works as intended and if it will satisfy the most important customer needs. Beta testing is evaluating products manufactured to both specification and using the processes that are planned for mass production. Products in beta testing undergo intense in-house testing by developers in addition to be testing in actual use-case environments by a representative set of the target customer base. The goal of beta testing is to assess the performance and reliability of the product before it goes to the general market.

Test planning and testing during the product development process are different from one another and involve different strategies. Table 1 presents a comparison between test planning and testing in product development. One significant difference between the two is that a technology created in the product development process is being developed for the primary purpose of bringing profit to the technology developer and shareholders. This contrasts with the goals of developing technologies whose evaluations are designed using test planning.

Technology development here is usually motivated by a government effort to improve efficiency and/or safety of personnel conducting specific tasks.

Table 1 - Comparison between Test Planning and Testing in Product Development



	Test Planning	Testing in Product Development
Initiator	Sponsor (Independent of Technology Developer)	Technology Developer
Technology Type	Unique and Experimental (Advanced, Emerging and/or Intelligent)	Derivative (Based upon existing technology)
Test Goals	1 - Update the technology developers of areas for improvement, 2 - Solicit End-User feedback so modifications can be made in future revisions, 3 - validate the extent of a technology's capabilities so sponsors, buyers, and end-users know what they are getting	1 - Determine if product works as intended, 2 - Determine if product meets customer needs, 3 - Assess overall performance and reliability
Evaluation Scope	Hierarchical (System, Capabilities, Components)	System
Funding Source	Sponsor or Buyer	Technology Developer
Evaluation Designer(s)	Independent Third Party	Technology Developer
Test Iterations	Varies depending upon programmatic goals and the specific technology	Two (Alpha and Beta, usually)
Test Personnel	End-User, Trained User, and/or Technology Developer	End-User and/or Technology Developer
Test Environments	Controlled laboratory to Actual use-case	Controlled laboratory to Actual use-case

1.3. *Test Plan Design Iteration*

The timing of test plan design varies based upon the nature of the program, the technology, etc. Test plan design does not occur at a single instant in the life of a program. Like all design, creating test plans is an iterative process. Designing test plans requires many iterations in the weeks and months leading up to an evaluation based upon changing sponsor preferences, resource availability (or unavailability), technological breakthroughs (or delays), etc. Test plan design becomes even more iterative between scheduled evaluations. After observing the technology in action, gauging the users' perceptions, analyzing the data, etc., the evaluation designers have a baseline of comparison. The evaluation designers can identify successes, failures, and shortcomings in the test plans so adjustments can be made for future test events. This phenomenon is illustrated through a NIST testing experience.

Personnel at NIST were tasked by the Department of Defense (DoD) to evaluate emerging speech-to-speech (S2S) translation technology (Schlenoff et al, 2009; Weiss and Schlenoff, 2010; Weiss et al, 2006). The goal of this DoD program was to demonstrate capabilities to rapidly develop and field free-form, two-way, translation systems that enable speakers of different languages to communicate with one another in real-world tactical situations without an interpreter. Figure 1 depicts the speech-to-speech translation process. The S2S software is composed of three critical and sequential components; Automatic Speech Recognition (ASR) to turn English speech (or the target language) into English text; Machine Translation (MT) to translate English text into the foreign language text; and Text To Speech (TTS) to turn the target language text into corresponding speech. The primary use cases of these translation technologies involve US military personnel conversing with local foreign language speakers. The NIST team's responsibilities included analyzing the performance of the technologies by designing and executing multiple technology evaluations and assessing the results. Between 2006 and 2010, the NIST evaluation team designed and implemented seven evaluation test events, each of which last a week.

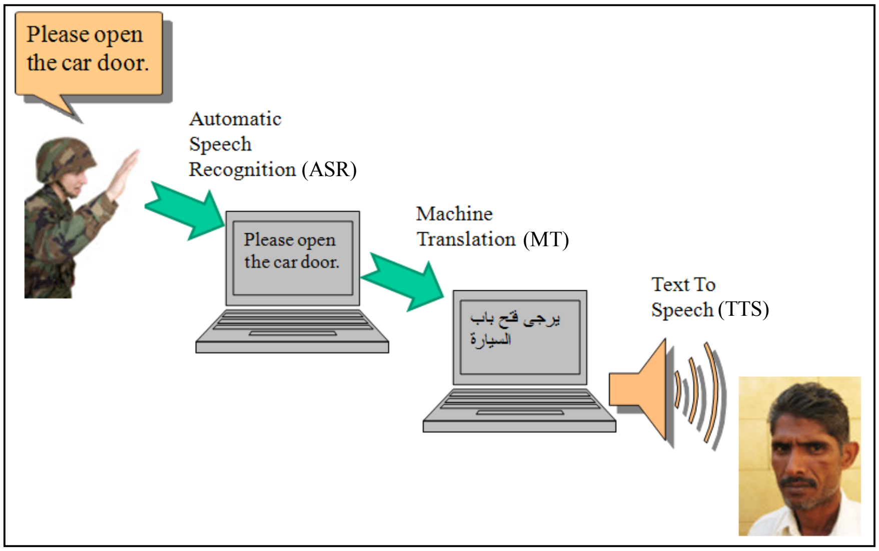

Please open the car door.
Automatic Speech Recognition (ASR)
Please open the car door.
Machine Translation (MT)
يرجى فتح باب
السيارة
Text To Speech (TTS)

Figure 1: How Speech-to-Speech Translation Works (Schlenoff et al., 2010)

The DoD set forth two main evaluation objectives for testing S2S translation technology. They are listed below (Weiss et al., 2008).

- System usability testing - Provide overall scores and assessments to the capabilities of the whole system.
- Software component testing - Evaluate individual components of the system to see how well they perform in isolation.

This program is like many other government programs in that the constituent technology was funded and evaluated in phases. Each phase includes one or more technology developers tasked to create a specific technology. This technology is evaluated one or more times (as stipulated by the program manager) to capture performance metrics. The performance of each technology is considered when the program manager is determining who will be funded for the following phase(s).

Table 2 presents the timeline of the seven technology evaluations that occurred between 2007 and 2010. The table provides a glimpse into the iterations in the test design process. The information presented in this table an overview of the interations: only the first three evaluations are presented with a moderate amount of detail while the remaining four are discussed from a broader perspective. The details of these evaluations have already been extensively documented (Schlenoff et al., 2010; Schlenoff et al., 2009; Weiss and Menzel, 2010; Weiss and Schlenoff, 2010; Weiss and Schlenoff, 2009; Weiss et al., 2008). Note that the 2007 evaluations occurred under Phase A, the 2008 evaluations occurred under Phase B and the remaining evaluations occurred in Phase C of the program. Most of the significant test plan revisions occurred when transitioning from one phase to the next.

Table 2 - Technology Evaluation Timeline

	DATE	SPECIFIC EVALUATION GOALS	TEST PLAN SUMMARY
PROGRAM PHASE A	2007 - January	Capture baseline performance of the laptop-based technologies translating between English and Foreign Language A when used by potential end users.	Created live lab and field evaluations enabling English and foreign language A speakers to converse using the technology in both controlled and simulated environments.
		Capture baseline performance of the three S2S software components translating in both directions (English to Foreign Language and Foreign Language A to English)	Tested individual software components by feeding in specific component inputs and evaluating the outputs.
	2007 - July	Capture current performance and measure improvement (from the 2007-January data) of the laptop-based technologies under conditions similar to 2007-January	Similar to 2007-January
		Capture current performance and measure improvement (from 2007-January data) of the three S2S software components under condititions similar to 2007-January	Similar to 2007-January
		Capture system performance of the laptop-based technologies translating between English and Foreign Language B when used by potential end users. Technology developers were only provided with 90 days to prepare their systems for this new language.	Created live lab evaluations enabling English and foreign language B speakers to converse using the technology in controlled environments
		Capture three S2S software components translating in both directions between English and Foreign Language B	Tested individual software components by feeding in specific component inputs and evaluating the outputs.
PROGRAM PHASE B	2008 - June	Capture current performance and measure improvement (from the 2007-July data) of the laptop-based technologies under conditions similar to 2007-January. Note the technology developers were able to improve their technologies from prior evaluations with both time and additional training data.	Similar to 2007-January except for the following improvements: 1) the live evaluation scenarios were based upon more recently collected training data
		Capture current performance and measure improvement (from 2007-July data) of the three S2S software components under conditions similar to 2007-January.	Similar to 2007-January except for the following improvements: 1) the test data set input into the components was new to be representative of the new training data.
		Capture baseline performance of "utility-based" technologies (S2S technologies running on systems more portable than laptops) when used by English and Foreign Language A speakers	Created live field evaluations in more operationally-relevant environments (compared to 2007-January/July). Enhanced evaluation scenarios were used in this test to elicit more natural, unbounded dialogue from the speakers
	2008 - November	Still focused on English <-> Foreign Language A and sought to measure improvement in all tests from 2008-June	Similar tests to 2008-June with some strategic test plan improvements.
PROGRAM PHASE C	2009 - June	Now focused on English <-> Foreign Language C, smaller form-factor utility platforms and more operationally-relevant evaluations. Comparisons are to be made between Foreign Languages A (2008-Nov) and C when being translated to and from English.	Software testing still occurs analyzing individual elements. Live evaluation is now conducted on a military base in a tactical environment. No lab evaluations are conducted.
	2010 - April	The evaluation focus shifts to test English <-> Foreign Language D and a smartphone platform is selected for technology deployment. Comparisons are to be made between Foreign Languages A (2008-Nov), C (2009-Jun), and D.	Software testing still occurs analyzing individual elements, but the software is now run off of the smartphones as opposed to laptops. Both live (controlled environment) and field (simulated environment) are conducted with the technology operating on smartphones.
	2010 - August	The evaluation focus returns to English <-> Foreign Language C. Comparisons are made to Foreign Languages A (2008-Nov), D (2010-April), and C (current data only)	Similar to 2010-April

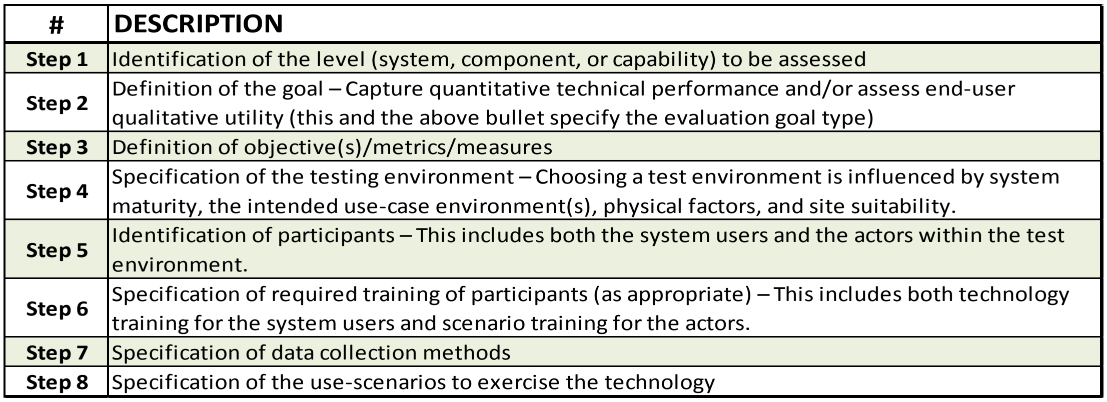
The S2S evaluation timeline presented in Table 2 encompasses over a dozen test plans. One specific test plan iteration is described here; it occurred between the 2007-July and 2008-June test events. The 2007-July evaluation data revealed that the test speakers (both English and foreign language) felt over constrained by the evaluation scenario format. They perceived the conversations to be unnatural. This observation of unnatural dialogue meant that the evaluation team was not getting the necessary representative data from evaluation scenarios and a test plan iteration was warranted. The evaluation design team altered the test scenarios for the 2008-June test event to promote greater realism (Weiss and Menzel, 2010). The 2008-June evaluation scenarios received positive feedback from the participants. This test plan re-design example highlights the potential benefit of a systematic test plan design tool. The availability of such a tool would enable the evaluation design team to be more effective by reducing the time required to reiterate the test plans. Improving effectiveness is a significant priority considering most test plans are subject to frequent redesigns.

1.4. Example Test Plan Design Framework

NIST personnel created the System, Component, and Operationally-Relevant Evaluation (SCORE) framework to assess the performance of advanced and intelligent systems (Schlenoff et al., 2007; Schlenoff et al., 2009; Weiss and Schlenoff, 2008). The author of this proposal is a co-creator of SCORE. SCORE provides a set of guidelines to aid test designers in creating evaluation plans. SCORE has been successfully applied to fifteen evaluations across several technologies (Schlenoff et al., 2010; Schlenoff et al., 2006; Weiss and Schlenoff, 2010; Weiss et al., 2008; Weiss et al., 2006). SCORE has yielded significant qualitative and quantitative data which has proven valuable to system developers, evaluation designers, potential end-users and funding sponsors. This work will draw upon SCORE's success to introduce a new evaluation framework that will automatically generate evaluation test plans. An example test plan output from one SCORE framework application will be presented in this section. Discussion of these successful plans is important to further illustrate the necessity of creating a new framework.

The NIST team's application of SCORE enabled the creation of multiple test plans to evaluate the speech-to-speech (S2S) translation technologies. These test events can be classified by different evaluation goal types that evaluate the technology at different levels capturing quantitative and/or qualitative data (Weiss and Schlenoff, 2008). Overall, SCORE aided the test designer by providing them with evaluation elements they should specify when creating test plans for a specific goal type. These elements are presented in Table 3 (Schlenoff et al., 2007).

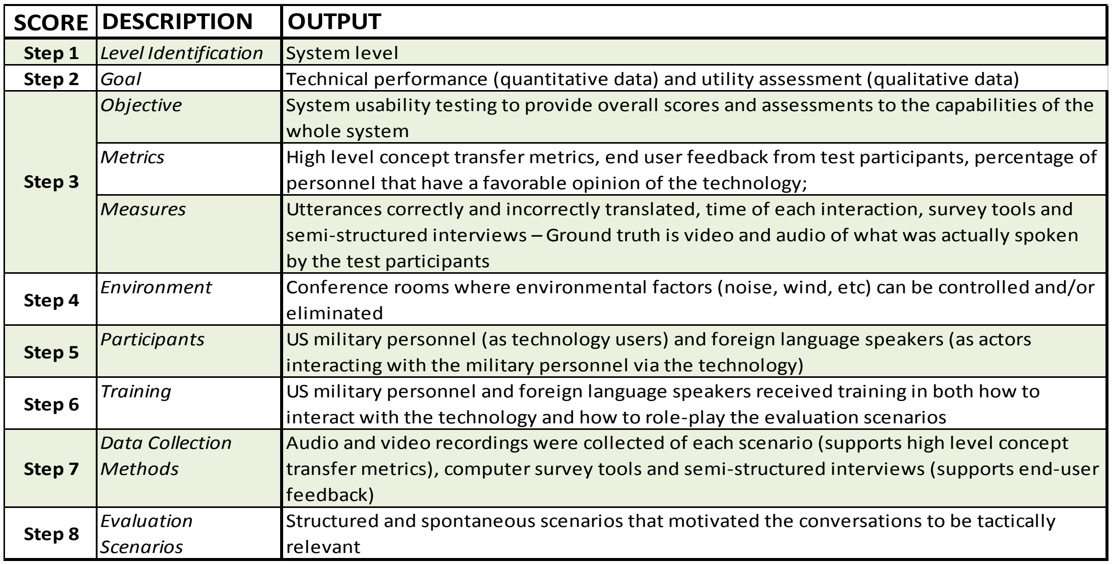
Table 3 - SCORE Test Plan Generation Process



#	DESCRIPTION
Step 1	Identification of the level (system, component, or capability) to be assessed
Step 2	Definition of the goal – Capture quantitative technical performance and/or assess end-user qualitative utility (this and the above bullet specify the evaluation goal type)
Step 3	Definition of objective(s)/metrics/measures
Step 4	Specification of the testing environment – Choosing a test environment is influenced by system maturity, the intended use-case environment(s), physical factors, and site suitability.
Step 5	Identification of participants – This includes both the system users and the actors within the test environment.
Step 6	Specification of required training of participants (as appropriate) – This includes both technology training for the system users and scenario training for the actors.
Step 7	Specification of data collection methods
Step 8	Specification of the use-scenarios to exercise the technology

Following SCORE guidelines, evaluators designed multiple test plans for the speech-to-speech translation technology. Over a half dozen different test plans were specified throughout the course of this multi-year effort and one of most frequent test plans, known as Lab Evaluations, is presented in Table 4 according to the SCORE framework (Weiss and Menzel, 2010; Weiss and Schlenoff, 2009).

Table 4 - SCORE Test Plan for the S2S Translation Lab Evaluations



SCORE	DESCRIPTION	OUTPUT
Step 1	*Level Identification*	System level
Step 2	*Goal*	Technical performance (quantitative data) and utility assessment (qualitative data)
Step 3	*Objective*	System usability testing to provide overall scores and assessments to the capabilities of the whole system
	Metrics	High level concept transfer metrics, end user feedback from test participants, percentage of personnel that have a favorable opinion of the technology;
	Measures	Utterances correctly and incorrectly translated, time of each interaction, survey tools and semi-structured interviews – Ground truth is video and audio of what was actually spoken by the test participants
Step 4	*Environment*	Conference rooms where environmental factors (noise, wind, etc) can be controlled and/or eliminated
Step 5	*Participants*	US military personnel (as technology users) and foreign language speakers (as actors interacting with the military personnel via the technology)
Step 6	*Training*	US military personnel and foreign language speakers received training in both how to interact with the technology and how to role-play the evaluation scenarios
Step 7	*Data Collection Methods*	Audio and video recordings were collected of each scenario (supports high level concept transfer metrics), computer survey tools and semi-structured interviews (supports end-user feedback)
Step 8	*Evaluation Scenarios*	Structured and spontaneous scenarios that motivated the conversations to be tactically relevant

One challenge faced by the NIST evaluation team was that SCORE had to be manually applied to create these test plans. Another obstacle was addressing changing test requirements stemming from numerous factors including:

- Evolving technology states (e.g., were all of the promised capabilities going to be available for the evaluation?)
- Updated programmatic goals (external sponsors providing new areas of focus, etc.)
- Changing availability of evaluation resources (the preferred evaluation location may or may not be available, etc.)

Test plans were frequently revised since they were impacted by changing circumstances. After numerous week-long evaluations, it was observed that relationships existed between some of the evaluation elements. Exploiting these relationships during the test design process would have enabled the evaluation team to design the tests more efficiently.

1.5. Research Focus

This research created an evaluation methodology and algorithm that automatically designed test plan blueprints to collect data for quantitative and/or qualitative metrics. This methodology is named Multi-Relationship Evaluation Design (MRED). MRED is an evaluation design methodology and algorithm that will take inputs from three specific categories and output test plan blueprints that specify critical elements of the test event(s) (Weiss et al., 2010).

1.5.1. Research Questions

The following questions, with associated tasks, are the focus of the research:

1. How should an evaluation test plan generator be modeled to exploit the relationships among multiple deterministic inputs and output test plan blueprints? Tasks that address this question include:
    a. Model an evaluation test plan blueprint generator and provide a formalization of the inputs and outputs and operations.
    b. Identify relationships among the inputs and determine how they influence the outputs.
    c. Implement and verify the MRED framework in software.
    d. Validate the results of MRED output against test plans generated by other methods.

2. How should MRED integrate stakeholder preferences into the design of test plan blueprints? Tasks to address this question include:
    a. Determine MRED's output format that reflects the input stakeholder preferences.
    b. Choose and implement the chosen candidate preference handling method into the existing MRED framework.
    c. Verify the preference handling method within the MRED planner by applying the inputs based upon other previously-created test plans.
3. How can the chosen preference handling method be validated?

Chapter 2: Background

It is important to understand a technology's readiness and maturity as it is developed from a concept into a fully-functional system. Background is presented on readiness and maturity since these two concepts play a role in determining when a particular technology (or constituent element) is developed enough for a specific test. Likewise, numerous test design methodologies have been researched to demonstrate the current capabilities and limitations of the existing methods. Existing methods for preference capture and handling are discussed in this chapter since this is another area that MRED leverages in its methodology and algorithm.

2.1. Technology Readiness and Maturity

"The challenge for system and technology managers is to be able to make clear,well documented assessments of technology readiness and risks, and to do so at key points in the life cycle of the program" (Mankins, p. 1216, 2009). The operational readiness of the technology's constituent elements and the system must be assessed, before a technology may be tested. This is the case whether it's a fully-functional system or has yet to have all its subsystems functional and integrated. NASA defines a systematic process to perform what is known as a Technology Assessment (TA) to establish a relevant measure of the technology's maturity (NASA Systems Engineering Handbook, 2007). The TA is composed of two parts: a Technology Maturity Assessment (TMA) and an Advancement Degree of Difficulty Assessment (AD^2). The TMA determines the maturity of a technology using NASA's Technology Readiness Level (TRL) scale (NASA Systems Engineering Handbook, 2007).

NASA developed an initial TRL scale in the 1980s and Mankins further expanded it in 1995 to its existing state with 9 levels (Mankins, 1995). Mankins defines TRLs as “a systematic metric/measurement system that supports assessments of the maturity of a particular technology and the consistent comparison of maturity between different types of technology” (Mankins, p. 1, 1995). Besides NASA, other organizations have used the TRL scale to measure technological readiness at critical program milestones (Air Force Space Command, 2008; NASA Systems Engineering Handbook, 2007). The nine TRLs are presented and defined in Figure 2.

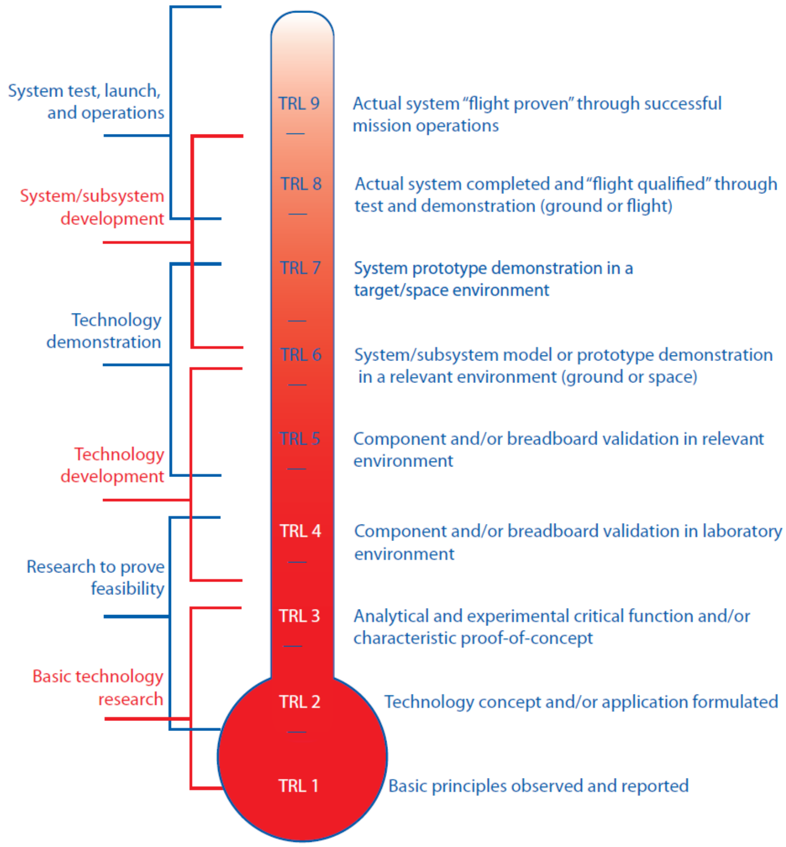

System test, launch, and operations
System/subsystem development
Technology demonstration
Technology development
Research to prove feasibility
Basic technology research
TRL 9
Actual system “flight proven” through successful mission operations
TRL 8
Actual system completed and “flight qualified” through test and demonstration (ground or flight)
TRL 7
System prototype demonstration in a target/space environment
TRL 6
System/subsystem model or prototype demonstration in a relevant environment (ground or space)
TRL 5
Component and/or breadboard validation in relevant environment
TRL 4
Component and/or breadboard validation in laboratory environment
TRL 3
Analytical and experimental critical function and/or characteristic proof-of-concept
TRL 2
Technology concept and/or application formulated
TRL 1
Basic principles observed and reported

Figure 2: Overview of the technology readiness level scale (NASA Systems Engineering Handbook, p. 296, 2007)

Although the descriptions of the TRLs appear clear, challenges frequently arise when attempting to assign a specific level to a technology. NASA condenses the TRL assignment task to three steps (NASA Systems Engineering Handbook, 2007):

1. Define the terms to be used to maintain a consistent set of definitions throughout the life of program.
2. Quantify "judgment calls" given past experience. This includes detailing what has been previously done with respect to form, fit, and function of the technology.
3. Determine who, among the project team, is the ideal candidate to make the "judgment call" regarding the status of the technology.

The TRL methodology of assigning readiness to developing technologies has been adopted and adapted by organizations that are involved in a much greater range of system development. In this case, each of the individual TRL assignment definitions are modified to best suit the organization and the specific system development program.

To reiterate, the Technology Readiness Levels are a means of assigning a readiness level to the entire system in order to complete its intended operational objectives. This is implied from Figure 2. TRLs 4 and above can only be assigned following a structured evaluation for the entire system. This approach is clearly useful to NASA given the commonality of operating environment (i.e. space).

Sometimes it's relevant to calculate readiness for a technology's subsystems. Then a system's constituent elements have to be classified by the TRLs. NASA's approach is to assign the same TRL to the technology as the lowest TRL of its constituent pieces. If a technology is made up of three subsystems (A, B, & C), where the TRL of A is 5, the TRL of B is 4, and the TRL of C is 6; then the TRL of the technology would be 4 since this is the lowest level of the subsystems (NASA Systems Engineering Handbook, 2007). Once the components are assigned TRLs, then TRLs may be assigned to the subsystem levels. Some components and subsystems of a system will never be independently demonstrated in relevant and/or operational environments. When a technology is demonstrated at a specific TRL, NASA's TMA process continues by determining the AD^2. The AD^2 is defined as the process to "…develop an understanding of what is required to advance the level of system maturity" (NASA Systems Engineering Handbook, p. 266, 2007).

Researchers looking to expand the TRL concept into other areas (e.g. DoD), have rejected the sufficientcy of TRLs and extended this work to develop additional classifications. This expansion has led to the development of System Readiness Levels (SRL) and Integration Readiness Levels (IRLs) (Tetlay and John, 2009; Sauser et al., 2006). The objective of SRLs is to index maturity so it correlates with systems engineering management principles during design and verification of subsystem technologies. The goal of IRLs is to assess the maturity of the integration points between multiple subsystems interacting with one another in a technology. The SRL is more relevant to this research effort and is discussed further.

Tetlay and John define SRLs as having "been developed as a project management tool to capture evidence, and assess and communicate System Maturity in a consistent manner to stakeholders" (Tetlay and John, p. 2, 2009). SRLs represent a split between the two concepts of maturity and readiness. System maturity is defined as "verification within an iterative process of the system development lifecycle and occurs before system readiness, i.e. the system must first be fully 'mature' before it can be 'ready' for use" (Tetlay and John, p. 3, 2009). Tetlay and John defend the notion that system maturity and system readiness are two distinct concepts that address entirely separate questions within the scope and context of system development (Tetlay and John, 2009). Figure 3 illustrates where system maturity and system readiness are applied in the system development lifecycle.

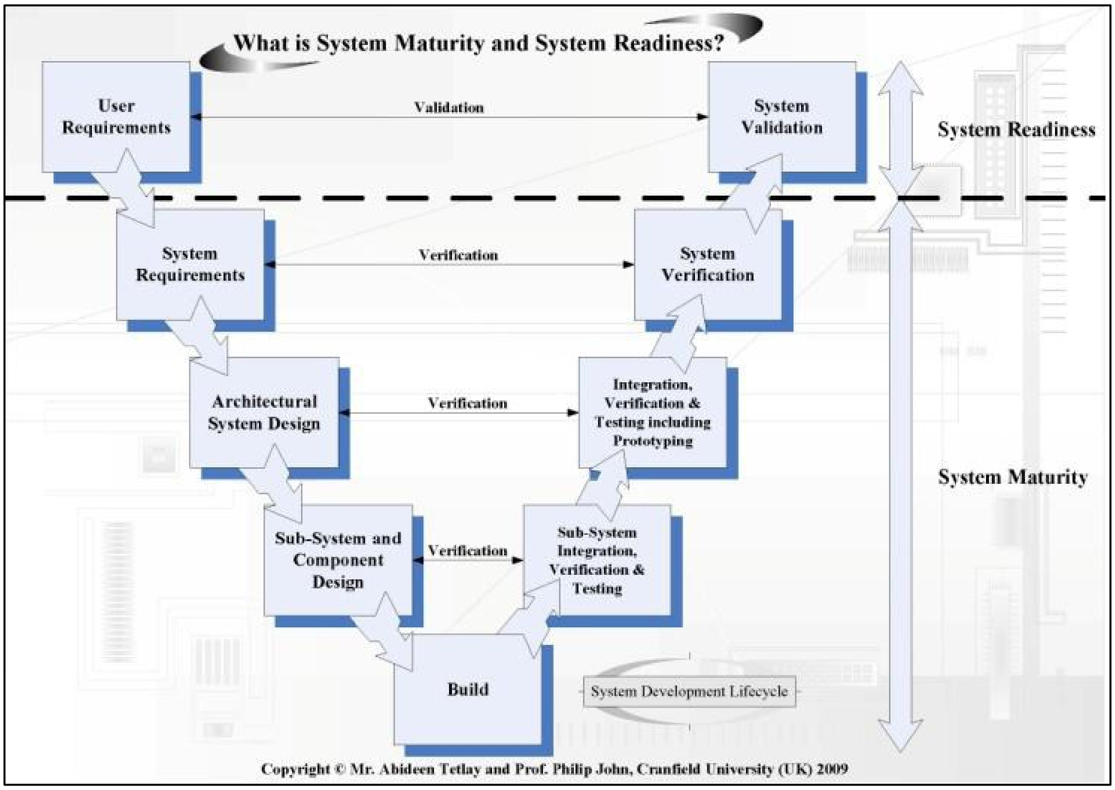

What is System Maturity and System Readiness?
User Requirements
Validation
System Validation
System Readiness
System Requirements
Verification
System Verification
Architectural System Design
Verification
Integration, Verification & Testing including Prototyping
System Maturity
Sub-System and Component Design
Verification
Sub-System Integration, Verification & Testing
Build
System Development Lifecycle
Copyright © Mr. Abideen Tetlay and Prof. Philip John, Cranfield University (UK) 2009

Figure 3: System Maturity and Readiness (Tetley and John, 2009)

System maturity begins to evolve at the systems engineering product realization process (see Section 3.1) and ends when the system has completed verification testing.

During verification testing the constituent components and subsystems are labeled according to three maturity states (Tetlay and John, 2009):

- System is Immature – the product realization process has not started yet.

- System Maturity is in Progress – working your way through the product realization process of systems engineering. Subsystem technologies are being verified.
- System Maturity has been Achieved – the design, development and testing of the system is now complete, fully mature and validation can begin. To achieve System Maturity the system must be verified against the System Requirements.

Once the system is fully mature, then system readiness testing begins.

System readiness is defined as "the validation and Boolean (either the system is 'ready' for use or not) aspect of the system development and overall lifecycle and occurs after system maturity…" (Tetlay and John, p. 3, 2009). Tetlay and John use a nine level index for SRLs. Other researchers have selected different SRL indexes. For example, Sauser et al. define SRLs to have five levels (Sauser et al., 2006).

2.2. Test Design Methodologies

Efforts have been put forth to design and implement test planning systems for complex emerging and advanced technologies across many domains. Test plan strategies have been devised to test a range of technologies with varying levels of autonomy and collaboration, both between humans and robots and amongst robots, themselves. Test plan generators have been developed to create tests to evaluate specific technologies, produce different types of data, assess either individual capabilities or systems as a whole, occur in simulation or the physical world, etc. This section focuses on a background literature review of test design methodologies and cites evidence why the current methods are insufficient.

Human-Robot Interaction (HRI) and Human-Computer Interaction (HCI) is an important element to consider in testing an advanced or intelligent system. NIST personnel have devised and implemented numerous strategies throughout various field studies of Urban Search and Rescue (US&R) and Explosive Ordnance Disposal (EOD) Robots, examining the successes and shortcomings of the technologies' interfaces with humans (Scholtz et al., 2004; Scholtz et al., 2005; Scholtz et al., 2006). Their work dissects the user experience with the human-robot interfaces to determine what actions led to successes and failures. NIST researchers have also created an evaluation method to capture usability of representative end-users operating US&R robots (Stanton et al., 2006). To determine a sufficient test method to capture usability, test designers identified several essential tasks, collected data in a pilot study, and further iterated upon their test plans. Focusing on a specific aspect of human-robot interfaces, an assessment tool was devised that focused on evaluating the situation awareness of an operator as they interact with an autonomous ground vehicle (Scholtz, 2005). The assessment tool supported several experiments that occurred in simulation where researchers could assess the situation awareness of their test subjects. Discussion was presented on metrics and methodologies for evaluating human information interaction received by intelligence analysts (Morse et al., 2005). The tests focused on capturing the perceptions of analysts when given certain pieces of intelligence information. One group of researchers took a different approach in that they proposed to develop and execute an evaluation focused on testing and comparing two different control system interfaces for military technologies (Bialczak et al., 2002). Their testing is planned for a highly-controlled, laboratory-style environment where all variables will be fixed except for that of the two control system choices so direct comparisons can be made among the collected quantitative and qualitative metrics. Bialczak et al. examine the technology at the system level and do not discuss any efforts to test the technologies at their constituent capability or component levels. Another team of researchers developed a complete simulation environment to evaluate human-robot team performance (Freedy et al., 2006). The simulation is designed to test mixed manned and unmanned ground vehicle teams in both training and real world operational military operations. Freedy et al. were very focused on capturing mixed initiative team performance metrics in simulation (i.e., metrics that assess the cooperation and/or interaction between a human operator and a robot). Yet, their test does not cover performance metrics outside of this scope or evaluate these technologies in the physical world.

The importance of evaluating human-robot interaction led to the development of several methodologies focused on this specific area. One such evaluation framework was developed by a group of researchers to address usability, social acceptance, user experience, and societal impact (USUS) to ultimately enhance the way humans interact with robots (A. Weiss et al., 2009); the USUS framework is a multi-level model aiming at its four core evaluation factors mentioned previously. This framework appears successful at testing a technology against these four factors, yet it is limited to only testing human-robot interaction.

Evaluation methodologies have been developed to test a variety of unmanned systems and vehicles beyond the human-robot interaction. An evaluation framework was created to test mobile robots for planetary exploration across applicable terrains, but it did not factor the element of human interaction and was designed with a mission-specific emphasis (Sukhatme and Bekey, 2005). More generically, a conceptual framework for the development of technical features and operational performance of unmanned systems has been devised (Schenk and Wade, 2008). Their work endorses multiple technical performance tests at the atomic, aggregate, and task levels. Atomic tests are focused on specific abilities; aggregate tests are focused on testing several coupled capabilities; tasks are focused on the evaluating the technology's ability to employ multiple capabilities to accomplish a goal.

The military community has also invested in the development and implementation of test frameworks. The United States (US) Army has assessed network-enabled systems, although this has required the usage of multiple methodologies as opposed to using a single unified framework (Conley, 2009). Four strategies were employed with each being capable of generating evaluations at specific technology test levels. All four must be applied to produce comprehensive assessments.

The US Army has also supported the development of the Unmanned Autonomous System Testing (UAST) methodology which is intended to evaluate the intelligence of unmanned autonomous systems (Thompson, 2008). The UAST framework can evaluate both virtual and physical systems, yet its current work hasn't focused on producing qualitative measures specified by the users and only specifies pass/fail measures based upon mission tasks. Other military researchers are proposing to apply the Mission Based Test and Evaluation (MBT&E) framework to support testing of unmanned and autonomous systems (Djang and Lopez, 2009). This framework has the advantage of designing rigorous and real-world testing based upon expected military missions. Another benefit of MBT&E framework is that testers can evaluate autonomous and collaborative vehicles in simulation to identify points of failure. Testers can use this data to inform the technology developers so future technology iterations can avoid failure at these instances. The drawback is that pass/fail data does not provide detailed performance criteria.

Evaluation frameworks have also been devised to focus on testing specific intelligent algorithms and/or agents. One such framework was created to specifically assess vehicle motion algorithms where the test output produced comprehensive quantitative data (Calisi et al., 2008). This method was successful in capturing technical performance in a simulated environment but has yet to capture qualitative data from human users or assess these algorithms operating on physical systems. The same researchers created a methodology to quantitatively evaluate robot-generated maps across a range of criteria in varying environments (Calisi and Nardi, 2009). Their method proved successful yet is restricted to highly-specialized assessments of automatically-generated maps. Table 5 presents the capabilities of the test methods researched in this effort.

Table 5 - Capabilities of Existing Test Methods

CURRENT TEST METHOD CAPABILITIES											

LITERATURE REVIEW		Bodt et al., 2009	Calisi et al., 2008;
			Calisi & Nardi, 2009;
			Scrapper et al., 2009
				Conley, 2009	Djang & Lopez, 2009;
					Sukhatme & Bekey, 1995
						Frost et al., 2002;
						Schenk & Wade, 2008
							Jacoff et al., 2008	Messina & Jacoff, 2007	Scholtz et al., 2004;
									Scholtz et al., 2006
										Schlenoff et al., 2007;
										Schlenoff et al., 2009;
										B. Weiss et al.; 2008
											A. Weiss et al., 2009

Technology Test Level	System			✓	✓	✓		✓	✓	✓	✓
	Component	✓		✓	✓			✓		✓	
	Capability		✓	✓		✓	✓	✓	✓	✓	
Metric Type	Quantitative	✓	✓	✓	✓	✓	✓	✓		✓	✓
	Qualitative							✓	✓	✓	✓
Stakeholder Preference Capture		N	N	N	N	N	Y	Y	N	Y	N

2.3. *Preference Methods*

Preference capture is a topic that has been studied for decades by researchers in many fields including economics and, most recently, engineering design. In layman's terms, preference can be defined as *the power, right, or opportunity of choosing*[1] and as a *positive regard for something*[2]. As used in engineering decision-making, Hazelrigg defines preference as "the desire on the part of the decision maker to obtain one outcome over another" (Hazelrigg, pp. 15-16, 2012). A preference is a decision made when neither empiricity nor objectivity are present, such as when someone states "the outdoor site is nicer than the indoor site" (Cecconi et al., 2007). 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 years of study. Erlandson examines several different evaluation methodologies where one approach captures attribute preferences from a sample user population while another approach looks into aggregating these preferences into a representative ranking (Erlandson, 1978). Wu proposes a combined data envelope analysis and fuzzy preference relation ranking method to evaluate design alternatives with unique attributes (Wu, 2009). His method accounts for

[1] http://www.merriam-webster.com/dictionary/preference
[2] http://www.merriam-webster.com/thesaurus/preference

benefit and cost attributes, yet relies upon specific quantitative attribute values and does not account for qualitative preferences.

There are numerous challenges to effectively capturing group preferences including (Thurston, 2011):

- 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
- Weighing 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
- 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 (Thurston, 2011).

2.3.1. Measurement Scales

Preferences must be captured on a measurement scale[3] in order to be meaningful. Stevens defines four scales: nominal, ordinal, interval, and ratio (Stevens, 1946). Stevens specifically defines these as:

[3] "A measure or grade is a message that has strictly nothing do with a utility… A measure provides a common language, be it numerical, ordinal or verbal, to grade and classify" (Balinski & Laraki, pp. 8720-8721, 2007a)

- Nominal – This scale indicates the least constrained assignment of numerals. This scale includes labels or types designated as numbers (e.g. members of a team, etc.). Mode is one of the few statistical measures that can be captured from data on a nominal scale. Deemed the most basic of scales, the nominal scale serves to identify an element's membership within a group.
- Ordinal – This type of scale associates elements with a ranking sequence. "The ordinal scale arises from the operation of ranking order" (Stevens, p. 679, 1946). Ordinal scales are predominantly used to preserve the order of elements within a group. Ordinal scales do not distinguish the distance (i.e. strength of preference) between neighboring elements. Median is a measure that can be obtained from data presented on an ordinal scale.
- Interval – An interval scale is a quantitative numerical scale. Interval scales support more statistical measures than an ordinal scale. For example, mean and standard deviation are common measures that can be obtained from an interval scale. "The zero point on an interval scale is a matter of convention or convenience…" (Stevens, p. 679, 1946).
- Ratio – Ratio scales only exist when there are operations to support the four relations of "equality, rank-order, equality of intervals, and equality of ratios" (Stevens, p. 679, 1946). Ratio scales are common in physics and enable numerical values to be transformed such as centimeters to meters. "An absolute zero is always implied [on a ratio scale], even though the zero value on some scales (e.g. Absolute Temperature) may never be produced" (Stevens, pp. 679-680). All of the afore-mentioned statistical measures are applicable for ratio scales.

Some researchers recommend relaxing the constraints that Stevens defines. These recommendations are based upon the notion of enabling researchers the opportunity to use scales and statistical measure that are most meaningful to their work. For example, Velleman and Wilkinson have taken a deeper look into Steven's development of the four measurement scales (Velleman & Wilkinson, 1993). Velleman and Wilkinson question the rigidity of Steven's scales; "…the use of Steven's categories in selecting or recommending statistical analysis methods is inappropriate and can often be wrong. They [the measurement scales] do not describe the attributes of real data that are essential to good statistical analysis." (Velleman and Wilkinson, p. 65, 1993). Velleman and Wilkinson offer several criticisms of Stevens work. Some of them include (Vellemen & Wilkinson, p. 67, 1993):

- Allowable statistical calculations for a data set should not depend upon the representation of the problem but should be concerned on the meaningfulness of the data.
- "…taxonomy [measurement scale] is too strict to apply to real-world data." Alternate taxonomies (what Stevens refers to as scales) have been proposed including one by (Mosteller and Tukey, 1977) as follows: names, grades, ranks, counted fractions, counts, amounts and balances.
- "…Stevens' prescriptions often lead to degrading data by rank ordering and unnecessarily resorting to nonparametric methods" (Velleman & Wilkinson, p. 67, 1993) An example of this criticism is in the attempt to assign measurements to an interval scale in the presence of calibration errors that impact the value measured. In this case, all measuered values do not conform to an interval scale.

According to Stevens' definitions of scales, these measurements would have to placed on an ordinal scale. However, much of the data in the information is lost on an ordinal scale, especially when the errors are small relative to the measurements and are not seen in all the measured values. The data must and statistics must be meaningful whatever scale is selected to support a specific data analysis.

Wright and Linacre look at scales from a different perspective. Their work can be summed up by the title of one of their publications "Observations are always ordinal; Measurements, however, must be interval" (Wright and Linacre, p. 857, 1989). For example, observations that map ordinal linguistic terms to an interval scale, e.g. "none," "plenty," "nearly all," and "all," can be represented as a series of steps. "None" could be zero steps on the scale; "plenty" could be one step on the scale; etc. Alternatively, "plenty" could mean 20 steps up the rating scale. Measurements are always defined on interval or ratio scales (Wright and Linacre, 1989). Measurements are defined as numbers within a set whose statistics (addition, subtraction, multiplication, etc.) can be calculated such that the results retain their numerical meaning. Wright and Linacre also note that the definition of a measurement scale's origin is somewhat arbitrary and often based upon convenience. Temperature has three measurement scales (Celsius, Fahrenheit, and Kelvin) whose origins were determined using different theoretical reasons and convenience for multiple applications (Wright and Linacre, 1989). A common theme between Wright and Linacre's research and that of Velleman and Wilkinson is meaningfulness.

Another measurement scale that is used by the community is that of the cardinal scale. The term cardinal scale is used to denote either an interval scale or a ratio scale (Fundamentals of Statistics, 2010). Researchers have captured and processed preference data on cardinal scales (Harvey & Osterdal, 2010). A cardinal scale is based upon the notion of cardinality. Cardinality implies the presence of numerical scale that implies quantifiable measurements between scale values (Fleming, 1952). Conversely, ordinality implies the presence of a ranking scale where exact measurement between scale values is unknown (van Praag, 1991).

Researchers in other domains have done statistical analysis (requiring interval-scale data) on ordinal scale data. Forrest and Andersen surveyed 12 medical journals and found that that over 120 papers published in 1982 applied statistical methods to data captured on ordinal scales (Forrest & Andersen, 1986). The article was not approving the choice of analysis, rather it suggested the need for more study on the appropriateness of using parametric statistics with ordinal scale data. Whatever scale is chosen and whatever statistics are calculated they must provide a valuable analysis of the data.

2.3.2. Rational Consensus

Research has shown that a rational consensus process should be adopted when capturing and handling expert opinions to enhance the legitimacy of the result. Ayyub states that a rational process must meet following requirements (Ayyub, 2001):

- Reproducibility – The process of capturing preferences and calculating the consensus (i.e. group decision) should be documented. This allows the process to be reproduced by peers and is consistent with the philosophy of scientific research.

- Accountability – Each individual that is providing their opinion should be identified. This minimizes the potential of individuals from voting dishonestly since they cannot hide behind anonymity.
- Neutrality – Preference capturing and handling methods should support voters in expressing their honest preferences. For example, voters could perceive the use of the median, as a method to calculate consensus, as a means to reward centrally-based preferences. In this case, some voters might perceive the use of the median as a biasing strategy.
- Fairness – All voters should be treated equally during the preference capturing process. An exception here is if voter weighting factors are used where all preferences are not weighted equally.

These requirements should be considered as preference capture and handling methods are explored.

2.3.3. Preference Capture and Handling

One method of preference capture and group decision-making is the Borda count, which is often referred to as a voting method (Dummett, 1998; Dym et al., 2002; Hazelrigg, 2012; Saari, 2006). The Borda count was developed as a method to allow a group of individuals to rank order candidates and select the 'most preferred' candidate. 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-1 points to the candidate ranked 1st, n-2 points to the candidate ranked 2nd, etc. Voters' ranks for each candidate are added together and the candidate that receives the highest score is considered the winner (or 'most preferred'). For the test design methods described in 2.2 and that captured stakeholder preferences, neither all of the evaluation stakeholders were solicited for their preferences nor were formal methods were applied to handle these preferences.

In general, the Borda Count satisfies Arrow's first four axioms yet violates Arrow's fifth axiom, *Independence of irrelevant alternatives*[4] (Dym et al., 2002). Specifically, it is susceptible to agenda manipulation (Dummett, 1998) in that it does not account for majority preferences at all. The Borda Count is strictly ordinal and it does not enable voters to delineate the strength of preference between two-sequentially-ranked alternatives. In this sense, a candidate that a voter is indifferent to would be scored the same as a candidate the voter finds least appealing (last).

Pairwise comparison (also referred to as the Condorcet method) is another method of preference capture and can be used to achieve a group decision (Dym et al., 2002; Hazelrigg, 2012). Pairwise comparison is predicated upon all alternatives being compared on a one-to-one basis. Although this method has been proven effective in some applications, it is not practical when many alternatives must be considered. Performing pairwise comparison of a large number of alternatives is a

[4] *Independence of irrelevant alternatives* (IIA) is defined as follows: 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.

time-consuming process for the decision-maker. If a decision-maker is faced with 25 different alternatives, they would have to perform 300 pairwise comparisons. Further, Arrow's Impossibility Theorem restricts aggregation of pairwise comparisons (Geanakoplos, 2005).

Cook explored both the Borda and Condorcet methods during his study of distance-based and ad hoc consensus models in ordinal preference ranking (Cook, 2006). Cook classifies these methods as ad hoc, non-elimination methods. Elimination methods include runoff voting methods which require more than one round of voting and eliminate at least one candidate after each round until the winner is selected. Cook also defines a distance function to aggregate a set of ordinal preferences (Cook, 2006). His formula determines a consensus candidate when multiple voters rank order a set of candidates. Cook also applies his distance method to achieve consensus when voters' strength of preference is determined for each candidate. A significant concern with Cooks distance-based methods is the violation of Arrow's independence of irrelevant candidates axiom. Given that candidates are rank-ordered or strength of preference is captured for every pair of candidates, it's likely that pre-existing preferences will become invalid if one or more candidates are eliminated from consideration.

Approval Voting is a method that enables voters to score each candidate with either a 0 or 1 expressing either their disapproval or approval on a nominal scale (Brams and Fishburn, 1978). There is no limit to the number of alternatives that a voter may score either 0 or 1. A voter may give their most preferred candidate a "1" and score all other candidates with a "0." Or, they can score all of the candidates they approve with a “1” and score the remaining with a “0.” For multiple voters, the greatest sum of each candidate’s score determines the winner. Unfortunately, Approval Voting does not enable the voter to express any ranking or strength of preference. Rather, they are only allowed to “approve” or “disapprove.”

Alternatively, plurality voting is a voting method that restricts a voter from only voting for a single candidate (i.e. giving them a “1” and all other candidates are then given a “0”). This method forces the voter to pick their most-preferred alternative whether or not the alternative significantly overwhelms the voter’s next most-preferred alternative. Although a voter is allowed to clearly indicate their most preferred alternative, all of the remaining alternatives are treated equally and seen as being disapproved (even if some alternatives are highly regarded).

Ayyub explores the aggregation of expert opinions using 25, 50, and 75 percentile values (Ayyub, 2001). These values are calculated based upon the number of expert opinions provided. By virture of the definition of percentiles, the median is considered to be the 50-percentile value (Ayyub, 2001). Ayyub presents computations of percentiles (including the 25, 50, and 75 percentiles calculated by the arithmetic and geometric averages) to account for the opinions of four to 12 experts. Ayyub’s approach is designed to address expert opinions and may not be relevant to MRED. MRED captures the preferences of personnel with varying knowledge levels and these individuals are not necessarily experts in their specific field.

Majority Judgment is a method that aggregates voters’ preferences to produce either a “Majority-Grade” or a “Majority-Ranking” (Balinski & Laraki, 2007a, 2007b). Majority Judgment relies upon voters expressing a grade on an ordinal scale where grades are expressed linguistically or numerically. This method is capable of producing either a rank ordering or assigning grades among a set of candidates. The rankings or grades are determined by selecting the middlemost value of each candidate's votes. Selecting the middlemost value (median) balances the number of votes above this value to the number of votes below this value. The median value is selected for an odd amount of voters. If the number of voters is even the mean of the two middlemost values is used. Balinski and Laraki offer a "simplified" rule to break ties (Balinski & Laraki, 2007b). The Majority Judgment method offers the benefit of "strategy-proofing" judges' votes. Majority Judgment guards against those voters who choose to vote strategically, not honestly, to ensure a certain candidate is ranked high or low.

Majority Judgment is used in a similar manner to rank alternatives. When ranking a set of alternatives it is plausible that more than one alternative will have the same rank. The objective of ranking requires the generation of "an ordered list from first to last and a clear winner is absolutely necessary" (Balinski & Laraki, p. 8724, 2007a). Since ranking does not permit two alternatives to share the same ranking, the method to determine the median (noted above) must be augmented. If two alternatives are shown to have equal ranks, then their majority judgment (the middlemost value) is removed from their respective sets and the median is recalculated.

Other researchers have conducted a critical evaluation of the Majority Judgment method which reveal both advantages and disadvantages. The advantages include (Felsenthal & Machover, 2008):

- Voter-expressivity – Voters are allowed to award ordinal grades to each alternative
- Anonymity and Neutrality – Voters and candidates are both treated equally
- Unanimity – If all voters award candidate x a higher grade than every other candidate, then x is considered the "winner"
- Independence of Irrelevant Alternatives – If a candidate x wins, then x would remain the winner if another candidate, y, is removed.
- Encouragement of Sincere Grading – Median calculations inspire the voters to honestly grade each alternative.

Majority Judgment reduces to Approval Voting in the presence of two grades (e.g. 'approve' or 'disapprove') (Felsenthal & Machover, 2008).

Felsenthal and Machover have also identified disadvantages with Majority Judgment. They include (Felsenthal & Machover, 2008):

- Discrepancy in tie-breaking – the application of the Balinski and Laraki's "simplified" rule and the iterative steps to break ties could lead to different results.
- Strategic grading – Circumstances may exist where voters would vote strategically. This is likely to occur if voters are aware of each other's ratings and Majority Judgment is used for ranking (as opposed to grading) where the highest ranking alternative is declared the winner.
- Violation of Reinforcement – Reinforcement states that if the same candidate is elected by separate groups of voters, then this same candidate should also be

elected when all of the groups are merged into one. Examples are shown where Majority Judgment violates the concept of reinforcement.

- Indifference and abstention – Majority Judgment treats these two states differently which can lead to adverse effects. Majority Judgment requires all alternatives to have the same number of grades. When a voter fails to grade an alternative their non-vote is translated to the lowest grade for that alternative (Balinski & Laraki, 2007b). The "no-show" paradox is highlighted as an extreme form of abstention. This occurs when one or more voters choose not to vote so their preferred alternative is chosen. Otherwise, their votes would cause another alternative to be selected.
- Violation of Majoritarianism – The candidate whose median is greater than all others' will be selected regardless of the ratings. One example is two candidates (x, y) that each have five votes. The median of x's votes is greater than the median of y's votes thereby declaring x the winner according to Majority Judgment. However, these median votes could have been the preference of one of the five voters whereas the remaining four voters could prefer y over x.

Evaluative Voting is a method where voters score each alternative on a cardinal scale defined by Hillingerto signify their preference for, neutrality toward, or preference against a specific alternative (Hillinger, 2004). Hillinger suggests using integers to rate alternatives to make the scale context independent. Using Hillinger's general election *EV-3* scale (-1,0,1), a voter would give each alternative a score of '-1' (reject the alternative), '0' (neutral stance), or '1' (prefer the alternative). Applying the *EV-3* scale to a general election would be asking the voters to score each of the candidates with respect to the voters' preference for a candidate's election. A voter would vote '-1' indicating they reject the candidate; '0' to indicate their neutral preference; or '1' to indicate they prefer the candidate. Any voter choosing not to vote on a specific candidate (due to a lack of information or indifference) would have a corresponding score of "0" for that candidate. Besides the three-point scale, other sizes of interval scales are defined for Evaluative Voting including EV-5 (-2, -1, 0, 1, 2) and EV-11 (-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5) (Hillinger, 2004).

Hillinger determines that preference aggregation requires a two dimensional measurement scale. The first dimension states whether the scale is ordinal or cardinal. The second dimension is referred to as being context dependent or independent. Hillinger cites that "The paradoxes of social choice arise from the fact that the ordinal scales that are taken as the starting point are also context dependent. On such a scale, the distance between two alternatives is given by the number of intermediate alternatives and change as these are added or subtracted" (Hillinger, 2004). This rationale is what violates Arrow's axiom of independence of irrelevant alternatives. Hillinger claims that independent ordinal scales eliminate paradoxes, yet there is a lack of clarity as to how votes can be aggregated to achieve a decision without unanimity. Thus Hillinger devised Evaluative Voting as a cardinal aggregation method to capture "cardinal preferences" without imposing a uniform, standardized scale. Evaluative Voting aggregates preferences by averaging all of the voters' numerical ratings. Hillinger conducts his work under the auspices of cardinal utility theory. Over time, this has been replaced by ordinal utility theory. Hillinger's approach is considered questionable by some researchers given its reliance upon cardinal utility.

Researchers have explored cardinal voting methods including Evaluative Voting. It has been demonstrated that cardinal voting can satisfy Arrow's Impossibility theorm in that it "can satisfy Pareto efficiency, independence of irrelevant alternatives, unrestricted domain, and… it can be nondictatorship…" (Vasiljev, 2008). One of the greatest challenges of implementing the Evaluative Voting method is the determination of the scale size. Meaningfulness is a critical element to instituting any measurement scale (as discussed in Section 2.3.1). Success has been demonstrated using EV-3, EV-5, EV-11, and EV-100 scales. The specific application motivated the use of these scales.

The necessity and validity of aggregating preferences has been debated for many decades. Social choice theory was developed on the premise that alternative methods may be evaluated under collective decision-making through aggregation (Arrow, 1978; Arrow et al., 2002). The concept of preference aggregation has been re-examined for appropriateness and biases have been identified through preference studies (Morrison, 2002; Sen, 1977). Many methods have been devised using preference aggregation as a foundation. For example, group preference aggregation methods have been incorporated in the Analytical Hierarchy Process (AHP) to determine individual decision-makers' preference weights (Ramanathan and Ganesh, 1994). Another method has been devised to aggregate multiple decision-makers' non-uniform preferences to determine the preferred alternative (Xu, 2007). However, Hazelrigg would argue that "no matter what aggregation procedure we use, we cannot be assured of obtaining a result that is valid with respect to our criteria" (Hazelrigg, p. 247, 2012).

Given the diversity among the personnel who provide test plan input, it is critical to research preference aggregation. There are many methods available to capture individual preferences and a yield a decision. One such category includes methods in the area of Multi-Attribute Decision-Making (MADM). These methods will be discussed in the following section.

2.3.4. Multi-Attribute Decision-Making

Multi-Attribute Decision-Making (MADM) encompasses a group of methods that have been proven beneficial when a selection must be made amongst various alternatives (Fan & Ma, 1999; Ma et al., 2009; Pei-You & Yi-ling, 2009; Yakowitz and Lane, 1993; Whitcomb et al., 1999; Zhang et al., 2009). Whitcomb et al. classify MADM as being a category within Multi-Criteria Decision-Making (MCDM). The other major category within MCDM is Multi-Objective Decision-Making (MODM). MADM is employed for ranking multiple alternatives composed of numerous criteria while MODM is a design process involving vector optimization to achieve a solution. MADM methods are highlighted for their relevance to this research.

MADM is typically based upon the necessity to pick from a list of x alternatives where each alternative has y attributes of differing values or properties. An objective function must first be defined that seeks to maximize benefits or minimize costs of the attributes. According to Hazelrigg, "The purpose of the objective function is to map the outcomes of possible choices onto the real number line in accordance with the preferences of the decision maker." (Hazelrigg, 2012).

Simply put, the objective function serves as a replacement for the decision-maker to allow swift comparisons of many alternatives without having to query the decision-maker for their preferences regarding specific alternative. A weighting factor may also exist allowing different attributes to carry individual levels of importance.

MADM methods examine alternatives after they are fully detailed so that an objective function can be defined to predict their outcomes. The design of test plans incorporates the key decision-makers into the process of selecting specific blueprint elements which is premature for defining an objective function. Without an objective function, MADM would require all possible evaluation blueprints to be input as the list of alternatives where decision-makers would need to specify their preferences for each test plan element within each alternative (blueprint). This would potentially lead to a combinatorial explosion of blueprints. 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 provide their preferences that cover all possible blueprints would be tremendously time-consuming, especially considering that not all Stakeholders will care to test every level of a technology, generate every potential metric, etc. Realizing that methods exist 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.

2.4. Summary

Technology development practitioners have developed, refined, and TRLs as a means to assess a technology's ability to successfully behave under specific conditions or within a specific environment. This area of study has been augmented by defining readiness and maturity separately in the scope of their influence on technology development. These concepts provide valuable information to evaluation stakeholders to aid personnel in devising appropriate test plans given a technology's development state.

Test design methods are presented that have supported the successful generation of test plans to evaluate a variety of complex and robotic technology implementations. Although these methods were consistently used by their respective evaluation designers, they all required a significant level of personal test planning experience to execute. These methods also lack automation forcing test plan iterations to be done manually, further drawing upon prior experience for correctness.

This chapter also lays the foundation for choosing a preference elicitation method. It includes discussion of measurement scales and the requirements of rational consensus that form a sufficient preference method. Several preference capture and handling methods are presented, each with their advantages and disadvantages. Majority Judgment and Evaluative Voting, detailed in 4.6.3, are integrated into MRED and presented alongside several other methods in Table 6. Additionally, Chapter 6 provides a greater exploration of Majority Judgment and Evaluative Voting.

Table 6 - Capabilities of Some Current Preference Handling Methods

SOME CURRENT METHODS										
					MADM					
		Approval Voting	Borda Count	Evaluative Voting	Fan & Ma, 1999	Ma et al., 2009; Pei-you & Yi-ling, 2009	Zhang et al., 2009	Majority Judgment	Pairwise Comparison	Plurality Voting
Preference Capture		✓	✓	✓	✓	✓	✓	✓	✓	✓
Preference Handling for…	Single Stakeholder		✓	✓	✓		✓		✓	✓
	Multiple Stakeholders	✓	✓	✓		✓		✓	✓	✓
Preference Aggregation	Nominal	✓								✓
	Ordinal		✓	✓				✓	✓	
	Interval				✓	✓	✓			
Preference Weighting		N	N	N	N	N	N	N	N	N
Handles Uncertain		N	N	N	Y	Y	N	N	N	N

Chapter 3: MRED Development Strategy

The creation of a technology that undergoes development testing is typically cast as a systems engineering project. Test planning is a essential process integrated through systems engineering processes. Test plan blueprint design is intended to be a foundation for test planning. It is envisioned that MRED's tehnology modeling will derived from systems engineering prinicples and will be integrated with the systems engineering test processes of verification and validation.

This chapter presents MRED's development methodology. The methodology consists of the following elements:

1. Research Technology Development and Test Methodologies – Given systems engineering's wide acceptance as a process to develop advanced technologies, determine where test planning fits into the structure of systems engineering. Leverage experience from implementing the SCORE framework and creating previous test plans to further understand the state of the art and how it can be enriched from a systems engineering perspective.
2. Output Modeling – Develop a model of MRED's output that will identify the key elements of test plan blueprints.
3. Integration with Technology Development – Determine where MRED can be integrated into Systems Engineering.
4. Requirements and Technology Modeling – Devise a model of a technology's physical and functional elements and their relationships based upon information provided during the corresponding technology development process. Identify

performance metrics based upon requirements and define the relationships between these metrics and the physical and functional elements.

5. Define Output Blueprint Element Relationships – Identify the relationships among the various output blueprint elements so they can reasonably constrain the feasible set of output blueprints.
6. Formalization – The process to formalize MRED is outlined in a series of publications written by the dissertation author on this effort. The process includes the execution of the strategies identified in this list's preceding steps along with further steps taken to realize MRED.
7. Preference Handling & Capture Strategy – A strategy is devised sufficiently capture the preferences of the evaluation stakeholders. This strategy also includes a means of using these preferences to determine the most preferred blueprints of the feasible options.
8. Implement – MRED is developed in software to support the verification and validation efforts.
9. Verify & Validate – Check the output blueprint content and MRED-generated output against test plans produced by other methods.

3.1. Systems Engineering

Many organizations have adopted Systems Engineering as their defacto process to shepherd a product through its design and development (Air Force Space Command, 2008; Department of the Navy, 2004; NASA Systems Engineering Handbook, 2007; Office of the Deputy Assistant Secretary of Defense – Systems

Engineering, 2012; SE Handbook Working Group, 2010; US Department of Transportation, 2007). NASA has put forth considerable effort in articulating systems engineering and defines it as…

> "…Systems engineering is the art and science of developing an operable system capable of meeting requirements within often opposed constraints… The systems engineer will usually play the key role in leading the development of the system architecture, defining and allocating requirements, evaluating design tradeoffs, balancing technical risk between systems, defining and assessing interfaces,..Systems engineering is about looking at the 'big picture' and not only ensuring that they get the design right (meet requirements) but that they get the right design" (NASA Systems Engineering Handbook, pp. 3-4, 2007).

System Engineering (SE) is applied to the development and implementation of large and small projects and programs (Air Force Space Command, 2008; Department of the Navy, 2004; NASA Systems Engineering Handhook, 2007; US Department of Transportation, 2007). SE is well-documented and has become the development backbone for many government- sponsored technology development efforts including those in the Air Force, NASA, Navy and the Department of Transportation. The Department of Defense even has an Office of the Deputy Assistant Secretary of Defense, Systems Engineering[5] whose mission is to "Develop and grow the Systems Engineering capability of the Department of Defense – through engineering policy, continuous engagement with Component Systems Engineering organizations,…"

[5] http://www.acq.osd.mil/se/

SE's objective is provide a process to ensure that a system is designed, produced, and operated so that it accomplishes its purpose in the most cost-effective way possible considering performance, cost, schedule, and risk (NASA Systems Engineering Handbook, 2007). To assure effective technology design and implementation, SE places a significant emphasis on test planning and execution.

The SE design process in the US Government agencies cited in the introduction is characterized by the "V" Model (Figure 4). The "V" Model is used to illustrate the SE activities during the life cycle of a product (ModelBased Systems Engineering Initiative, 2008; US Department of Transportation, 2007).

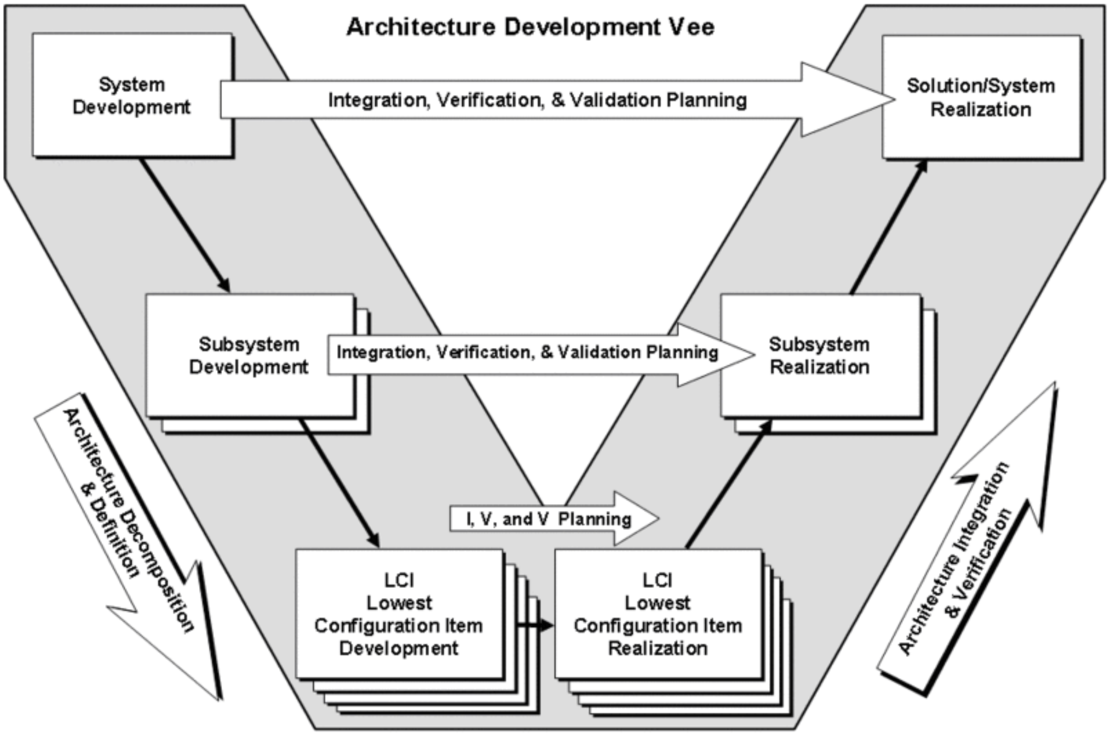

Architecture Development Vee
System Development
Integration, Verification, & Validation Planning
Solution/System Realization
Subsystem Development
Integration, Verification, & Validation Planning
Subsystem Realization
I, V, and V Planning
LCI Lowest Configuration Item Development
LCI Lowest Configuration Item Realization
Architecture Decomposition & Definition
Architecture Integration & Verification

Figure 4: Architecture Development Vee Model (Forsberg et al., 2005; SE Handbook Working Group, 2010)

The "V" Model is composed of three sets of common technical processes: system design, product realization, and technical management. The goal of the system design processes is to define and baseline stakeholder expectations, generate and baseline technical requirements, and convert technical requirements into a design solution (NASA Systems Engineering Handbook, 2007). The product realization process begins at the bottom of the “V.” The goal of the product realization processes is to create a design solution for each subsystem within the product (NASA Systems Engineering Handbook, 2007). The technical management processes are employed to manage the communication across the subsystem interfaces, assess the project’s progress and control the technical execution of the project to it’s conclusion (NASA Systems Engineering Handbook, 2007).

The top-down systems design processes feature the capture of detailed expectations of technology performance which are converted into requirements. These requirements are then turned into specifications. The technology’s requirements are decomposed using models and diagrams to show the relationships among elements (e.g., requirements, subsystems and components). The decomposition of the requirements drives the design of one or more feasible solutions. As the requirements are decomposed, abstract specifications can be identified for each subsystem and component level without detailing specific design solutions. Knowing the potential subsystems enables the technology designers to decompose the subsystems with diagrams, requirements, and concepts of operations to produce feasible design solutions (NASA Systems Engineering Handbook, 2007).

3.1.1. Testing Forms in Systems Engineering

Test and evaluation is a significant segment of systems engineering. Requirements are validated against stakeholder expectations going down the left side of the “V” model during the requirements decomposition process. The “V” Model in Figure 4 presents the need to initiate the identification of verification and validation planning during the systems design processes (SE Handbook Working Group, 2010). Verification planning includes the definition of an “Initial Requirements Verification and Traceability Matrix (RVTM)” (SE Handbook Working Group, 2010). This matrix maps the list of requirements to specific verification attributes (synonomous with metrics). Validation planning includes the specification of personnel who will perform the validation exercises along with the environments the technology should be tested. These planning activities also include the definition of minimum and ideal system performance characteristics. These thresholds and goals have a significant impact in motivating these plans. The verification and validation activities noted are synonomous with test planning.

Verification and validation plans are iterated upon throughout the requirements definition process (left side of the “V”) and executed at various intervals during the product realization process (right side of the “V”). The earliest validation performed in SE is ensuring that the defined requirements align with the stakeholder expectations. If this alignment is unsuccessful, it is doubtful that the right technology will be produced.

Further tests come in the form of technology assessments, verification and validation throughout the “bottom-up” product realization processes. These three forms of testing focus on the evaluation of the physical elements of the technology (NASA Systems Engineering Handbook, 2007):

- *Technology assessment* – Facilitates the interaction between the technology development and design processes to confirm that the design mirrors the realitites of the available technology. Technology assessments are done until requirements and available resources are aligned with the program stakeholders' wishes.
- *Verification* – Shows proof of compliance with requirements. Specifically, verification indicates that the technology can meet each requirement as proven through performance of a test, analysis, inspection or demonstration. Verification testing relates back to the requirements set and must be performed at different stages in the product life cycle (NASA Systems Engineering Handbook, 2007; SE Handbook Working Group, 2010).
- *Validation* – Demonstrates that the technology accomplishes the intended purpose in the intended environment. Sucessful validation demonstrates that the technology meets the expectations of the stakeholders as shown through performance of a test, analysis, inspection or demonstration. The intent of validation is to determine the effectiveness and suitability for use in mission operations by typical users (NASA Systems Engineering Handbook, 2007; SE Handbook Working Group, 2010).

The selection of methods for verification and validation is based upon engineering judgement on the most effective way to demonstrate the technology's conformance to requirements (NASA Systems Engineering Handbook, 2007). In the systems engineering domain, verification proves whether the technology was created properly while validation proves whether the proper technology was created. "End-to-End System Testing" is among one of the many test events that SE promotes

(NASA Systems Engineering Handbook, p. 93, 2007). The goal of end-to-end testing is to present the interface compatibility and required functionality among the various elements of a system, between multiple systems and within the entire system. End-to-end testing typically showcases the entire system satisfying its mission requirements and goals under operational scenarios.

3.1.2. Technical Measures in Systems Engineering

Systems engineering states that technical measures are an output of the technical requirements definition process. SE defines technical measures as a "set of measures based on the expectations and requirements that will be tracked and assessed to determine overall system or product effectiveness and customer satisfaction" (NASA Systems Engineering Handbook, page 41, 2007). The common SE measure terms include Measures of Effectiveness (MOEs), Measures of Performance (MOPs), and Technical Performance Measures (TPMs).

3.1.2.1. Measures of Effectiveness (MOEs)

Measures of Effectiveness are success measures that relate to the attainment of mission goals within the targeted operational environment (NASA Systems Engineering Handbook, 2007). MOEs are aimed at demonstrating how well the mission goals are achieved as opposed to how they are accomplished making them solution independent. "Time to mission completion" is an example of a MOE if the technology under test is a man-portable urban search and rescue robot. This enables MOEs to be applicable across multiple technologies as long as the technologies are intent on accomplishing the same mission goals. System engineering uses MOEs to perform the following (NASA Systems Engineering Handbook, 2007):

- Identify high-level operational requirements from the stakeholder's perspective
- Explore the relationships between technology parameters and mission success
- Ensure that the quantitative mission goals are viable as technology development progresses

MOEs are established during processes modeled on the left side of the "V."

3.1.2.2. Measures of Performance (MOPs)

SE defines MOPs as the measures that describe physical or functional characteristics relating to the technology (NASA Systems Engineering Handbook, 2007). "Maximum Torque" and "Lift Capacity" are two examples of MOPs relevant to evaluating an urban search and rescue robot with a manipulator. MOPs are usually captured under very specific test scenarios or relevant environments. One or more MOPs contribute to a MOE, yet MOPs are not measured directly in mission effectiveness. Rather MOPs typically turn into technology performance requirements. When these MOPs are met by a design solution the critical threshold for system MOEs is usually obtained (NASA Systems Engineering Handbook, 2007).

The significant difference between MOEs and MOPs is that MOEs are formulated from mission success criteria while MOPs are produced from actual system performance criteria with respect to a specific technology. Mission success criteria are often expressed directly from the statekholder's point of view while system performance criteria is indirectly related to the stakeholder's perspective. Often system performance criteria is established amongst the technology developer(s), evaluation designers and any other parties involved in the evaluation strategy.

3.1.2.3. Technical Performance Measures (TPMs)

All TPMs are ultimately derived from MOEs or MOPs. TPMs are those critical mission success or performance parameters that are observable through testing and/or analysis of the technology and its constituent subsystems (NASA Systems Engineering Handbook, 2007). The intent is to compare the observed TPM values, with those that are expected during test events. TPMs either verify that a technology is making the necessary developmental progress or that it is falling short at these measurable milestones. If the technology is not meeting expectations, then TPM data can determine where the shortcomings are and provide the technology developers with further information to make the required improvements. Systems engineering utilizes TPMs for several reasons including:

- Estimate values to be attained by key parameters at crucial activities during the implementation process.
- Highlight differences between measured and expected parameter values
- Identify estimated values for those parameters to evaluate the implications on system effectiveness
- Enhance the assessments of proposed design changes

TPMs can be generic where they are applicable to all subsystems (e.g. mass, reliability, etc.) or can be specific to one or more subsystems. Measures must meet three specific criteria in order for them to be considered useful TPMs. The criteria are:

- Be an important descriptor of the technology that can be scrutinized at specific milestones

- Be measureable
- Support the establishment of planned progress profiles (e.g. from prior data or based on evaluation design activities)

The relationships among the MOEs, MOPs, and TPMs are visually presented in Figure 5.

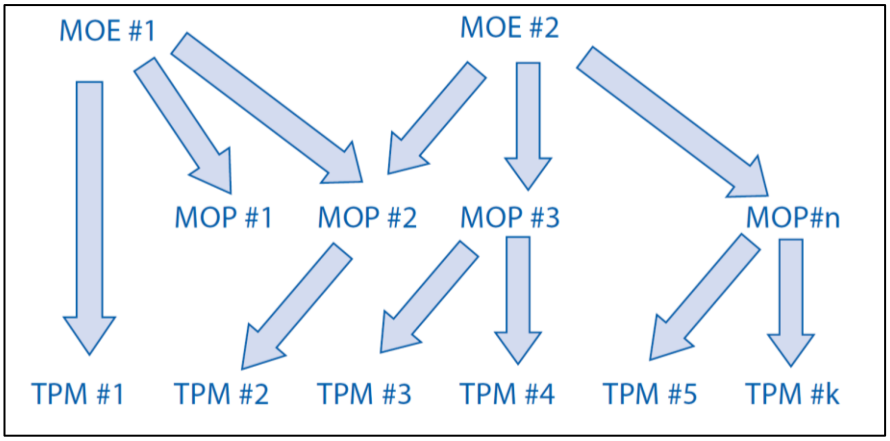


Figure 5: Relationships among MOEs, MOPs, and TPMs (NASA Systems Engineering Handbook, p. 192, 2007)

3.1.3. Integration of Test and Evaluation

Given the importance SE places upon verification and validation, it's apparent that technology assessments are critical to the Systems Engineering process. Both the International Council on Systems Engineering (INCOSE) and the government organizations listed in Section 3.1 present verification and validation in their SE handbooks. Given the range of organizations that employ Systems Engineering, it can be stated that technology assessments are universal to nearly all technology design. The importance of assessments is noted in that: 1) stakeholders rely on results of assessments to make key decisions throughout the technology's life-cycle, 2) assessments are not restricted to the end of the development cycle; they are iterative, and 3) assessments are conducted throughout the course of a technology's development beginning with focused tests of basic components to complex tests of the complete system.

The "V" Model in Section 3.1 presents SE as a process beginning with a top-down approach, where the technology requirements definitions flow top-down, followed by a product realization process flowing in a bottom-up approach. As the technology requirements definition process begins, so to does test planning. INCOSE includes the following outputs from the stakeholders when specifying their requirements (SE Handbook Working Group, 2010):

- *MOE Needs* – The MOEs that represent the "operational" measures of success that are closely connected to the accomplishment of the mission goal(s). The MOEs take into account the targeted operational environment.
- *MOE Data* – Data employed to measure the MOEs.
- *Initial Requirements Verification and Traceability Matrix* (RVTM) – This matrix features a list of requirements, their corresponding verification attributes, and their traces.

Once the MOEs are defined at the complete system level, the requirements definition process continues down successive levels, from one or more subsystem levels down to the component level. MOPs and/or TPMs are derived at these constituent element levels providing the SE team with pertinent measures. As these three types of measures are defined (MOEs, MOPs, and TPMs), test planning can occur. Verification begins once the bottom of the "V" is reached and the product realization process starts. The lowest built levels of the technology can be tested to capture the pertinent MOPs and TPMs. Likewise, at the highest levels of the "V," validation testing is conducted to obtain the MOEs.

Test planning is an iterative process as a technology continues to grow and evolve. A verification (or validation) plan may be altered prior to a test event if the technology's development has changed from its original plan. Technology development and testing are symbiotic; test feedback (output TPMs, MOPs, and MOEs) during the verification and validation process impacts future iterations of the technology (and its sub-levels of technology); and updated technology developments impact test plans.

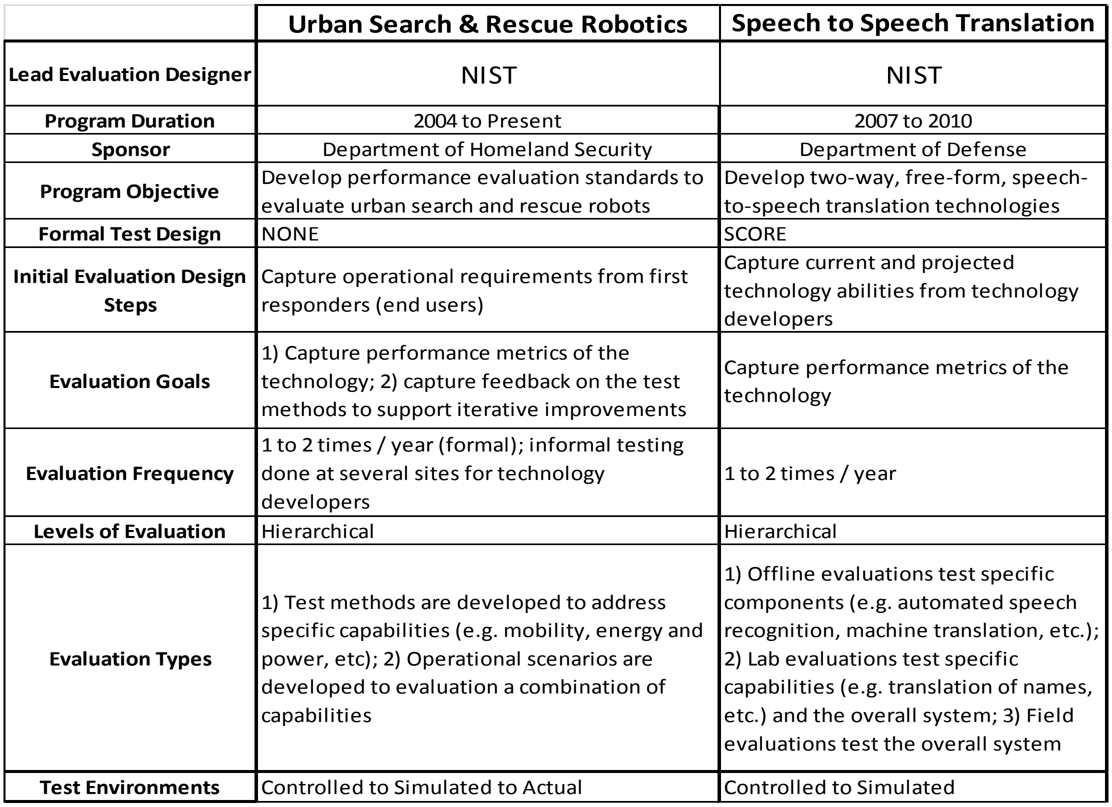
3.2. Existing Technology Test Processes

Many advanced technologies have been evaluated using methods that did not consider the Systems Engineering (SE) perspective. This is evident in several significant evaluation efforts led by researchers at NIST (Jacoff et al., 2003; Schlenoff et al., 2006; Weiss et al., 2007; Weiss et al., 2008; Weiss & Schlenoff, 2008). Program sponsors initiated these evaluation efforts by providing NIST personnel with key information of the technologies to be tested. This information included stakeholder requirements, MOEs, and MOPs (although this terminology was not used). Adhoc methods derived from experience were used to turn these requirements and measures into feasible test plans. Recently, researchers have documented and organized commonly used Adhoc methods to both add structure and facilitate their usage within the test and evaluation community. The SCORE methodology discussed in Section 1.4 is one such example. Evaluation personnel worked with other stakeholders to define additional metrics beyond the initial MOEs and MOPs. These newer metrics could would be classified as TPMs and/or additional MOPs in Systems Engineering parlance.

Evaluation personnel primarily operated on the product realization (right side) of the “V” model. Evaluation personnel were provided the key information, they were not involved in the technology requirements definition process but were focused only on test planning and execution. Seldom were these test planning and execution exercises referred to as verification and/or validation by the stakeholders. Rather, test events of advanced technologies were viewed as milestones to assess the current state of a technology’s development. The over-arching goal of these test events was 1) to sufficiently inform the program manager of the technology’s performance prior to key decision points, 2) inform the technology developers as to the status of their technologies, and 3) capture feedback from the expected user community. These test events are more in line with verification exercises, as opposed to validation events, since NIST personnel evaluated multiple facets of a technology at levels lower than the complete system and rarely tested the fully-developed technology in the intended operational environments.

Table 7 presents a comparison between two evaluation programs led by NIST; one aimed at developing performance evaluation standards of Urban Search and Rescue (US&R) robots and the other aimed at evaluating Speech-to-Speech (S2S) translation technologies.

Table 7 - Test Plan Design Comparison: US&R robots & S2S translation systems



	Urban Search & Rescue Robotics	Speech to Speech Translation
Lead Evaluation Designer	NIST	NIST
Program Duration	2004 to Present	2007 to 2010
Sponsor	Department of Homeland Security	Department of Defense
Program Objective	Develop performance evaluation standards to evaluate urban search and rescue robots	Develop two-way, free-form, speech-to-speech translation technologies
Formal Test Design	NONE	SCORE
Initial Evaluation Design Steps	Capture operational requirements from first responders (end users)	Capture current and projected technology abilities from technology developers
Evaluation Goals	1) Capture performance metrics of the technology; 2) capture feedback on the test methods to support iterative improvements	Capture performance metrics of the technology
Evaluation Frequency	1 to 2 times / year (formal); informal testing done at several sites for technology developers	1 to 2 times / year
Levels of Evaluation	Hierarchical	Hierarchical
Evaluation Types	1) Test methods are developed to address specific capabilities (e.g. mobility, energy and power, etc); 2) Operational scenarios are developed to evaluation a combination of capabilities	1) Offline evaluations test specific components (e.g. automated speech recognition, machine translation, etc.); 2) Lab evaluations test specific capabilities (e.g. translation of names, etc.) and the overall system; 3) Field evaluations test the overall system
Test Environments	Controlled to Simulated to Actual	Controlled to Simulated

There are several notable differences between the two programs and their respective evaluations. They are:

- The objective of the US&R program is performance evaluation development; the objective of the S2S program is technology development.
- No formal methods were used to develop performance evaluations in the US&R program; the SCORE method was applied to develop the S2S evaluations.
- Evaluation goals were used to assess specific technologies and solicit feedback on test methods for the US&R program; the sole evaluation goal of the S2S tests was to assess the technology.

Prior to and during the implementation of the SCORE framework (discussed in 1.4), stakeholder motivations were solicited. The US&R and S2S programs (noted in Table 7) aimed to capture some stakeholder preferences almost immediately in the evaluation design process. Devising test plans for US&R robots began with direct interaction with first responders (the targeted user population of these robots) followed by discussions with the technology developers. Similarly, creating evaluation plans to test S2S systems began with conversations with the technology developers in parallel with meetings with the program manager. Two questions about these information-gathering sessions were 1) At what level should detailed information be captured? and, 2) To what extent? Sometimes, stakeholders provided very detailed preferences that could not be accommodated in the test plans due to a lack of resources or an immaturity of a technology. Some objectives of evaluation designers and user conversations were: 1) to understand the environment in which the technology would be deployed; 2) to understand the need the technology would fulfill if successful; and, 3) to determine the most important characteristics of the technology from the user perspective (e.g. accuracy, speed, etc.).

The objectives of the interactions with the technology developers were to: 1) understand the existing abilities and limitations of the technology, 2) determine the expected state of development of the technology at the time of the evaluation, 3) determine what type(s) of testing their technology had previously undergone and how the technology performed, and 4) capture their preferences regarding evaluation characteristics. Consultation with the technology developers was usually an ongoing process. The constant communication was critical in case technology development sped up or slowed requiring modifications to the test plans.

Evaluation designers often have to revise their test plans before their execution. Often time-consuming, the task of revising test plans is usually prompted by either the program manager adjusting the program's focus or the technology developer updating the state of their technology at a rate inconsistent with previously captured information. The evaluation design process between different programs (as evidenced by the US&R and S2S programs shown in Table 7) was seldom consistent and largely relied upon the experience of the evaluation design team. The process was based on personal heuristics that were not articulated or encoded into a method. A common challenge when a technology is still under development or has never been tested is determining exactly how to evaluate it. Evaluation practitioners not only rely on prior, comparable evaluation experience, they also rely on the technology developers and the users for key information. Given the complexity and potential changes to an advanced, developmental technology, pertinent information becomes outdated at a quicker pace as compared to information associated with a fully-developed or previously-tested technology.

These issues noted above are largely attributed to the lack of a structured test plan design method and are without the context of the formal Systems Engineering process. A structured approach that output test plan blueprints would have streamlined the evaluation design process for both the US&R and S2S efforts. Constructing blueprints would provide evaluation stakeholders with the following benefits:

- Capture the evaluation stakeholders' preference in a structured manner
- Ensure critical evaluation resources are identified

- Identify the state of development of the technology and its constituent elements
- Chronicle the key evaluation elements over multiple evaluations to aid performance improvements over time
- Chronicle stakeholder preferences as they evolve, both prior to a specific test event and between multiple test events, and understand their impact on output blueprints
- Shorten the time it takes to identify current and applicable blueprints given changing preferences, resources, and technology state.

3.3. Model of MRED's Output – Test Plan Blueprint Elements

Creating the overall test plan blueprint generator model began with determining the appropriate outputs. When an evaluation designer is tasked with creating test plans, it is practical for them to first ask "What do I want to accomplish? What are the goals of this evaluation? and, What metrics do I want to capture? How should the technology under test be used during the evaluation?" Systems Engineering disciples will ask the same questions, just in SE terms. When tasked with devising a test plan, SE-practicing evaluation designers would ask "What technology and/or its constituent elements are the focus of the evaluation?; What are the technology requirements of the elements to be evaluated?; What MOE, MOP, and TPM data is required?; and What is the concept of operations?" (NASA Systems Engineering Handbook, 2007; SE Handbook Working Group, 2010).

MRED's goal is to identify a set of feasible and desirable blueprints. The feasibility of the blueprints is based upon the development state of the technology and the availability of test resources. The desirability of the blueprints is based upon the stakeholder preferences. The model of MRED's output is driven by the blueprint's desired outputs noted above. Likewise, this output is influenced by the outputs documented from prior test plans, such such as the test plans created to assess US&R and S2S technologies. Specifically, MRED seeks to output the following:

- Physical and/or functional element(s) of the technology to be tested. Systems Engineering requires verification of these elements during the course of a technology's development. This includes one or more the following physical SE elements: system, subsystems, and components. The technology's capabilities are described by different functional elements.
- Metric(s) to be captured from the assessment and data analysis. MOEs, MOPs, and/or TPMs represent the metrics to be acquired from the physical and/or functional element(s) being tested.
- Personnel required to directly and indirectly interact with the technology during the test.
- Environments in which the technology and its interactions with specified personnel will be tested. Test environments range from laboratory settings capable of isolating specific physical or functional elements of a technology to operational environments specified during the SE requirements definition process.
- Levels of technical and operational understanding required from personnel.
- Levels of decision-making to appropriately empower the personnel during their interactions with the element(s) under test and with other personnel.
- Evaluation scenarios dictating personnel actions within the environment, with other personnel, and the technology.

- Complexity required within the environment to appropriately evaluate the element(s).
- Equipment necessary to make the observations and/or collect the data to support the desired metrics

Although each of the blueprint elements are noted separately above, relationships exist among many of these elements. These relationships and any dependencies are defined in Section 4.4. The MRED blueprint is verified by taking test plans generated by other methods and modeling their output according to MRED'S defined output (see Section 4.7).

3.4. Integration of MRED with Systems Engineering

MRED is designed to fit into the context of Systems Engineering. MRED's activities fit into the "V" Model of Systems Engineering activities. These activities begin as soon as the Systems Engineering product realization process starts and include using information regarding pertinent metrics, the state of the technology and preferences of the stakeholders. Figure 6 highlights MRED's integration into the Systems Engineering "V" Model. Specifically, MRED can generate initial test plan blueprints at the onset of each level of the product realization process (right side of the "V" Model). As the product realization process matures and the technology is further developed and refined, MRED facilitates the refinement of test plan blueprints as updates are made to the technology's design.

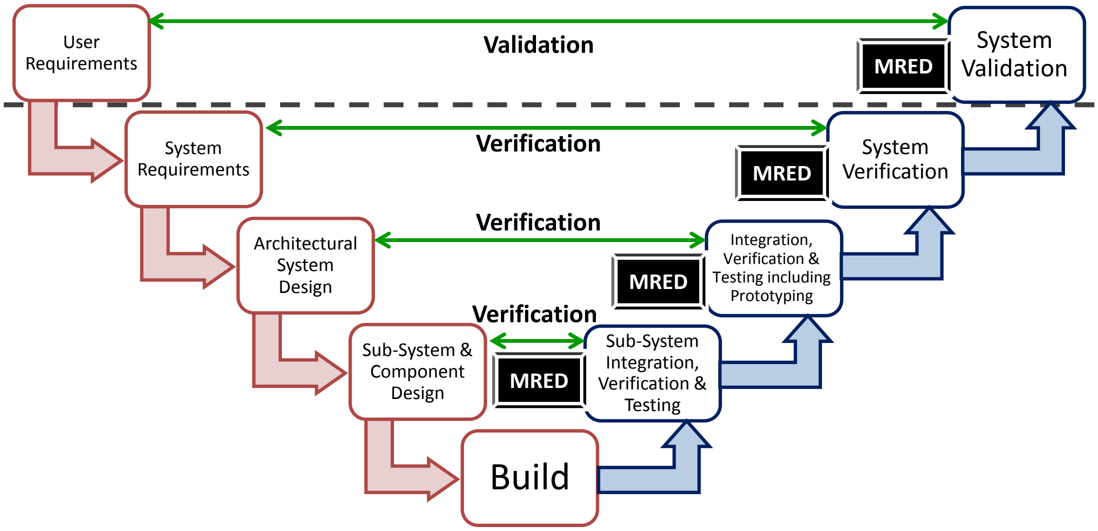

User Requirements
Validation
MRED
System Validation
System Requirements
Verification
MRED
System Verification
Architectural System Design
Verification
MRED
Integration, Verification & Testing including Prototyping
Verification
Sub-System & Component Design
MRED
Sub-System Integration, Verification & Testing
Build

Figure 6: MRED's Integration into Systems Engineering (Adapted from figure in Tetlay and John, 2009)

MRED also leverages the structure and hierarchy employed in Systems Engineering to model the technology to support test plan blueprint generation. This hierarchy is used to further identify relationships among the physical and functional elements of a technology; these relationships are elaborated upon in the next section.

3.5. *Requirements and Technology Modeling Strategy*

A strategy was developed to model the technology and its requirements for MRED. The purpose of this strategy is to define several key relationships: the relationship between a technology's requirements and pertinent metrics; the relationship between a technology's physical elements and functions; and the relationship between the metrics, physical elements, and functions.

3.5.1. Requirements Modeling

Systems Engineering provides evaluation designers the opportunity to use the technology requirements that were previously defined. The requirements definition process produces a requirements tree that details the MOEs, MOPs, and TPMs (see Section 3.1.2). An abstract requirements tree is shown in Figure 7.

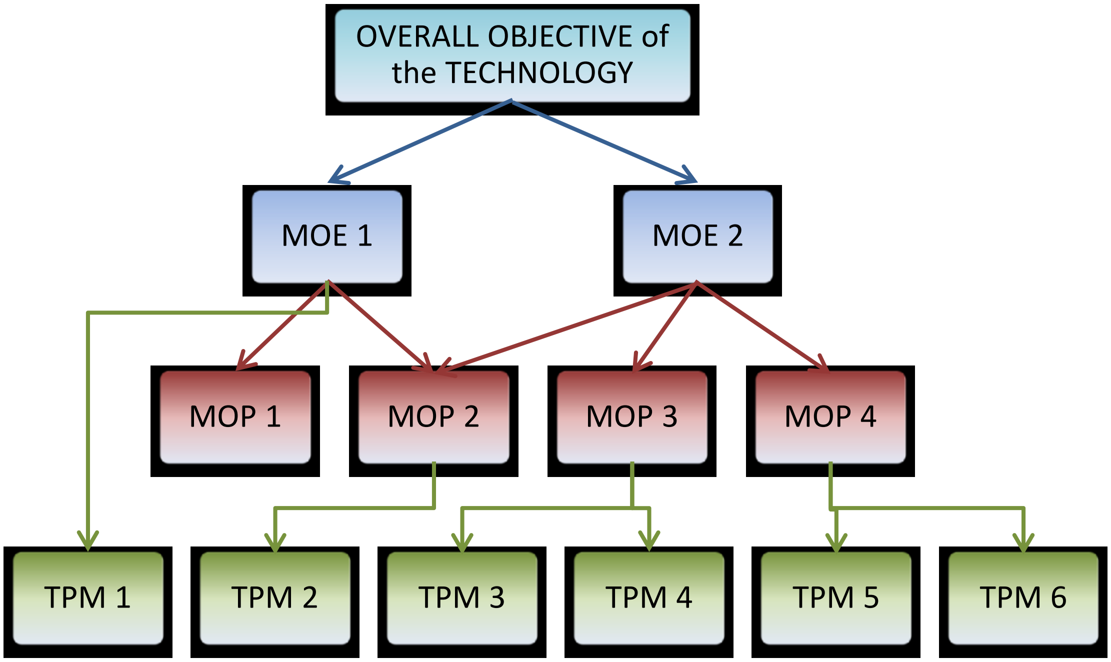


Figure 7: Example Requirements Tree Presenting the Relationships between MOEs, MOPS, and TPMs

The requirements definition process begins with defining the Measures of Effectiveness (MOEs) applicable to the technology's ultimate use in its intended operational environment by the targeted users. The process continues by extracting Measures of Performance (MOPs) from the MOEs. As discussed in Section 3.1.2.2, MOPs are quantitative and/or qualitative measures describing the performance of physical or function elements of the technology in controlled or relevant environments. Technical Performance Measures (TPMs) may be extracted from either MOEs or MOPs and are quantifiable measures representing a constituent element's performance at key intervals of the product realization process. Defining these requirements enables the product realization process to build the technology up from the lowest component level.

3.5.2. Technology Modeling

A technology is physically comprised of systems, sub-systems, and components. Figure 8 presents an abstract physical decomposition of a technology.

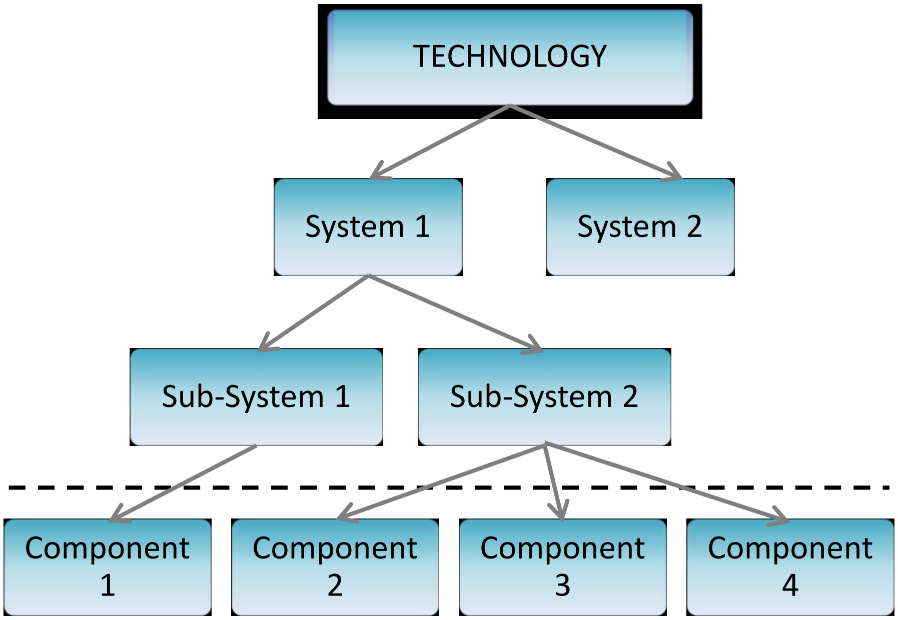


Figure 8: Physical Model of a Technology

The broken line between the Sub-System and Component levels in Figure 8 represents where additional levels may be necessary to represent the specific decomposition of the technology. Each of the physical elements play a role in enabling a specific function of the technology. The modeling strategy is to simplify the physical and functional relationships of a technology to support test plan blueprint generation.

MRED's technology model is defined by three key elements: component, capability, and system. They are as follows:

- Component - Essential physical part or feature of a System that contributes to the System's ability to accomplish a goal(s).

- Capability - A specific ability or function of a technology. A Capability is enabled by either a single Component or multiple Components working together.
- System - A group of cooperative or interdependent Components forming an integrated whole to accomplish a specific goal(s). As used here, the System is synonomous with the whole technology.

These terms are elaborated upon in Section 4.3.1. The following section presents the relationships between these three terms.

3.5.3. MRED Technology Relationships

Prior test plan development experience has shown that relationships between test plan inputs exist that impact the feasibility and quality of the test plans. Some relationships are technology dependent (i.e. must be explicitly defined by the technology developer). Figure 9 presents an abstract relationship diagram that illustrates technology dependence (i.e. the physical elements required for the operational of functional elements). All of the components work together to make up the physical system. Likewise, the sum of capabilities yields the overall functions of the system to accomplish its intended mission.

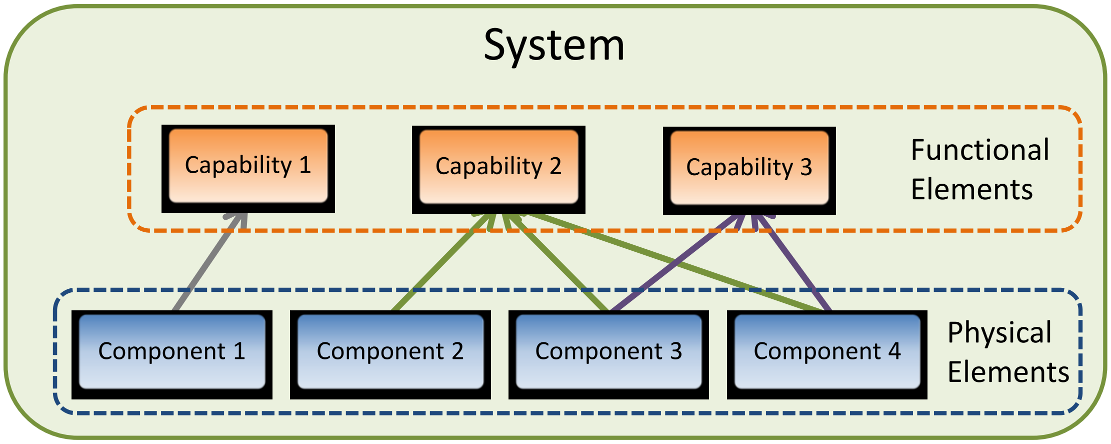
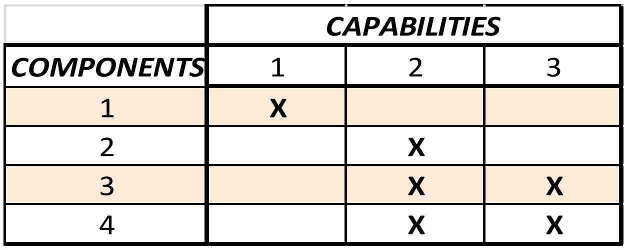


Figure 9: Relationships Between the Components and Capabilities of a Technology

The simplest causal relationship between components and capabilities is that shown between Component 1 and Capability 1 in Figure 9. Component 1 is solely responsible for the Capability 1; likewise Capability 1 is enabled only by Component 1. In intelligent systems, this simple relationship is often the exception, not the norm. This simple causal relationship links the capability's maturity to the component's maturity. If the component is immature, so is the capability. If the component is fully mature, then the capability is also fully mature.

Table 8 presents a matrix representation of relationships presented in Figure 9. Table 8 shows that Component 1 only fulfills a single capability since a single "X" is present in Component 1's corresponding row.

Table 8 - Relationship Matrix between Components and Capabilities corresponding to Figure 9



	CAPABILITIES		
COMPONENTS	1	2	3
1	**X**		
2		**X**	
3		**X**	**X**
4		**X**	**X**

The relationships between components and capabilities are often complex. Typically, a component will have causal relationships with multiple capabilities. Component 3 (in Figure 9) is an example of this type of relationship. According to Figure 9 and Table 8, Component 3 supports both Capability 2 and Capability 3. It's possible that Component 3 could equally support Capability 2 and Capability 3 or Component 3 could play a stronger role in one capability. Likewise, multiple components may have casual relationships with a single capability. Capability 2 is supported by Component 2, Component 3 and Component 4. Similar questions can be raised about the level of support one component provides a capability over another component.

MRED uses these relationships to aid in establishing the technology state of components, capabilities, and the complete system. These relationships, coupled with the technology developer's knowledge of the maturity of the physical components, allow MRED to quickly identify those capabilities that are not fully-mature. The use of this strategy is detailed in Section 4.3.3. MRED does not consider the reliability of a technology's components, capabilities, or the complete system when determining the technology state. This is because reliability information is not available for these elements in the use cases since this research effort is focused on developmental technologies. In the context of MRED, Reliability would be defined as the probability that a specific component, capability, or the system will continue to function under certain conditions for a specific period of time.

An alternative one-to-one approach to further decompose the physical and functional elements shown in Figure 9 is also explored and documented in Appendix A: Technology Decomposition. The one-to-one approach symbolically decouples the components and capabilities to produce a one-to-one mapping of the relationships. This approach is not integrated into MRED because symbolic-decoupling does not yield any additional value to MRED when binary relationships are documented between components and capabilities. Symbolic-decoupling could prove valuable in future research if the *Maturity* of a component can be described in greater detail than 0 or 1. Likewise, further work can explore deeper relationship classification to document which *Components* play a stronger role than others in the function of specific *Capabilities.* Overall, future work could couple this research with other approaches in identifying component and capability technology states to produce a more beneficial approach.

3.5.4. MRED Metrics and Technology Relationships

MRED uses two types of metrics; each of which can be mapped to SE measures of MOEs, MOPs, and TPMs. MRED's classes of metrics are:

- *Technical Performance* – Metrics that quantify behavior (e.g. accuracy, distance, time, etc.). These metrics may be required by the program sponsor to meet user expectations, inform the technology developers on their design, etc.
- *Utility Assessments* – Metrics expressing qualitative factors that express the condition or status of being useful and usable to the target user population.

Figure 10 presents the mapping between MRED's metric classes and SE's measures.

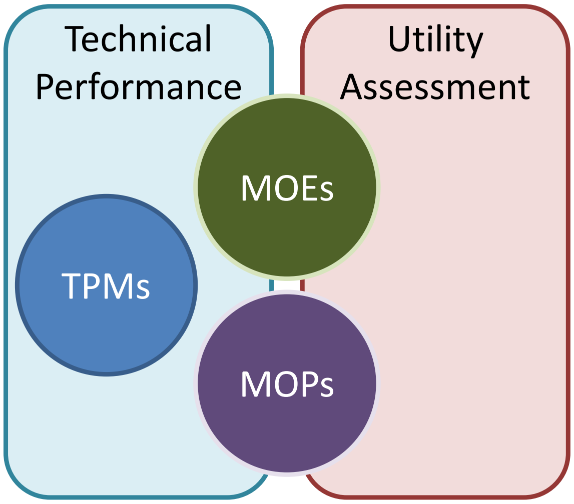


Figure 10: Relationship between MRED Metrics and SE Measures

Note that both MOEs and MOPs can be characterized as *Technical Performance* and *Utility Assessment* metrics. TPMs are mapped soley to *Technical Performance* metrics. MRED relationships between a technology's physical elements, functional elements and its measures are consistent with Systems Engineering principles. Figure 11 shows which metrics can be obtained from testing which of the physical and/or functional elements. *Technical Performance* metrics can be captured from evaluations of the components, capabilities or the complete system. *Utility Assessment* metrics can be captured from the capabilities and the complete system.

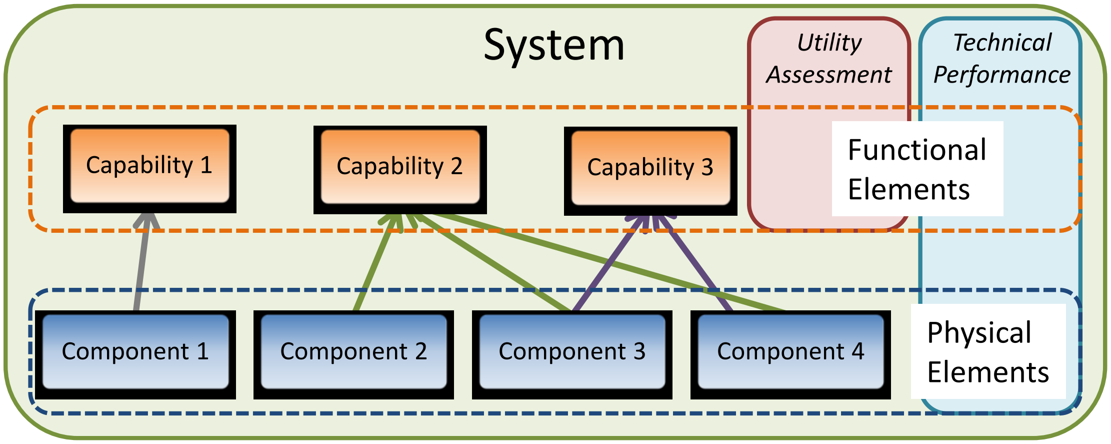


Figure 11: Relationships between MRED Technology Elements and Metrics

MRED uses the relationships among the physical elements, functional elements, and the metric types in the process of generating test plan blueprints. The benefit of these relationships is they practically constrain which metrics can be captured from the various technology elements and they allow the stakeholders to observe the coverage of metrics across the entire system and its individual elements. Enabling the stakeholders to track which metrics are applied to which elements offers a holistic view of a technology's evaluation including metric gaps.

3.6. <u>Output Blueprint Element Relationships</u>

Many relationships exist among the output blueprint elements defined in Section 3.3. These relationships impose constraints upon the blueprint elements. Figure 12 presents a constraint graph of the output blueprint elements. Each node represents a blueprint element while each link represents the presence of a relationship (leading to a constraint). These will be discussed in detail in Section 4.4.

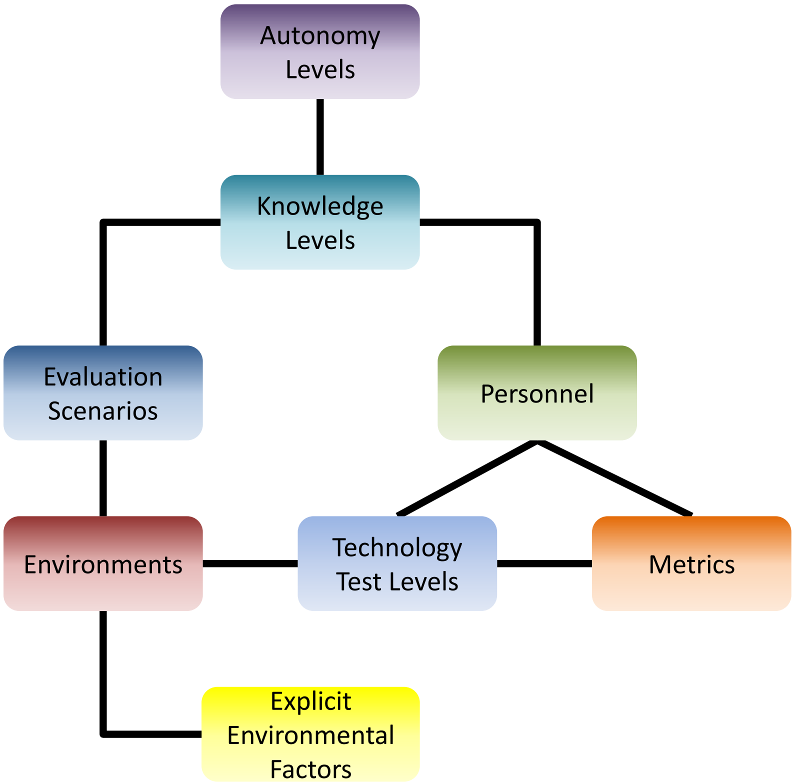


Figure 12: Constraint Graph of Blueprint Elements

The relationships between the different technology test levels (i.e. components, capabilities, and the system) and the types of test environments is an example of a relationship between two output blueprint elements, also shown in Figure 12. Three types of test environments are defined within MRED as test plan blueprint outputs. Detailed in Section 4 and 4.4.5, the three environments are simply known as the lab (highly-controlled and structured), the simulated (less controlled than the lab, yet not the true operational environment), and the actual (operational environment) environments. Figure 13 presents the relationships between a technology's elements and the types of environments that can support testing.

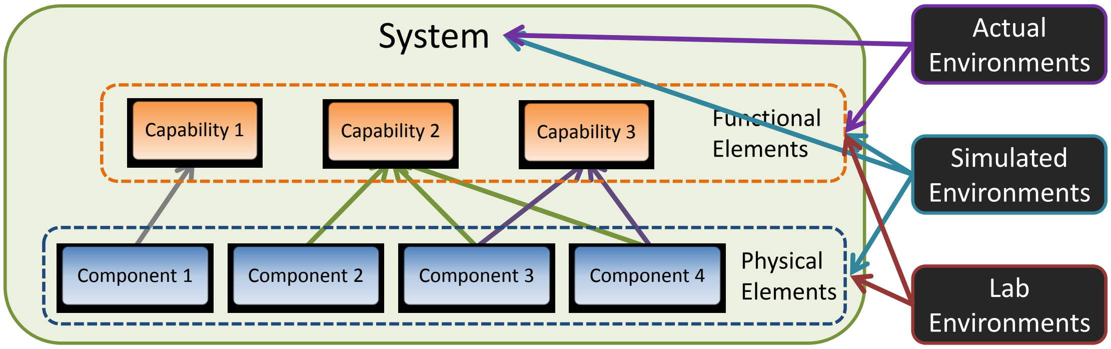


Figure 13: Relationships Between the Technology Elements and the Environments

Given prior evaluation design experience, the lab environment (highly-controlled and structured) is suitable to test components and capabilities. Likewise, the simulated environment (less controlled than the lab, yet not the true use-case environment) can test all three technology test levels. Finally, the actual environment (intended use-case environment) can test a technology's capabilities and the system. This is just an overview of a few of the many relationships that MRED handles (further details are defined in Chapter 4).

3.7. *Preference Capture & Handling Strategy*

Stakeholder preferences are time intensive to capture and have a short shelf life. Given the expense of capturing stakeholder preferences, they should ideally be captured only once during the evaluation design process and done so in an efficient manner. It is critical to capture stakeholder preferences at a time after the available technology state and resources have been identified, but before the final design of the test plan(s). It is important to acknowledge that not all stakeholders care about all test elements, so stakeholders should have the option to refrain from providing preferences on any evaluation elements without impacting the corresponding element selection.

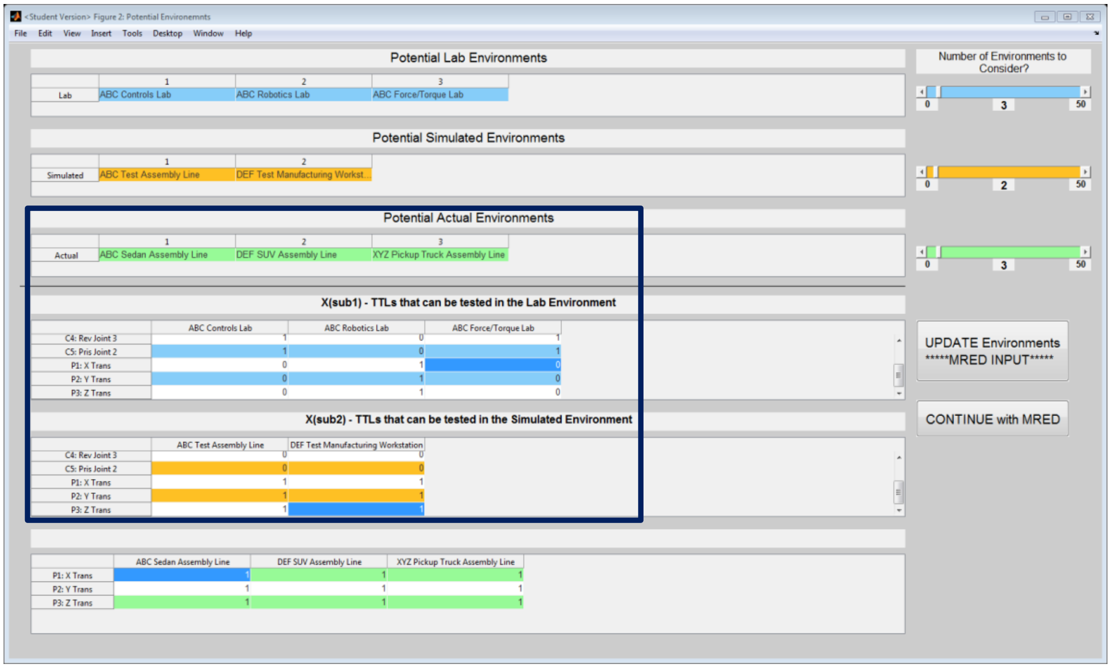
Each stakeholder wants the technology to perform well, yet they may have varying perspectives as to what "perform well" means. Some stakeholders may only care about individual *Capabilities* and overall *System* performance. However, other stakeholders may care about detailed performance at all levels of the technology. *Stakeholder Preferences* should be provided without bias.

MRED uses preference capture and handling cycle to identify blueprint elements: 1) stakeholders provide their preferences for a single type of evaluation element, 2) the preferences are recorded and processed leading to a rating of elements (detailed in Section 4.6.3), and 3) the least preferred elements are eliminated from further consideration. This strategy minimizes the burden on the stakeholders by only asking for their preferences with respect to feasible alternatives (as opposed to asking for their preferences on all evaluation elements where some could be rendered irrelevant given various relationships).

3.8. Implementation

Both the process and preference strategies were implemented in software to aid in the development of MRED. Matlab was chosen to offer the evaluation designer an iterative and interactive process by which to input information and data. The use of Matlab supported the development of MRED's interactive features by enabling interface design. There are numerous advantages to using Matlab. They are that 1) it has built-in graphical user interface tools, 2) supports the required linear algebraic equations to process MRED's defined vectors and matrices and 3) provides a coding environment with loops, arrays, etc. to effectively capture and manipulate the input. Figure 14 shows a screenshot of an sample Matlab interface screen.

Development in Excel began when Matlab development concluded by presenting the remaining candidate evaluation elements prior to capturing stakeholder preferences. Excel is a better development tool than Matlab to examine alternative strategies of *Stakeholder Preference* handling. Excel is utilized in order to efficiently step through multiple preference handling methods and compare them. The main advantage of using Excel is that it allowed specific stakeholder preferences to be adjusted to observe their impact on the overall output. This process was determined to be more efficient than using Matlab.


<Student Version> Figure 2: Potential Environemnts
File Edit View Insert Tools Desktop Window Help
UPDATE Environments *****MRED INPUT*****
CONTINUE with MRED

Potential Lab Environments

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

Potential Simulated Environments

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

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?

0 3 50

0 2 50

0 3 50

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

	ABC Controls Lab	ABC Robotics Lab	ABC Force/Torque Lab
C4: Rev Joint 3	1	0	1
C5: Pris Joint 2	1	0	1
P1: X Trans	0	1	0
P2: Y Trans	0	1	0
P3: Z Trans	0	1	0

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

	ABC Test Assembly Line	DEF Test Manufacturing Workstation
C4: Rev Joint 3	0	0
C5: Pris Joint 2	0	0
P1: X Trans	1	1
P2: Y Trans	1	1
P3: Z Trans	1	1

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

Figure 14: Matlab Screen Capture of MRED Interface - Environment Input and Relationships

3.9. *Verification and Validation*

Verification and Validation was done using using test plans developed at NIST. This dissertation's author selected those test plans that he was involved in designing and implementing. The test plan blueprints are abstracted from the detailed test plans to provide a means of comparison to MRED-generated blueprints.

Likewise, test plan inputs are also obtained by reviewing the previously-generated test plans. These inputs and outputs served as the benchmark for verification and validation of MRED.

3.9.1. Example Selection

A robot arm and speech to speech translation technologies are selected as examples for MRED because they contain several key characteristics.

- Development State
    - Robot Arm – This example portrays a technology in the early stages of development where many of its components and capabilities (and therefore, the system) are not fully-developed.
    - Speech to Speech Translation – This example portrays a technology more developed than the robot arm, yet still not fully-developed.
- Evaluation Emphasis
    - Robot Arm – This example targets the generation of blueprints to evaluate hardware elements
    - Speech to Speech Translation – This example focuses on generating blueprints to evaluate software elements
- Metric Range – Both examples necessitate the capture of quantitative and qualitative metrics across the component, capability, and system levels of the technologies.
- Test Plan Acess – The speech to speech translation example is selected given that the dissertation author has extensive experience evaluating these technologies and was a core member in devising and successfully implementing relevant test plans.

3.9.2. The factors noted above played a decisive role in the selection of examples for this research effort. Verification

Verification answers the question “Is it built right?” (discussed in 3.1.1). Verification is conducted several times during the course of MRED’s development to ensure MRED is “built right” (see Section 4.7). Specifically, verification is conducted on the following:

- Output Blueprint Elements – Test plan blueprint elements are extracted from previously generated test plans and are modeled as MRED blueprints. Verification checks that MRED’s blueprint model sufficiently presents the output information necessary to produce the corresponding test plans.
- MRED Algorithm – Verification is performed on the MRED algorithm at the conclusion of the algorithm’s development. The algorithm is verified by comparing the output blueprints generated from MRED based upon inputs from an example technology. The comparison of the blueprints focuses on whether or not the MRED output conforms to the blueprint element relationships (discussed in Section 3.6 and accurately reflect the stakeholders’ preferences (discussed in Section 3.7).

Validation of MRED occurs after it is verified.

3.9.3. Validation

Validation answers the question “is the right thing built?” (discussed in Section 3.1.1). Validation begins by taking input from the prior design of speech to speech technology evaluation test plans generated at NIST (see Chapter 5). This data is input into MRED and the output is compared to the extracted blueprints (from the original test plans) for accuracy and conformity; not only should the output conform to the verification requirements (conform to the blueprint element relationships and reflect the stakeholders' preferenecs), it should also be consistent with the test plans that were created using prior test methods.

3.10. Summary

Test planning methods increase in importance as do product development processes. Systems Engineering is one such development process that has been adopted by many technology developers. Although it has proven effective in generating products on time, on budget and to specification, its focus on the test planning process is sparse.

The history of evaluation test planning design is full of methods that are tailored to specific technologies. The US&R program did not apply any formal methods for creating their test plans nor did they formally solicit technology developers for their preferences. The S2S method did apply the SCORE framework to generate test plans, yet this did not include any rigorous methods in capturing stakeholder preferences. Understanding the test planning methods that designed these technology evaluations also brought to light the importance of developing test plan blueprints. Output blueprint elements are identified based upon the key pieces of information that drive the test planning process.

The methodology established to develop MRED was founded on the dissertation author's experience in designing and implementing test plans for over a decade. Part of this methodology was to identify where MRED would realistically fit into the technology development process. In addition, the dissertation author's evaluation design and implementation experience informed him on the necessity of identifying and using relationships among the various MRED input and output elements. The goal was to model a process that had not been modeled before in a way that was consistent with the experience of evaluation designers. The result is a methodology that combines practical evaluation design experience with mathematical methods proven in the literature.

Chapter 4: Multi-Relationship Evaluation Design (MRED)

This chapter presents the Multi-Relationship Evaluation Design (MRED) methodology and algorithm. Before discussing the MRED in detail, a robotic arm example is presented. This example is referenced throughout this chapter to aid in explaining MRED's operations. The overall model is then presented in Section 4.2 followed by descriptions of the input into MRED and its output blueprint elements. Next, key relationships among inputs and those between input and output elements are detailed. In Section 4.6, the MRED process is presented including the mathematical equations used to generate evaluation blueprints. A robotic arm example is defined to explain and verify the MRED process.

4.1. Robotic Arm Example

An example robotic arm[6], shown in Figure 15, is used to present the MRED process (Weiss and Schmidt, 2012). The reference frame of these capabilities is the coordinate frame at the tool point with respect to the base shown in Figure 15. The arm depicted in Figure 15 weighs ~500lbs and has a reach of ~63". For the sake of discussion, it is assumed that the robot arm is being designed and built to primarily function in an automobile manufacturing facility. Of course, this type of arm could reasonably be deployed in other types of manufacturing facilities and across other industries. This example is further elaborated as MRED is defined.

[6] The example arm is based upon a real industrial arm robot with revolute joints and no gripper. Two prismatic joints and a gripper were added to make the example more complex.

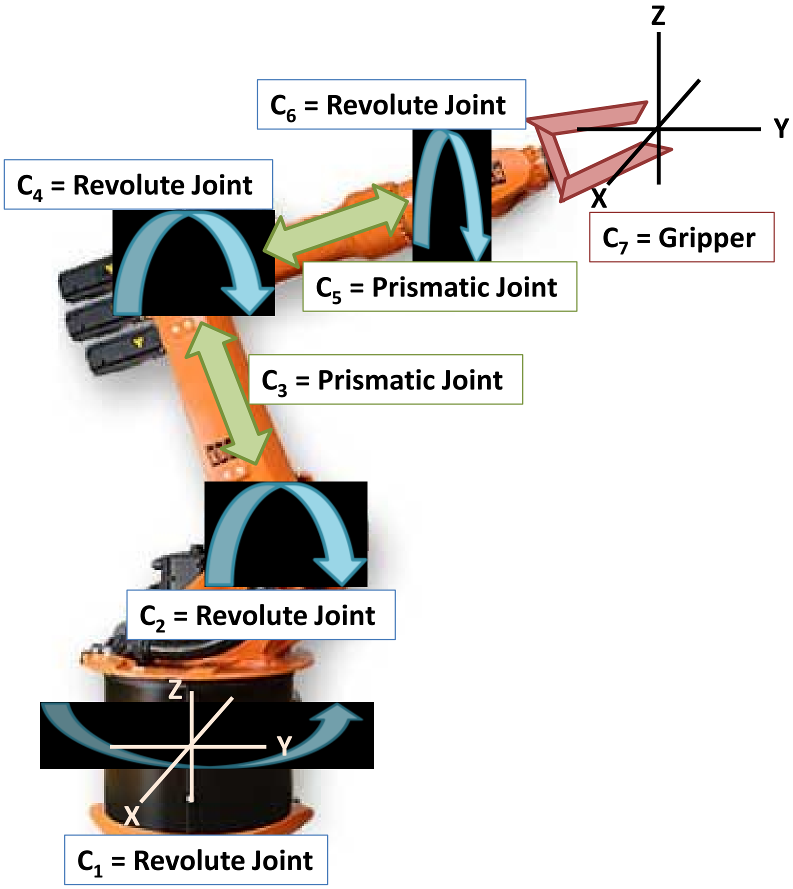


Figure 15: Example Robotic Arm[7]

4.2. *Overall Model*

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. During this process MRED invokes the relationships among the inputs and the impacts the inputs have on the outputs to generate one or more sets of evaluation blueprints. The overall model, including input and output, is shown in Figure 16. The model requires six different types of input in order for it to output one or more evaluation blueprints. The person responsible for inputting this

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

information into MRED is defined as the *MRED Operator*. The *MRED Operator* will be discussed in greater detail as the *Stakeholders* are introduced in 4.3.

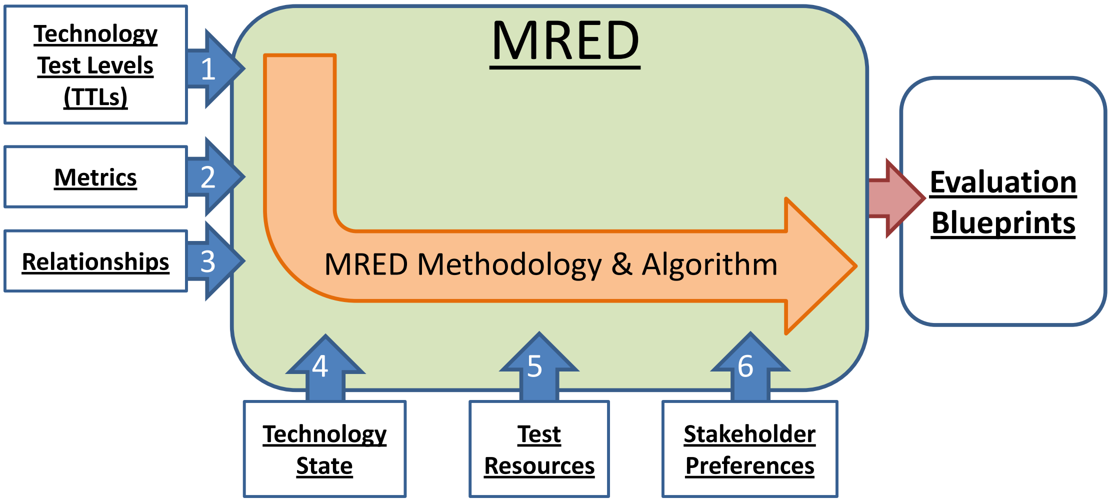


Figure 16: Overall MRED Model

4.3. *Inputs*

4.3.1. Technology Test Levels

The first and most important input into MRED is a description of the *Technology Test Levels (TTLs)*. *TTLs* are defined as the technology's (or *System's*) constituent *Components* and *Capabilities* (Weiss et al., 2010). MRED approaches a technology and its evaluation from a hierarchical perspective; it's important that the first step be to understand the levels of the technology. *Technology Test Levels* are synonomous with the technology elements defined in Section 3.5:

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

Given *TTL* terminology, the example robotic arm is 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; and 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 motion directions 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). The *MRED Operator* is responsible for distinguishing which *Components* and *Capabilities* will be input into MRED for test consideration. The dissertation author, acting as the *MRED Operator*[8]*,* has chosen to define seven *Components* and seven *Capabilities* for testing. However, different *Components* could be identified for testing including gears, motors, and actuators. If these specific pieces were constructed in-house and/or specially for this technology, then it may be practical to include them for test consideration.

4.3.2. Metrics

Test events are capable of collecting data that can be divided into two unique types of metrics. Before these two specific types are defined, it's important to differentiate between metrics and measures in the context of MRED (Weiss et al., 2010).

[8] For the robot arm example, the dissertation author is acting as all of the *Stakeholders* in addition to the *MRED Operator*. Reasonable values are being supplied as input for illustration purposes.

- *Measures* – A performance indicator that can be observed, examined, detected and/or perceived either manually (by human means such as a person pressing a stopwatch or measuring a distance) or automatically with a tool (such as emplacing a motion detector).
- *Metrics* – The interpretation of one or more contributing pieces of data that represent performance. *Metrics* may be composed of other *Metrics* and/or *Measures*. For example, the *Metric* of velocity may be directly captured using a radar gun. Likewise, velocity may also be captured by measuring *distance* and *time* where velocity = distance/time. Distance and time are both elements that contribute to the quality of the *Metric* velocity.

To reiterate from Section 3.5.4, the two types of *Metrics* are:

- *Technical Performance* – *Metrics* related to quantitative factors (e.g. accuracy, distance, time, etc.). These metrics may be required by the program sponsor, to meet user expectations, inform the technology developers on their design, etc.
- *Utility Assessments* – *Metrics* related to qualitative factors that express the condition or status of being useful and usable to the target user population.

Like *Technical Performance*, *Utility Assessment Metrics* may be of value to any and/or all of the evaluation stakeholders. In the case of the robot arm, some sample *Metrics* are:

- *Technical Performance* – Maximum Force, Maximum Linear Velocity, Range of Motion, Maximum Lift Capacity
- *Utility Assessment* – Responsiveness, Smoothness, Operator Satisfaction

4.3.3. Technology State – Maturity

MRED defines *Technology State* as a technology's fitness for testing. *Technology State* is described by the factor of *Maturity* (Weiss and Schmidt, 2011c). *Maturity* is the fitness for operation of individual *Components, Capabilities,* and the *System*. A technology's *Maturity* has a direct impact on whether a specific *TTL* is ready for testing and what, if not all, functions are available. 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 operational. Throughout the technology's development cycle, its *Maturity* is constantly updating. For instance, if several *Components* have *Maturity* value of fully-developed, then there are no technological restrictions on testing. If the *Components* are are not fully-developed, then either limited or no testing can occur. *Component Maturity* will be demonstrated with respect to the robot arm in the following subsections. *Capability Maturity* will be presented in later sections.

Maturity, must be input into MRED for a *TTL* 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:

- Immature – The *Technology Test Level* being tested has yet to be developed or is still in the process of being developed. This state combines the two states of

"System is Immature" and "System Maturity is in Progress" discussed in Section 2.1.

- Fully-Developed – The *Technology Test Level* is developed to the point of being operational and complete. A *TTL* that is classified as *Fully-Developed* has all behaviors available. This state is comparable to the maturity state of "System Maturity has been Achieved" discussed in Section 2.1.

The *Maturity* information for a technology's *Components* 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 have the most up-to-date information. The *Maturity* of *Capabilities* and the *System* is either provided by the technology developer (for *TTLs* that are less than Fully-Developed) or by MRED calculations (for *TTLs* that are Fully-Developed). The TRL definitions, presented in Section 2.1, are not relevant to MRED's concept of *Maturity*. This is because TRLs are defined for an entire technology as opposed to being defined for a technology's constituent physical and functional elements. TRLs cannot be defined for individual *Components, Capabilities,* or the *System* as they are defined in MRED. A *Component* cannot be tested at TRL-7 or above since these TRLs require a system prototype demonstration in the target environment. A TRL looks at the full technology whereas MRED requires a means of assessing the *Maturity* of individual elements. Another concern with TRLs is that a TRL can only be reasonably assigned after a technology has undergone a demonstration or evaluation in the corresponding conditions. Otherwise, it is up to the *Stakeholders* to note the existing TRL of a technology (based upon past exercises) and make a judgment as to whether or not the technology is ready to be tested in the next greatest TRL.

The specific approach to determining *Capability* and *System* maturity is defined in Section 4.5. In the case of the example robot arm, the *MRED Operator* defines C_1 and C_2 as Fully-Developed and C_3, C_4, C_5, C_6, and C_7 as Immature.

4.3.4. Resources

This category of inputs signifies the availability of the candidate *Environments, Tools,* and *Personnel*.

4.3.4.1. Environments

The *Environments* are defined as the physical venue, supporting infrastructure, artifacts, and props that will support the test(s) (Weiss and Schmidt, 2010; Weiss and Schmidt, 2011a). The setting in which the evaluation occurs can have a significant effect on the data. The testing *Environment* can influence the behavior of the personnel and can limit which levels of a technology can be evaluated. MRED defines three different *Environments:*

- *Lab* – Controlled environment where test variables and parameters can be isolated and manipulated to determine how they impact *TTL* performance and/or the technology user's perception of the technology's utility.
- *Simulated Environment*– Environment outside of the *Lab* that is less controlled and limits the evaluation team's ability to control influencing variables and parameters. This environment tests the technology in a more realistic venue. In this work, a *Simulated Environment* is a combination of controlled elements found in the *Lab* coupled with realistic environmental features found in the *Actual*

Environment. The *Simulated Environment* is usually a physical place as opposed to a simulation (created virtually) and is typically constructed for test purposes. Exceptions may exist if the candidate technology is itself virtual.

- *Actual Environment*– Domain of operations in which that the technology is intended to be used. This environment is the least controlled by the evaluation personnel given that any controls introduced would potentially alter this environment's reality. The evaluation team is limited in the data they can collect since they do not wish to control environmental variables. It is critical that all evaluation personnel and/or data collection equipment be transparent to the technology in the *Actual Environment*. If testing impacts the technology, then the *Actual Environment* is becomes more of a *Simulated Environment.*

Some environments that the robotic arm could be tested include:

- *Lab Environment* – Controls or robotics lab (e.g., a heavily-instrumented laboratory space affording the evaluation designer maximum control over the technology and environment)
- *Simulated Environment* – Manufacturing workstation build for testing (e.g., an isolated work cell on a factory floor that is not integrated into a working assembly line)
- *Actual Environment* – Assembly line where vehicles are produced

4.3.4.2. Tools

Tools are defined as the equipment that will collect quantitative and/or qualitative data during test events to support the generation of the necessary *Metrics*. *Tools* also include the equipment used to analyze or process captured data following the test event to produce the necessary *Metrics*. *Tools* are broken down into those supporting the capture of *Technical Performance* and *Utility Assessment Metrics*. Some example tools for the robot arm include:

- *Technical Performance Tools* – Tension Sensor (to support the *Metrics* of Maximum Force and Maximum Lift Capacity) and LADAR (to support the *Metric* of Maximum Linear Velocity and Range of Motion).
- *Utility Assessment Tools* – Web-based surveys and semi-structured interviews to support all of the afore-mentioned *Utility Assessment Metrics*.

Of course, additional tools may be available to capture data for the metrics.

4.3.4.3. Personnel

The *Personnel* are the individuals who will use the technology and indirectly interact with the technology during the test events (Weiss et al., 2010). *Personnel* can be classified into two categories: primary (those with direct interaction with the technology) and secondary (those with indirect interaction with the technology). The primary *Personnel* are classified as *Technology Users* (*Tech Users*) and are composed of three specific types of individuals:

- *Tech User* – Individual(s) that directly interacts with the technology during the test event. These individuals receive any training necessary to use the technology and are responsible for engaging/disengaging the technology's usage during the test event. *Tech Users* are typically the dominant source of qualitative data when evaluation goals require the capture of *Utility Assessments.* Three classes of *Tech Users* are defined below.

- *End-User* – Individuals that are the intended users of the technology. Depending upon the level and extent of the evaluation, all, some, or none of the *Tech Users* will be from the *End-User* class.
- *Trained User* (*Trn User*) – Individuals selected to be *Tech Users,* but are not *End-Users*. They receive all of the necessary training that *End Users* would receive, yet do not have the operational background or experiences of the *End Users* within the technology's targeted use-case environment(s).
- *Tech Developer* (*Tech Dev*) – Members of the organization that developed the technology being considered for testing. This personnel category does not have the operational background or experiences of an *End User*, yet they usually are deeply familiar with the technology's operations. *Tech Developers* may be the *Tech Users* depending upon the level and extent of the required testing. If so, then they may not require the full training complement.

An example of an *End-User* with respect to the robot arm would be the factory employee whose primary responsibility is to operate and/or monitor the arm. A *Trn User* could be an individual who is brought in from another industry to purely test the arm. The critical distinction is that the *Trn User* is neither the *End-User* nor the *Tech Dev*; a *Trn User* should have an unbiased opinion of the technology. The *Tech Dev* would be a representative from the company that manufactured the robot and has a working technical knowledge of this technology as opposed to being in a non-technical position (e.g. accountant, etc.).

The secondary personnel are those that indirectly interact with the technology and fall into the following two categories:

- *Team Member* – Individuals that work with *Tech Users* during the evaluation to realistically support the use-case scenario in which the technology is immersed. *Team Members* may be in a position to indirectly interact with the technology during the evaluation, but they are often in a situation to observe a *Tech User's* interactions with the technology. *Team Members* may be requested to provide their perceptions of a *Tech User's* use of the technology along with the *Tech User's* perceived level of situational awareness while using the technology, etc. *Team Members* may also be designated as secondary users in real situations meaning they would have some technology training.
- *Participant* – An individual that indirectly interacts with the technology during an evaluation. Typically, *Participants* are given specific tasks to either interact with the *Tech Users* and/or with the environment, but not with the technology (unless directed to do so by a *Tech User*).

In the context of evaluating the robot arm a *Team Member* could be another technician on the assembly line that works closely with the *Tech User* (i.e. robot arm operator). Likewise, a *Participant* could be anyone that is walking on the factory floor in close proximity to the robot arm, yet they are not a *Team Member*.

4.3.5. Stakeholder Preferences

Stakeholder preferences represent the desires of an individual or group of *Stakeholders* and are provided by the evaluation *Stakeholders* themselves. A *Stakeholder* is someone who has a vested interest in the technology, and therefore the evaluation. *Stakeholders* are classified into five categories which are presented in Table 9. Members of these categories have their own motivations when providing their preferences for the test plans. Likewise, they usually have differing interests in the results of the technology's performance at the conclusion of testing.

Table 9 - Stakeholder Categories (Weiss and Schmidt, 2011b)

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

There may be some overlap among the *Stakeholders* which occurs on a technology-by-technology basis. These possible relationships are shown in Figure 17.

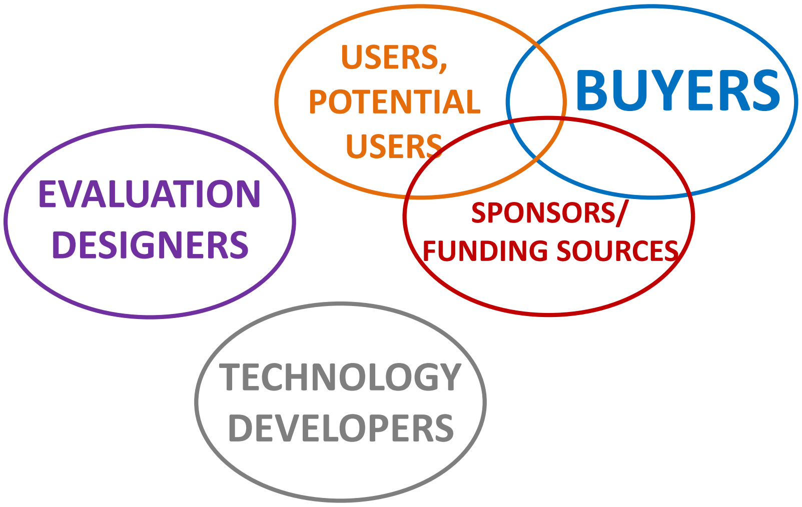


Figure 17: Potential Stakeholder Relationships

Technology Developers should not belong to any other *Stakeholder* category. *Technology Developers* have a natural bias to promote their system and see it do well in an evaluation, especially if it is tested against other comparable technologies. It's not unusual for *Technology Developers* to interact with *Evaluation Designers*. This enables the *Evaluation Designers* to get the latest information on the anticipated technology. The *Technology Developers'* test plan input is important, yet should be tempered by this implicit bias.

Other stakeholders are more impartial; *Sponsors* want to know the performance of the technologies they are funding; *Buyers* want to see benefits of technologies they are hoping to purchase; *Users* (including potential users) want clear evidence of the technologies' capabilities; and *Evaluation Designers* are impartial personnel whose goal is to design and execute fair and relevant tests. For example, personnel from NIST have acted as *Evaluation Designers,* and also as test executors, throughout many test events. In most instances, the NIST teams have had significant input into the test plans and served in an advisory capacity to other test *Stakeholders*.

For the robot arm, the *Stakeholders* could reasonably be:

- *Buyer* – the company that owns the manufacturing facility and are seeking to purchase a robot arm(s).
- *Evaluation Designer* – an unbiased third-party or a government agency (e.g. NIST) that has expertise in test plan generation.
- *Sponsor* – a venture-capitalist or government agency that is motivated to see the robot arm developed.
- *Technology Developer* – the company who designs and builds the robot arm
- *User* – the individual or group who are expected to use and/or operate the robot arm within its intended operating environment(s)

The exact nature of the *Stakeholder Preferences* will be elaborated upon in 4.6.3 when *Stakeholder Preferences* are captured and handled in the MRED process.

4.4. Output Elements

MRED processes all of the input information to output one or more evaluation blueprints with specific test plan elements. These output elements were directly input into MRED (e.g. *Environments, Personnel*, etc.), several inputs organized together (i.e. *TTL-Metric Pairs*) or derived from *Stakeholder Preferences* (e.g. *Explicit Environmental Factors, Evaluation Scenarios*, etc.). Each of the output elements is presented in the following subsections. In addition, there are relationships between specific output elements. These relationships are defined in the following subsections once each of the contributing output elements is detailed.

4.4.1. Technology Test Level – Metric (TTL-Metric) Pairs

A *TTL-Metric* pair is defined as a specific *TTL* that is coupled with a *Metric* that can be generated from testing this specific *TTL* (Weiss and Schmidt, 2012). The value of this output blueprint coupling is that most *TTLs* (if not all) can have more than one *Metric* captured during their evaluation. Capturing multiple *Metrics* from a single *TTL* is actually encouraged since it likely reduces the testing cost per *Metric*. Some *TTL-Metric* pairs that can be defined using the robot arm include C_1 (Revolute Joint) – Range of Motion, C_2 (Revolute Joint) – Range of Motion, C_3 (Prismatic Joint) – Range of Motion, C_3 (Prismatic Joint) – Maximum Linear Velocity, P_1 (X Translation) – Maximum Linear Velocity, P_1 (X Translation) – Responsiveness, System – Range of Motion, and System – Responsiveness (this is a small subset of the candidate *TTL-Metric* pairs). This collection of *TTL-Metric* presents the situations that can arise when producing test plans for a complex system. They are:

- Capturing the same *Metric* from similar *TTLs* of the same type – the Range of Motion *Metric* being paired with the two revolute joints, C_1 and C_2, is an example of when the same metric can be captured from two separate *TTLs*. It's likely that the same test plans would be sufficient to capture metrics from both of these *TTLs* given that they are similar *Components*.
- Capturing the same *Metric* from different *TTLs* of the same type – the Range of Motion *Metric* being paired with a revolute joint, C_2, and a prismatic joint, C_3, demonstrates this concept. Although Range of Motion can be measured from both *Components*, it's likely the test plans will look very different since range of motion of a revolute joint is logically measured in degrees (or radians) while the range of motion of a prismatic joint would be measured in length (feet, inches, meters, etc.).
- Capturing the same *Metric* from different *TTLs* of a different type – the Range of Motion *Metric*, coupled with the *Component* revolute joint, C_2, and the *Capability* of X Translation, P_1, is an example of this concept. This situation is often similar to that of the same *Metrics* being produced from testing different *TTLs* of the same type in that the test plans would likely be unique from one another.
- Capturing different *Metrics* from the same *TTL* – this is one of the most common cases and is easily illustrated with C_3 being paired with Range of Motion and Maximum Linear Velocity.

Grouping of *TTL-Metric* pairs is an option provided to the *MRED Operator* given the presence of the situations. The benefit of grouping is that it enables *Stakeholders* to provide preferences for a group of pairs as opposed to individual pairs. This creates greater efficiency in capturing and handling *Stakeholder Preferences* since *Stakeholders* would not have to rate as many options. Grouping is discussed in greater detail in 4.6.

4.4.2. Personnel

For every test plan, primary *Personnel* are assigned to act as *Tech Users*. They are *End-Users, Trained Users,* or *Technology Developers* as defined in 4.3.4.3. A test plan may also include the presence of secondary *Personnel* which could be *Team Members* and/or *Participants* (also defined in 4.3.4.3). MRED would output blueprint *Personnel* specifications which would be a subset of those available *Personnel* input (discussed in Section 4.3.4.3): an *End-User* would be an employee whose primary responsibility is operating and/or monitoring the robot arm; a *Tech Dev* would be a representative from the company that develops the arm and has a working technical knowledge of the robot arm; etc. In addition to *Personnel* being identified in the output test plans, *Knowledge* and *Autonomy Levels* for these individuals is also defined. *Knowledge* and *Autonomy Levels* are defined in the following section.

4.4.3. Knowledge and Autonomy Levels

The *Tech Users, Team Members,* and *Participants* involved in a test plan have varying levels of knowledge about the functionality and usage of the technology in addition to the testing environments (Weiss and Schmidt, 2011b). The scope of knowledge and their specific levels are defined by MRED for each test plan. The levels are defined as (Weiss and Schmidt, 2011b):

- *Operational Knowledge* – The level of practical information and experience an individual has about the *Actual* environment, the intended use-case situations for the technology and other pre-existing technologies that the technology under test supports. Varying levels of *Operational Knowledge* can be attained through real-world experience, repetitive training, trial and error exercises, etc.
- *Technical Knowledge* – The level of information and experience an individual has about the technology and how it should be employed to maximize success. *Technical Knowledge* is acquired through training and/or repetitive use of the technology.

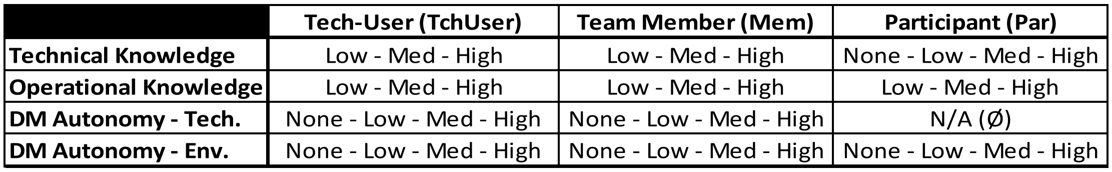
The *Tech Users*, *Team Members*, and *Participants* assigned within a test plan are assigned specific decision-making (DM) autonomy levels (Weiss et al., 2010; Weiss and Schmidt, 2011b). Autonomy scope and levels are set by MRED for each evaluation. Personnel could be fully restricted in their decision-making (i.e., no *DM Autonomy*), which requires that the evaluation plan design includes scripted actions. Alternatively, personnel may have unbounded authority where each individual is free to exercise their judgment. There are two types of *DM Autonomy* which are defined below:

- *DM Autonomy – Technical* (also known as *Technical Autonomy*) – This refers to the level of authority that the *Tech Users* have in operating the technology. Depending upon the specific evaluation, instructions provided to *Tech Users* could range from being restricted to using certain features of a technology, to being free to use any or all of its features as they see fit. *Team Members* may also be allowed a level of *DM – Autonomy – Technical* if the test plans provide the potential for these *Personnel* to use the technology at any point of the evaluation. Since *Participants* do not have any direct interactions with the technology, they are not afforded any *DM – Autonomy – Technical*.

- *DM Autonomy – Environmental* (also known as *Environmental Autonomy*) – This refers to the level of authority that the *Personnel* have in interacting with each other and the environment.

Each *Personnel* member's knowledge and autonomy levels range from "Not Applicable (N/A)" to "High" as specified by MRED in output test plan(s). *Autonomy Levels* must be equal or lower in value than their partner *Knowledge Levels*. Determination of *Autonomy Levels* is limited by multiple factors including candidate *Technology Test Levels, Tech User type*, etc. and ultimately determined by the *Stakeholders* in their preferences. The potential knowledge and autonomy levels for the evaluation participants are shown in Table 10.

Table 10 – Knowledge & Autonomy Ranges (Weiss and Schmidt, 2011b)



	Tech-User (TchUser)	Team Member (Mem)	Participant (Par)
Technical Knowledge	Low - Med - High	Low - Med - High	None - Low - Med - High
Operational Knowledge	Low - Med - High	Low - Med - High	Low - Med - High
DM Autonomy - Tech.	None - Low - Med - High	None - Low - Med - High	N/A (Ø)
DM Autonomy - Env.	None - Low - Med - High	None - Low - Med - High	None - Low - Med - High

"None" means that this *Personnel* group has no knowledge in a specific area, DM authority either over the technology and/or how they behave within the environment. "Low" means that this *Personnel* group has a small amount of knowledge or their DM autonomy is significantly limited in a specific area. "Med" (medium) means that this *Personnel* group has an average amount of knowledge and is given some DM autonomy in a specific area. "High" means that this personnel group has expert and/or extensive knowledge or full DM autonomy in a specific area. For example, suppose the automotive industry is testing the robot arm and a manufacturing employee is assigned as the *End-User*. The employee can be categorized as having no technical knowledge of the robot arm when they are seeing it for the first time. After an hour of basic training on the arm, it could be reasonably stated that the employee has “Low” technical knowledge of the system; after a week of training during some simulated situations, it could be stated that the employee has “Medium” technical knowledge of the technology; and after a month of continuous usage of the robot arm in realistic environments (e.g. automotive assembly line) it could be stated that the employee has a “High” amount of technical knowledge.

Similar statements could be made about the employee’s level of *Operational Knowledge* when they first take a job in this field; “None” - no operational knowledge, “Low” after having worked on the factory floor for a week at a single manufacturing station, “Medium” after having worked on the factory floor for a month at several manufacturing stations, and “High” after having worked many months in the manufacturing facility and becoming familiar with a majority of the manufacturing stations (high operational knowledge). Typically, all *Tech Users* (no matter what sub-group they fall into) have at least “Low” technical and operational knowledge prior to the evaluation due to initial system training and/or background information on their scenario objective (to support at least a minimal amount of *Operational Knowledge*).

The knowledge level of a *Personnel* group (i.e. “None,” “Low,” “Med,” or “High”) is dependent upon the technology and the environment. This dependency is demonstrated in the US&R and S2S technologies discussed in prior sections. Recall that the *End-Users* of US&R robots are first responders (local, state, or federal rescue personnel) while the *End-Users* of the S2S technology are US military personnel (Soldiers and/or Marines). US&R robots are deployed to assist first responders in search and rescue operations necessitated by earthquakes, tsunamis, building collapses, etc. Fortunately, these disasters are infrequent so first responders spend more time training on this technology as compared to deploying it in real situations. S2S technologies can be used on a daily basis during a Soldier's or Marine's deployment. Military personnel can be deployed for 9 to 15 months and tasked to interact with a foreign population for a majority (if not all) of this timeframe. It's reasonable to state that the average Soldier or Marine armed with an S2S technology uses this more frequently in a real use-case as compared to the average US&R robot operator. Only informed personnel within these domains could qualify what they consider "Low" on the knowledge level scale. Military personnel may state that anyone who has had less than an hour of training on the S2S technology is considered to have "Low" *Technical Knowledge*. However, first response personnel may consider anyone who has had less than 8 hours of training on a US&R robot to have "Low" *Technical Knowledge*. Similar statements could also be made with respect to *Operational Knowledge.*

4.4.4. Relationships – Personnel, Knowledge, and Autonomy

MRED defines a relationship between knowledge and autonomy in which the knowledge levels constrain autonomy levels (i.e., where autonomy cannot exceed knowledge). This rule holds for the *DM Autonomy – Technical* and *Technical*

Knowledge pair and the *DM Autonomy – Environmental* and *Operational Knowledge* pair. For example, a *Tech Developer*, who knows the intricacies of the new technology, may be assigned as the *Tech User* to test a specific *Capability.* This could be the result of MRED's output test plan stating that the *Tech User's Technical Knowledge* should be high. Furthermore, MRED could further dictate that the *Tech User* should have no *DM Autonomy – Technical* ("None") in the evaluation. In effect, this becomes a scripted test. However, MRED would not output a blueprint where a *Tech User* is required to have a "High" level of *DM Autonomy – Environmental* and a lesser level ("Medium" or lower) of *Operational Knowledge*. This would require the *Tech Users* to have authority in an area that is beyond their knowledge. This would not be considered a practical test plan since the *Tech User's* actions and responses are likely to be inappropriate and unrepresentative (since they have not had the training or experience in the given environment to act accordingly).

End-Users, *Trained Users* and *Tech Developers* are specific cases of *Tech Users* with their own characteristics. For example, *End-Users* will most likely have a greater level of *Operational Knowledge* and a lower level of *Technical Knowledge* compared to the other *Tech Users*. *End-User DM Autonomy* in both technical and operational categories will vary given the *TTLs* considered for testing, the *Metrics* to be captured, the *Environment(s)* under consideration and the *Stakeholder Preferences*. *Trained Users* will most likely have no ("None") to "Low" *Operational Knowledge* and *Technical Knowledge*. It is also likely that their *DM Autonomy* in both technical and operational categories will be significantly limited since their knowledge is also limited. *Tech Developers* are likely to have no ("None") to very little ("Low")

Operational Knowledge, but very “High” (if not expert) levels of *Technical Knowledge*. Their *DM Autonomy – Environmental* will probably be limited (due to their “Low” *Operational Knowledge*), but their *DM Autonomy – Technical* could range from “Low” to “High.” Of course, some exceptions may exist. For example, a former manufacturing facility employee may now be a *Tech Developer* on an emerging robot arm technology.

4.4.5. Environment

The setting in which the evaluation occurs can have a significant effect on the data since the *Environment* can influence the behavior of the *Personnel* and can limit which *TTLs* can be evaluated. Each MRED test plan will output a specified *Environment* that can be classified as a *Lab, Simulated,* or *Actual Environment* (as defined in 4). The relationships the *Environments* have with other output elements are discussed in Section 4.4.6 and Section 4.4.10.

4.4.6. Relationships – TTLs, Metrics, Tech Users, & Environments

There is a progression of test plans from controlled and restrained to natural and actual among the *TTLs*, *Tech Users*, and *Environments* as the technology develops. Altogether, there are numerous interdependencies among these three elements and *Metrics*.

The most basic pieces of a *System* are the *Components*. For evaluation purposes, *Components* need to be combined with other *Components* to comprise the entire *System*. The highest technology level is the *System*. Tests at the *System* level are influenced by the most inputs (as compared to the *Component* and *Capability* levels) and therefore yield a wider range of outputs. *Capabilities* are produced from *Components* interacting together to produce a specific action or function.

Not every *Tech User* is ideally suited to test the technology across all of the *TTLs*. For example, the *Components* are typically not elements of the *System* that *End-Users* would see during their natural deployment nor would it be practical to collect *Utility Assessments* here. Tests at these levels would be best left to the *Tech Developers* to act as the operators since they have the deepest understanding of the technology (compared to the other *Tech User* groups). If there are concerns with the *Tech Developers* acting as *Tech Users* in the evaluation, then *Trained Users* can be brought in to serve as the *Tech Users. Component* evaluations yield *Technical Performance* data as opposed to *Utility Assessment* data so the evaluation would not require *End-Users* for technology feedback. The *Tech User* pool greatly expands at the *Capability* level since *Capabilities* are something that the *End-Users* could naturally use.

Technical Performance and *Utility Assessments* are now related to *TTLs, Tech Users* and the *Environments*. Typically, when an advanced technology or intelligent system is in its infancy it's not ready for the *End User*. Early tests are usually conducted with *Tech Developers* as the *Tech Users* since it's likely that more issues will arise that they are better equipped to communicate about and efficiently address. Additionally, *Technical Performance* testing at these early stages can be more insightful than *Utility Assessments* to see if the *System* and/or its *Components* are working as intended. This is not to say that *Utility Assessments* are not important at the early stages of development. These metrics will still be useful in informing the

Technology Developers on *Tech User* perceptions of the technology. As a technology matures and *Capabilities* and the *System* become available for testing, it becomes more practical to get *Tech User Utility Assessment Metrics*, especially from the *End-User* community. A technology is going to have an easier time being adopted by the intended *End-User* community if their input is solicited during the development process.

Table 11 presents MRED's practical constraints relating *Tech Users, TTLs,* and *Metrics*. The two primary *Personnel* restrictions MRED places are: 1) *End-Users* should not evaluate a technology at the *Component* level and 2) *Tech Developers* should not be the *Tech Users* in any tests that generate *Utility Assessment Metrics*. The first restriction is in place since *End-Users* will never interact with *Components* during practical usage of the technology whereby it would be more efficient to select *Tech Users* who are more familiar with *Component* operations. The second restriction is in place since the *Technology Developers* will have a natural bias to their technology that could skew any *Utility Assessment Metrics.*

Table 11 - MRED constraints on Personnel at Testing Specific TTLs and Metrics

		TECHNICAL PERFORMANCE			UTILITY ASSESSMENT	
		Component	Capability	System	Capability	System
PRIMARY PERSONNEL	*Tech User: End-User*	NO	YES	YES	YES	YES
	Tech User: Trained User	YES	YES	YES	YES	YES
	Tech User: Tech Developer	YES	YES	YES	NO	NO

Employing different categories of *Tech Users* within an evaluation will produce results that can range from poor to optimal performance and from improper to proper usage of the technology. It is reasonably assumed that out of all of the potential *Tech Users*, the *End Users* will have the highest *Operational Knowledge* of the technologies' target usage environment, but will have the lowest *Technical Knowledge*. Conversely, the *Tech Developers,* assigned to act as *Tech Users,* will have the least *Operational Knowledge* of the technologies' target usage environments, but will have the greatest (if not complete) *Technical Knowledge*.

The *Environment* is now related to *TTLs, Metrics,* and *Tech Users*. Typically, immature technologies are evaluated in the *Lab* so that specific variables can be controlled in an effort to determine what impacts the technologies' performance and to what degree. As the technologies further develop, they are then evaluated in less controlled environments. Tests performed in *Simulated Environments* bring the *Stakeholders* one step closer to understanding how the technology behaves in more realistic environments. The technology is tested in the *Actual Environment* once it has significantly matured and nears its final design. Of course, it is possible to test an immature technology in an environment more advanced than its development (such as the *Simulated* or *Actual*), but it will be much more difficult to pinpoint the exact cause(s) of failure when the technology falters. The opposite is true; a very mature technology may be tested in a more basic environment (such as the *Lab* or *Simulated* depending upon the stage of development). However, it's likely that the results from these tests will be highly repeatable and not as practical (as compared to testing in a more advanced environment) to conduct after numerous test runs.

The evaluation pinnacle is to test a *System* in the *Actual* environment where the *Tech Users* are *End-Users*. At a minimum, *Utility Assessment* metrics could be collected to determine how well the technology aided the *End-User* in accomplishing their objective(s). Depending upon the nature of the *Environment*, certain *Technical Performance Metrics* could be captured to assist in validating the final design. This is as close to realistic usage of the technology as possible and therefore presents the truest indicator of how the technology would perform in common practice. It is understood that intelligent, advanced, and emerging technologies must go through numerous evaluations at idealized variable values within these four categories before the *System* can be tested in the *Actual* environment by the *End-Users*.

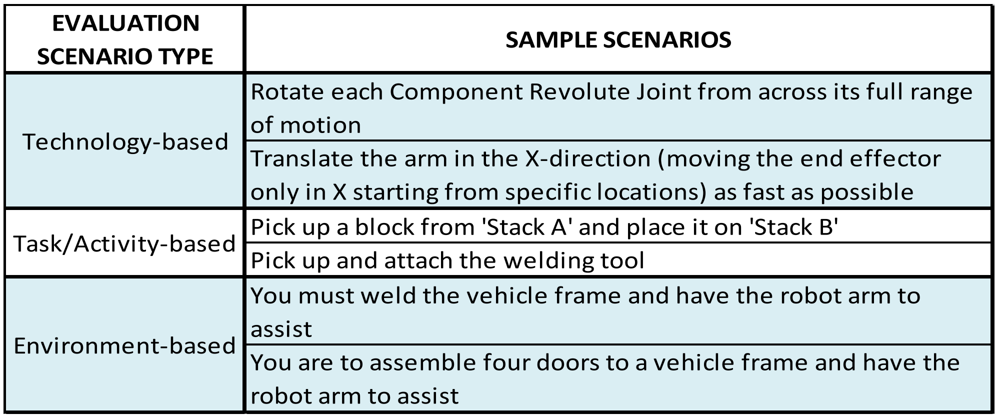
4.4.7. Evaluation Scenarios

The *Evaluation Scenarios* govern exactly what the technology will encounter within the *Environments*. Three types of *Evaluation Scenarios* are identified below. Each is unique in the relationships they have with Knowledge Levels, Autonomy Levels, and the *Environments*. The three *Evaluation Scenario* types are:

- *Technology-based – Evaluation Scenarios* in this category feature specific instructions to the *Tech User* in how they should use the technology within the *Environment*.
- *Task/Activity-based* – Type of *Evaluation Scenario* that specifies the *Tech User* complete a specific task within the *Environment* where they may use the technology as they see fit.
- *Environment-based* – Type of *Evaluation scenario* that enables the *Tech User* to perform the relevant activities within the *Environment* based upon an advanced Operational Knowledge and provided with a high-level objective.

Table 12 presents some sample scenarios by which to evaluate a robot arm. Note that these examples may be limited by available *TTLs, Environments, Personnel*, and other candidate evaluation elements.

Table 12 - Sample Evaluation Scenarios for Robot Arm



EVALUATION SCENARIO TYPE	SAMPLE SCENARIOS
Technology-based	Rotate each Component Revolute Joint from across its full range of motion
	Translate the arm in the X-direction (moving the end effector only in X starting from specific locations) as fast as possible
Task/Activity-based	Pick up a block from 'Stack A' and place it on 'Stack B'
	Pick up and attach the welding tool
Environment-based	You must weld the vehicle frame and have the robot arm to assist
	You are to assemble four doors to a vehicle frame and have the robot arm to assist

4.4.8. Relationships – Scenarios, Environments, Knowledge & Autonomy

Technology-based Evaluation Scenarios typically occur in the *Lab* or *Simulated* environments where the evaluation team can determine the exact test parameters and control the various test variables (Weiss and Schmidt, 2010). *Task/Activity-based Evaluation Scenarios* can occur across any of the three (*Lab*, *Simulated*, *Actual*) environments where the evaluation team has some measure of control of both the test parameters and variables. The *Environment-based Evaluation Scenarios* can only occur in the *Simulated* and *Actual* environments since these *Environments* are indicative of realistic operating *Environments*. The specific relationships among the *Evaluation Scenarios* and the *Tech User's* Knowledge Levels and Decision-making Autonomy are shown in Table 13.

Table 13 - Relationship among the Evaluation Scenarios, Environments, Knowledge and Autonomy Levels

EVALUATION SCENARIOS	TEST ENVIRONMENT(S)	KNOWLEDGE LEVEL		DECISION-MAKING AUTONOMY	
		TECHNICAL	*OPERATIONAL*	*TECHNICAL*	*ENVIRONMENTAL*
Technology	Lab, Simulated	M - H	L - M - H	N - L	N - L
Task/Activity	Lab, Simulated, Actual	L - M - H	L - M - H	L - M - H	L - M - H
Environment	Simulated, Actual	M - H	M - H	M - H	M - H

4.4.9. Explicit Environmental Factors

The *Explicit Environmental Factors* are characteristics within the environment that impact the technology and influence the actions of the *Personnel* thereby, affecting the outcome of the evaluation (Weiss and Schmidt, 2010). These factors pertain to the overall physical space, composed of *Participants* (constituent actors), 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 (shown in Figure 18).

- *Feature Density* – Refers to the number of features that impact the technology and influence the decision-making of the *Personnel* within the test *Environment*. The greater the *Feature Density*, the more challenging it is for a technology to effectively and efficiently interact within, identify objects/events/activities, operate within, etc. the *Environment*. *Feature Density* of a test *Environment* can be characterized as “Low,” “Medium,” and “High.”
- *Feature Complexity* – Refers to the intricacy of various features within the *Environment*. For example, a baseball (sphere) has a lower *Feature Complexity* as compared to a car. Similar to *Feature Density*, the greater the *Feature Complexity*, the more difficult it is for the technology to accurately and

appropriate operate and be beneficial to the *Tech User*. As with *Feature Density*, *Feature Complexity* can also be characterized as "Low," "Medium," and "High."

- *Overall Complexity* – This factor refers to the global combination of *Feature Density* and *Feature Complexity* within the *Environment*. *Overall Complexity* can range from "Low," "Low/Medium," "Medium," "Medium/High," and "High" is a result of an *Environment's Feature Density* and *Feature Complexity*.

An example of *Feature Density* in an *Environment* set up to test a robot arm would be the quantity of objects to interact with and/or detect. Providing the robot with a single object in its work-space is much less challenging than providing the arm with two or more objects. Similarly, providing the robot with three objects that have approximately a foot of space between them is much less challenging (to perceive and/or interact) than providing the same three objects lined up next to one another. An example of *Feature Complexity* within a robot arm test *Environment* would be the type of objects that the technology perceives and/or with which it interacts. It is probably less challenging for a robot to perceive a box (e.g. simple shape) as compared to a vehicle door (e.g. complex shape). The *MRED Operator*, in consultation with the other *Stakeholders*, determines what constitutes a "Low," "Medium," and "High" *Feature Density* and *Feature Complexity* since this factor is both dependent upon the technology and its operating environment. *Overall Complexity* results from the chosen *Feature Density* and *Feature Complexity* of an *Environment* according to Figure 18.

4.4.10. Relationships – Environments & Environmental Factors

Now that both *Environments* and *Explicit Environmental Factors* have been defined, the relationship between these two elements can be discussed. Figure 18 presents the relationship between the *Environment* and the *Explicit Environmental Factors* (Weiss and Schmidt, 2010).

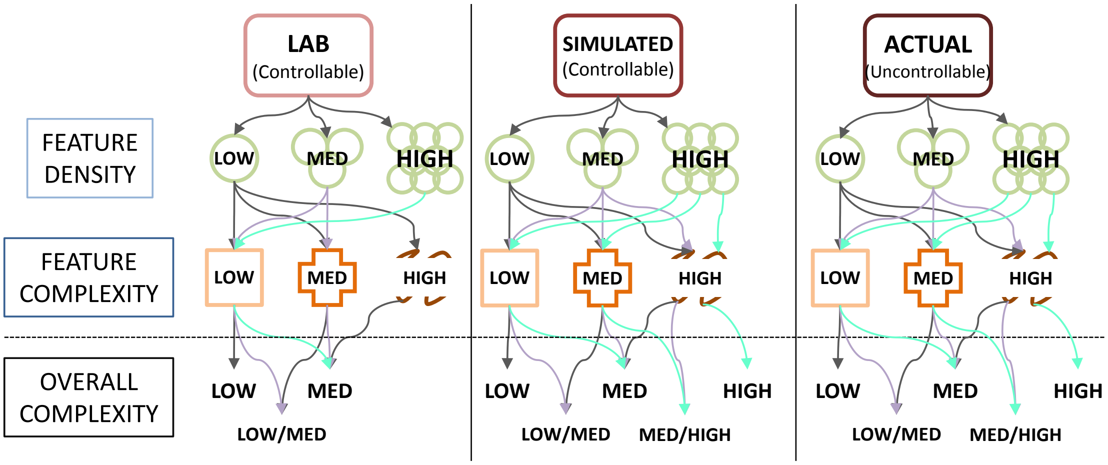


Figure 18: Relationships between the Environment and Explicit Enviornmental Factors (Weiss and Schmidt, 2010)

Note that "Low" and "High" *Overall Complexities* are achieved by single combinations of "Low" "Low" and "High" "High" *Feature Densities* and *Feature Complexities*, respectively. "Low/Medium" and "Medium/High" *Overall Complexity* can be obtained by two combinations of *Feature Density* and *Feature Complexity*."Medium" *Overall Complexity* is achieved by three unique combinations. Since the *Lab* environment is heavily controlled by the *Evaluators* and it's usually desired to obtain specific *Technical Performance* data during the technology's early stages of development, it's unlikely that the *Overall Complexity* will exceed the "Medium" level. It is possible to obtain *Utility Assessment* data in the Lab, but this *Environment* limits the type and range of qualitative data that can be captured since the *Lab* is not indicative of the *Actual Environment*. The *Simulated* and *Actual Environments* are capable of producing the full range of *Overall Complexities* where the significant difference between the two is that the *Evaluation Designer* has some measure of control over the parameters and variables present within the *Simulated Environment.* However, the *Evaluation Designer* has no control over test parameters and variables within the *Actual Environment*.

4.4.11. Tools

MRED outputs the available *Tools* to capture data for the output *Metrics* of each blueprint. The output *Tools* are no different than the input *Tools*. The only exception is that the output *Tools* are a subset of the *Tools* input into MRED.

4.5. *Input Relationships*

MRED exploits the relevant relationships that exist among the various inputs. 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 4.3. This section will present these specific relationships. Since the *MRED Operator* actively defines these relationships as they step through the MRED process, the robot arm's corresponding relationships will be presented in Section 4.6.

4.5.1. Components and Capabilities

The relationship between *Components* and *Capabilities* is the influence each *Component* has on performing or realizing the *Capabilities* within the *System*. It is defined in a single binary matrix (detailed in Section 4.6.1). The *Components – Capabilities* relationship is critical where MRED identifies those Capabilities and the System that are not Fully Developed. If the *Capability* and *System Maturities* are not provided by the *Technology Developer* or *Evaluation Designer*, then MRED uses the *Component Maturity* information to determine whether or not a *Capability* is Fully Developed. MRED calculates whether or not the *System* is Fully Developed based upon the *Capability Maturity* estimates.

4.5.2. Metrics and Technology Test Levels

The relationship between *Metrics* and *TTLs* is defined in two binary matrices and indicates which *Metrics* are applicable to each *TTL*. The first matrix represents which *Technical Performance Metrics* can be produced when testing the *TTLs*. The second matrix represents which *Quantitative Assessment Metrics* can be produced when testing the *Capabilities* and the *System*. The matrices are detailed in Section 4.6.1. 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 during certain points of the MRED process (discussed further in Section 4.6).

4.5.3. Technology Test Levels and Environments

The relationship between *TTLs* and *Environments* indicates which of the available *Environments* each of the *TTLs* can be evaluated within. It is defined in three binary matrices (detailed in Section 4.6). The first matrix represents which *Components* and *Capabilities* can be evaluated within the *Lab Environments*; the second matrix represents which *TTLs* (among all three types) can be evaluated within the *Simulated Environments*; and the third matrix indicates which *Capabilities* and the *System* can be evaluated within the *Actual Environments*. If there are no candidate *Environments* available to test a specific *TTL*, then MRED eliminates this *TTL* from further testing consideration.

4.5.4. Metrics and Tools

The relationship between *Metrics* and *Tools* is defined in two binary matrices: 1) Technical Performance Metrics – Tools and 2) Utility Assessment Metrics – Tools. The first relationship 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* (presented in Section 4.6). The benefit of these relationships 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).

4.6. MRED Process

The specific MRED process is detailed in this section, highlighted in Figure 19, and governed by a set of constraints. These constraints are presented in Table 14. Table 14 also lists the MRED Operator’s responsibilities which implies their authority throughout the blueprint generation process. These responsibilities highlight the interaction between the MRED Operator and the MRED process.

Table 14 and derived from the relationships among the blueprint elements. The process takes the inputs, defined in Section 4.3, and generates one or more blueprints with outputs, defined in Section 4.4, through the systematic application of the pertinent relationships identified in Sections 4.4 and 4.5. The robot arm example is continued to illustrate the process.

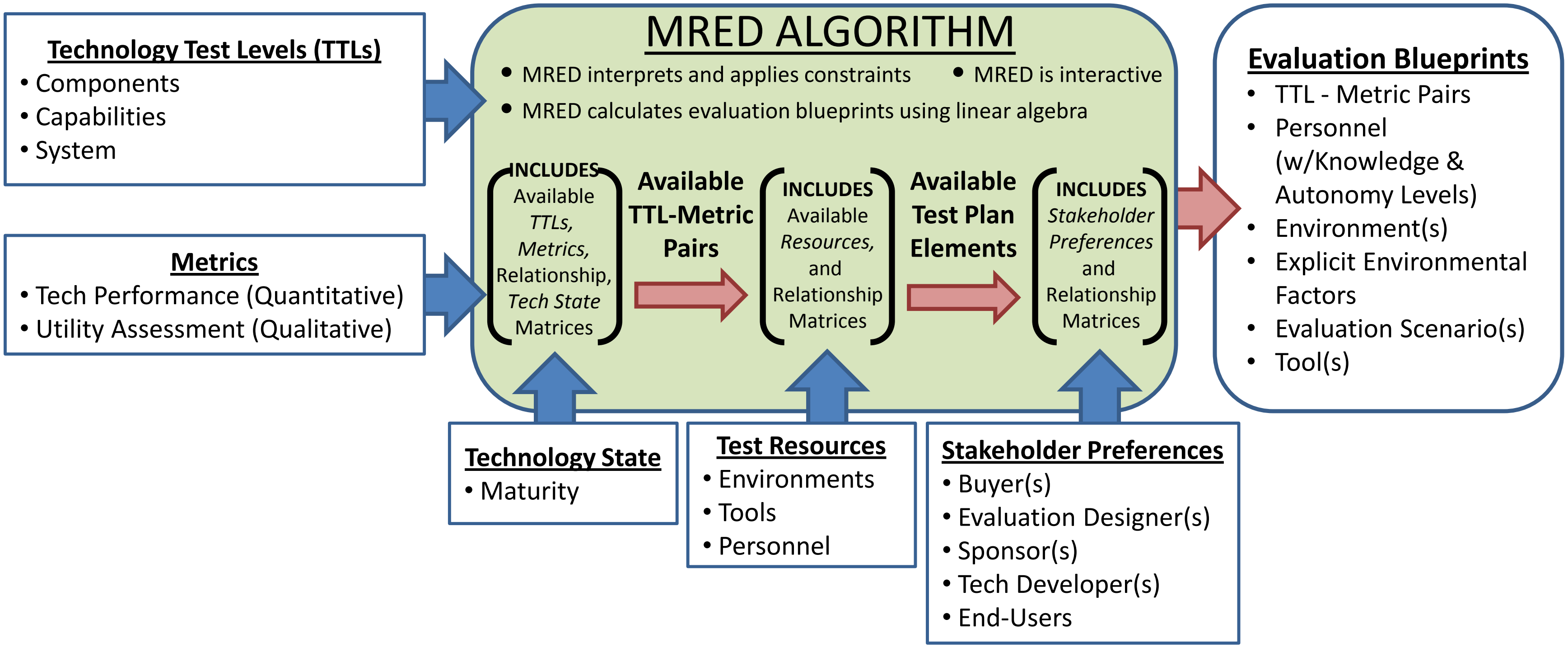

Technology Test Levels (TTLs)
• Components
• Capabilities
• System
Metrics
• Tech Performance (Quantitative)
• Utility Assessment (Qualitative)
MRED ALGORITHM
• MRED interprets and applies constraints
• MRED is interactive
• MRED calculates evaluation blueprints using linear algebra
INCLUDES Available *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
Test Resources
• Environments
• Tools
• Personnel
Stakeholder Preferences
• Buyer(s)
• Evaluation Designer(s)
• Sponsor(s)
• Tech Developer(s)
• End-Users

Figure 19: MRED Process and Algorithm

Table 14 also lists the MRED Operator's responsibilities which implies their authority throughout the blueprint generation process. These responsibilities highlight the interaction between the MRED Operator and the MRED process.

Table 14 - Constraints Governing MRED and MRED Operator Authority

#	CONSTRAINTS
1	One or more blueprints may be produced from a set of inputs.
2	If a Stakeholder overlaps two or more Stakeholder types, then their preferences are only counted once.
3	If all five types of Stakeholders are not present, stakeholder preferences may be captured from less than five stakeholder types.
4	A blueprint may contain one or more TTL-Metric pairs.
5	At least one group of Technology-Users is required for each set of blueprints.
6	End-Users should not be the Technology-Users for bleuprints evaluating the Component TTL.
7	Technology Developers should not be the Technology Users for for blueprints capturing Utility Assessment Metrics.
8	Team Members and/or Participants may be optional for each set of blueprints and are at the discretion of the MRED Operator.
9	A personnel group's DM Autonomy - Technical should either be less than or equal to this group's level of technical knowledge.
10	A personnel group's DM Autonomy - Enviromental should either be less than or equal to this group's level of operational knowledge.
11	Technology-based scenarios should not be implemented in the Actual Environment.
12	Environment-based scenarios should not be implemented in the Lab Environment.
13	Technical Knowledge of the Technology Users should either be Medium or High when paired with Technology-based scenarios.
14	Operational Knowledge of the Technology Users should either be Low, Medium, or High when paired with Technology-based scenarios.
15	DM Autonomy levels of the Technology Users should either be None or Low when paired with Technology-based scenarios.
16	Knowledge and DM Autonomy levels of the Technology Users should either be Low, Medium, or High when paired with Task/Activity-based scenarios.
17	Knowledge and DM Autonomy levels of the Technology Users should either be Medium or High when paired with Environment-based scenarios.
18	If Feature Density is Medium, then Feature Complexity cannot exceed Medium in the Lab Environment.
19	If Feature Density is High, then Feature Complexity must be Low in the Lab Environment.
#	**MRED OPERATOR RESPONSIBILITIES**
1	Defines the TTLs and Metrics for test consideration with input from stakeholders, as necessary.
2	Defines the available Resources with input from stakeholders, as necessary.
3	Defines relationships among the TTLs, Metrics, and Resources with input from stakeholders, as necessary.
4	Inputs Technology State data with input from Technology Developers
5	Determines which Metrics cannot be captured from Immature TTLs, with input from the technology developers.
6	Defines threshold to eliminate TTL-Metric Pairs after the Stakeholder Preferences are processed.
7	Groups TTL-Metric pairs, as appropriate.
8	Determines if one or more blueprint elements (non-TTL-Metric pairs) may be accepted based upon Stakeholder Preferences.

4.6.1. TTLs, Metrics, and Relationships

MRED begins with the *MRED Operator* inputting the available *TTLs* and corresponding *Metrics* (both *Technical Performance* and *Utility Assessment*). Pseudocode for this section includes:

```
Input Component names
Input Capability names
Input SystemPresence equal to one

Input Technical Performance Metric names
Input Utility Assessment names
```

Figure 20: Segment of Pseudocode for TTLs, Metrics, and Relationships

The complete pseudocode can be found in Appendix B: Pseudocode. The sets of τ *Components* (**c**), φ *Capabilities* (**p**) and the *System* (**s**) are defined as:

$$\mathbf{c} = \{\mathrm{c}_1, \mathrm{c}_2, \dots \mathrm{c}_\tau\} \tag{1}$$

$$\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \dots \mathbf{p}_\varphi\} \tag{2}$$

$$\mathbf{s} = \mathbf{1} \tag{3}$$

The sets of α *Technical Performance Metrics* and β *Utility Assessment Metrics* are expressed as:

$$\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, \dots \mathbf{t}_\alpha\} \tag{4}$$

$$\mathrm{a} = \{\mathrm{a}_1, \mathrm{a}_2, \dots \mathrm{a}_\beta\} \tag{5}$$

Table 15 applies these definitions to the robot arm example.

Table 15 - *TTLs* and *Metrics* defined for Robot Arm

c =	Rev 1 (C_1)	Rev 2 (C_2)	Pris 1 (C_3)	Rev 3 (C_4)	Pris 2 (C_5)	Rev 4 (C_6)	Gripper (C_7)	
τ = 7								
p =	X (P_1)	Y (P_2)	Z (P_3)	Roll (P_4)	Pitch (P_5)	Yaw (P_6)	Grasp (P_7)	
ϕ = 7								
t =	Maximum Force	Maximum Linear Velocity	Maximum Torque	Maximum Angular Velocity	Range of Motion	Maximum Lift Capacity	Speed	Force
α = 8								
a =	Responsiveness		Smoothness		Operator Satisfaction			
β = 3								

Next, the *MRED Operator* defines two sets of relationships; the *Components – Capabilities* relationship matrix and the *Metrics – TTLs* relationship matrices (discussed in Section 4.5). The *Components – Capabilities* relationship matrix, *O*, is defined:

$$O = \begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_\tau \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & \cdots & p_\varphi \end{array} \\ \begin{bmatrix} o_{11} & o_{12} & \cdots & o_{1\varphi} \\ o_{21} & o_{22} & \cdots & o_{2\varphi} \\ \vdots & \vdots & \vdots & \vdots \\ o_{\tau 1} & o_{\tau 2} & \cdots & o_{\tau\varphi} \end{bmatrix} \end{array} \qquad (6)$$

Values of *O* are either 0 or 1 where a 1 indicates that a specific *Component* influences the function of a specific *Capability* while a 0 indicates no such relationship exists. Table 16 presents the corresponding O matrix for the robotic arm example.

Table 16 - *O* Relationship Matrix for Robot 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**

Two *Metrics – TTL* binary relationship matrices are defined. *U_1* indicates which of the quantitative *Technical Performance* metrics can be measured from each type of *TTL*. *U_2* indicates which of the qualitative *Utility Assessment* metrics can be measured from the *Capabilities* and the *System*. Table 17 and Table 18 present the *U_1* and *U_2* matrices, respectively.

$$U_1 = \begin{array}{cccccccc} & c_1 & \cdots & c_\tau & p_1 & \cdots & p_\varphi & s \\ t_1 & u_{1_{1,1}} & \cdots & u_{1_{1,\tau}} & u_{1_{1,\tau+1}} & \cdots & u_{1_{1,\tau+\varphi}} & u_{1_{1,\tau+\varphi+1}} \\ t_2 & u_{1_{2,1}} & \cdots & u_{1_{2,\tau}} & u_{1_{2,\tau+1}} & \cdots & u_{1_{2,\tau+\varphi}} & u_{1_{2,\tau+\varphi+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ t_\alpha & u_{1_{\alpha,1}} & \cdots & u_{1_{\alpha,\tau}} & u_{1_{\alpha,\tau+1}} & \cdots & u_{1_{\alpha,\tau+\varphi}} & u_{1_{\alpha,\tau+\varphi+1}} \end{array} \tag{7}$$

$$U_2 = \begin{array}{ccccc} & p_1 & \cdots & p_\varphi & s \\ a_1 & u_{1_{1,1}} & \cdots & u_{1_{1,\varphi}} & u_{1_{1,\varphi+1}} \\ a_2 & u_{1_{2,1}} & \cdots & u_{1_{2,\varphi}} & u_{1_{2,\varphi+1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_\beta & u_{1_{\beta,1}} & \cdots & u_{1_{\beta,\varphi}} & u_{1_{\beta,\varphi+1}} \end{array} \tag{8}$$

Table 17 - U_1 Relationship Matrix for Robot Arm

		Technology Test Levels (TTLs)														
		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)
Metrics - Technical Performance	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

Table 18 – U_2 Relationship Matrix for Robot Arm

		Technology Test Levels (TTLs)							
		P_1	P_2	P_3	P_4	P_5	P_6	P_7	System (S)
Metrics - Utility Assessment	Responsiveness	1	1	1	0	0	0	1	1
	Smoothness	1	1	1	1	1	1	1	1
	Operator Satisfaction	1	1	1	1	1	1	1	1

4.6.2. Constraint Handling and Candidate Rejection

The next phase of the MRED process includes the inputs of *Technology State* and *Resources* (*Environments, Tools, Personnel*). These inputs and the relationships described in Sections 4.5.3 and 4.5.4 are captured in matrices so MRED can determine candidate output elements. These steps are described in the following subsections (Weiss and Schmidt, 2012) and the overall process is shown in Figure 22. Pseudocode of these steps is shown throughout this section where the complete pseudocode can be found in Appendix B: Pseudocode.

4.6.2.1. Technology State

The *MRED Operator* now inputs the *Technology State* information (*Maturity*) for the *Components* (step A of Box 1 in Figure 22). This step also includes inputting the *Technology State* information for the *Capabilities* and the *System*, if explicitly known. *Maturity* (**m**) is defined in three vectors: $\mathbf{m_1}$ corresponds to the *Maturity* of the τ *Components*, $\mathbf{m_2}$ corresponds to the *Maturity* of the φ *Capabilities* and $\mathbf{m_3}$ for the *System*. Values for these vectors input by the *MRED Operator* are either 1 (Fully-Developed) or 0 (Immature).

When *Maturity* is unknown for the *Capabilities* and *System* ($\mathbf{m_2}$, $\mathbf{m_3}$), MRED calculates these vectors (B of Box 1 in Figure 22). The Maturity for the Capabilities is presented in the normalized equation (9).

$$\mathbf{m_2}_j = \mathbf{m_1}\mathrm{Col}_i(O) \,/\, \textstyle\sum_{i=1}^{\tau} o_{i,j} \tag{9}$$

Like $\mathbf{m_1}$, values of $\mathbf{m_2}$ will range from 0 to 1. A *Maturity* equivalent to 1 indicates that a *Capability* is Fully Developed. A *Maturity* less than 1 indicates a *Capability* is Immature. *Maturities* less than 1 are used by MRED to alert the MRED Operator that it may not be possible to capture one or more corresponding metrics. The meaning of this number is that it signifies the number of contributing components that are immature out of the total number of contributing components as a percentage.

MRED estimates the *Maturity* of the *System* as the average of the individual *Capabilities' Maturities*. Similar to *Capability Maturity,* the *System Maturity* value is used to indicate whether the *System* is Fully Developed (*Maturity* equal to 1) or Immature (*Maturity* less than 1). This is presented for *Maturity* in equation (10).

$$\mathbf{m_3} = \left(\sum_{i=1}^{\varphi} \mathbf{m_2}_i\right)/\varphi \tag{10}$$

Next, MRED alerts the MRED Operator which *TTLs* are Immature (C of Box 1 in Figure 22. The MRED Operator removes the relationships between those *Metrics* and *TTLs* in U_1 and U_2 if a *TTL's* immaturity does not allow the corresponding *Metric* to be captured (D of Box 1 in Figure 22). Pseudo code for steps A through D of Box 1 in Figure 22 is presented in Figure 21.

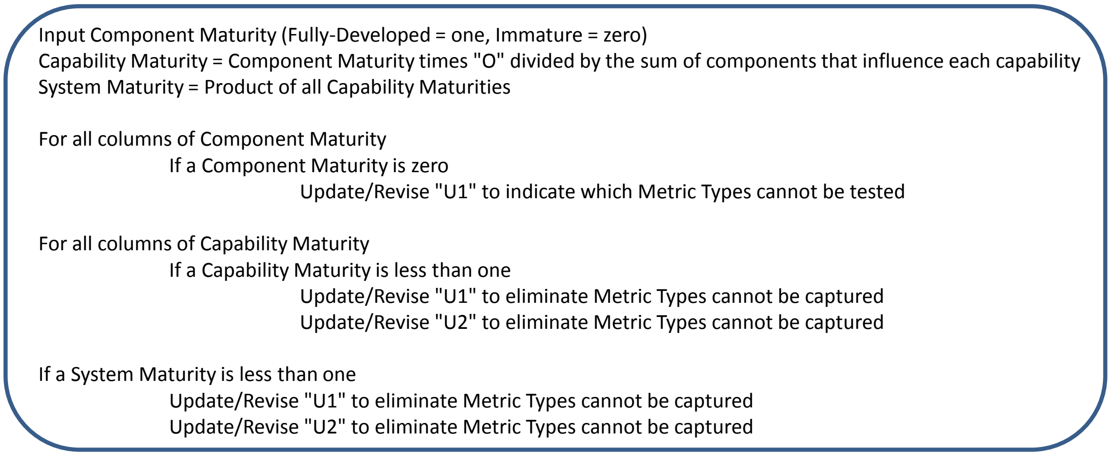
4.6.2.2. Constraint-Handling Process

Rejecting candidate *TTLs*, or any other blueprint element, is a non-trivial process that requires several steps. One way to characterize this process is as the elimination of elements due to constraints. It's a process that will be repeated several times. This process is composed of the following steps:

- INPUT (*Element*) – The MRED Operator inputs the stated information into the MRED algorithm.
- DEFINE Matrix (*Element1* & *Element2*) – The MRED Operator defines of various relationships among blueprint elements, those of which that are outlined in Section 4.5. DEFINE *X* (*TTLs – Env*) means that the *X* matrices are defined relating *TTLs* to the candidate *Environments* (*X* is defined in the following section).
- ELIMINATE (*Element*) – This step requires the removal of specific blueprint elements from their respective sets. For example, *ELIMINATE* (*TTLs*) would involve removing specific *Components* from **c**, *Capabilities* from **p**, and updating **s** to either be 0 or remain 1. This step involves decrementing the appropriate counters when blueprint elements are eliminated.
- FILTER Matrix (*Element*) – This step involves removing either the rows or columns corresponding to the indicated *Element* within the noted relationship matrix. A row or column within a matrix is removed for one of the reasons listed below:
    - The corresponding *Element* was removed as a candidate during the preceding elimination step.
    - The corresponding *Element* no longer has any relationships with its counterpart *Element* in the relationship matrix(ices) which is indicated by the sum of the row or column being equal to 0.

FILTER *U* (*TTLs*) means that those columns within the *U* matrices that correspond to eliminated *TTLs* or that have no available *Metrics* for measurement are removed. The only exception to this notation is FILTER *O* which calls for the removal of rows and/or columns corresponding to eliminated *Components* and/or *Capabilities*.

Figure 22 presents MRED's constraint handling and element filtration process as the *Technology State* and available *Resources* (*Environments*, *Tools*, and *Personnel*) are input. Since the *Maturity* has been defined for all *TTLs* at this point, the steps (D. through I.) in box 1 (Figure 22) are executed.



```
Input Component Maturity (Fully-Developed = one, Immature = zero)
Capability Maturity = Component Maturity times "O" divided by the sum of components that influence each capability
System Maturity = Product of all Capability Maturities

For all columns of Component Maturity
        If a Component Maturity is zero
                Update/Revise "U1" to indicate which Metric Types cannot be tested

For all columns of Capability Maturity
        If a Capability Maturity is less than one
                Update/Revise "U1" to eliminate Metric Types cannot be captured
                Update/Revise "U2" to eliminate Metric Types cannot be captured

If a System Maturity is less than one
        Update/Revise "U1" to eliminate Metric Types cannot be captured
        Update/Revise "U2" to eliminate Metric Types cannot be captured
```

Figure 21: Pseudocode corresponding to A, B, C, & D of Box 1 in

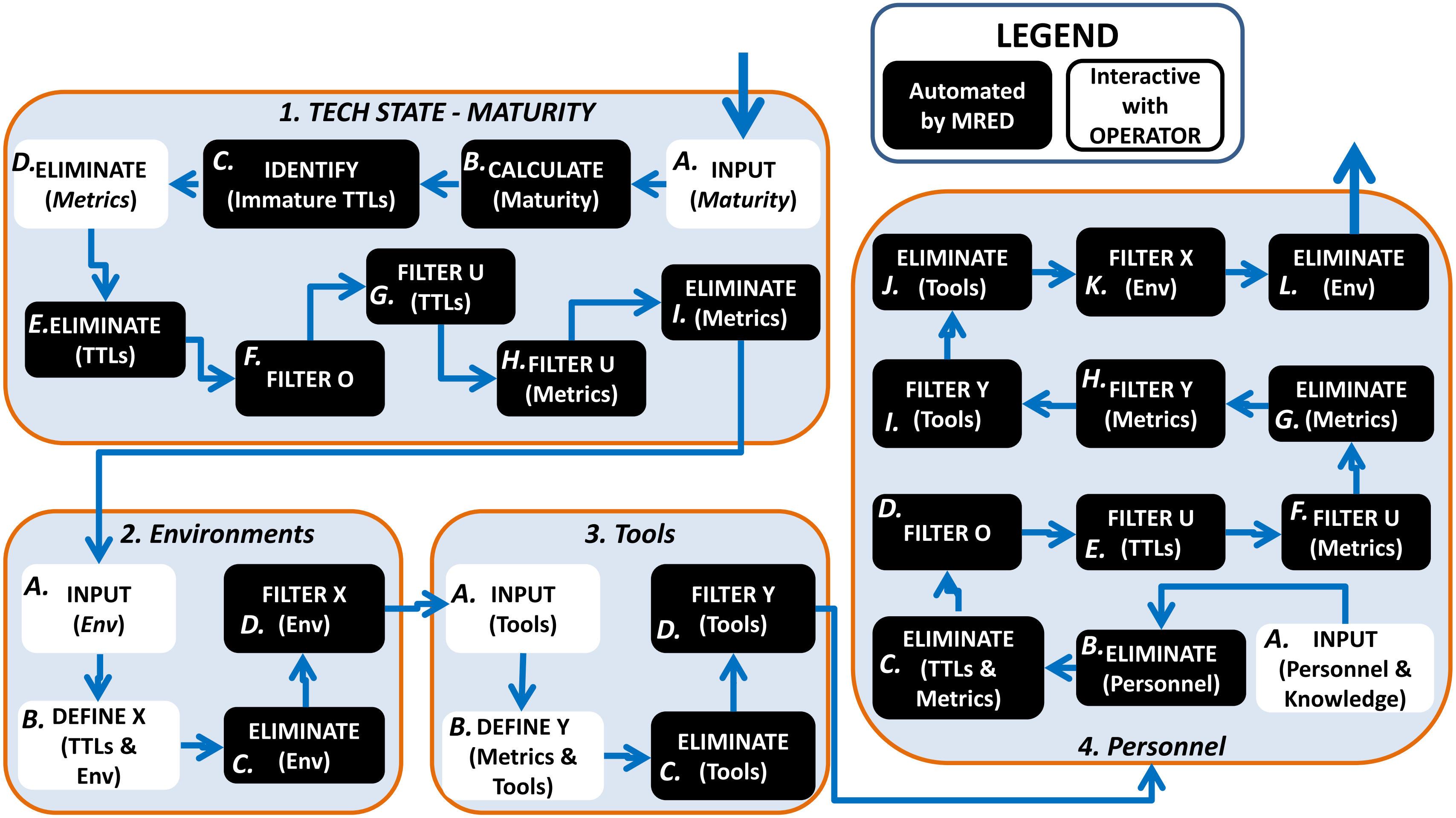


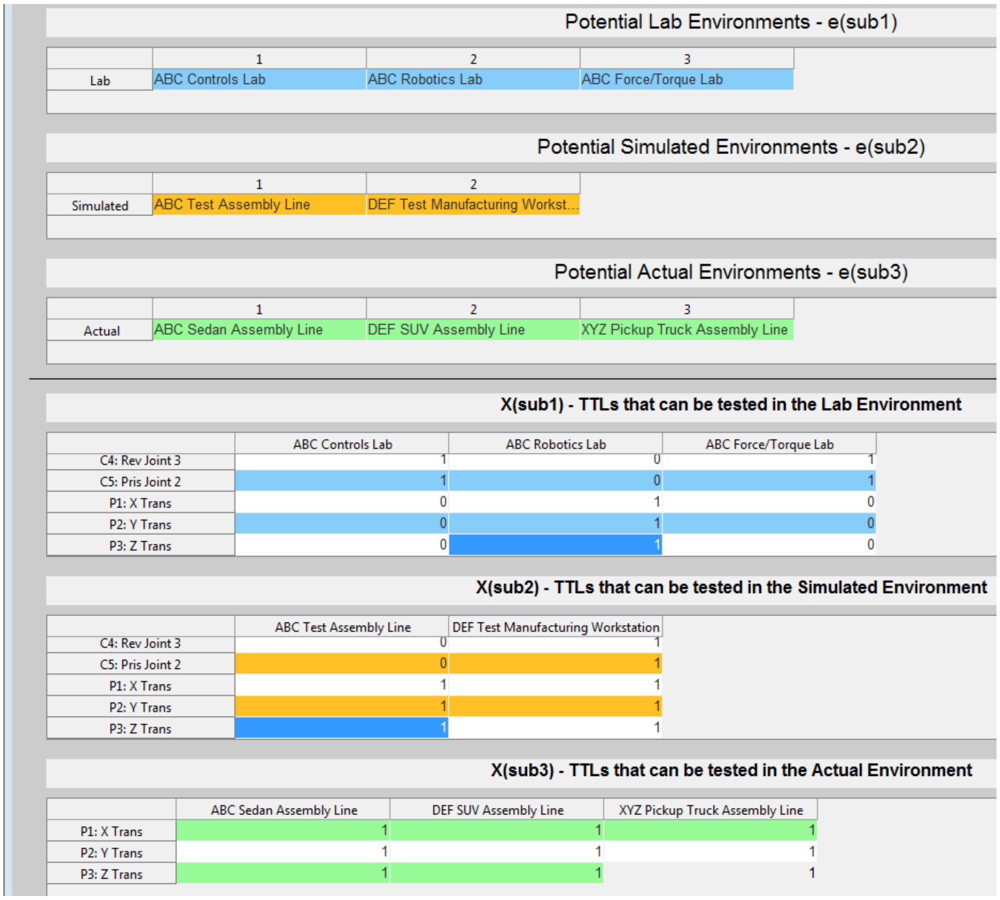
Figure 22: MRED Constraint Handling and Element Filtration Process

4.6.2.3. *Environments*

The process outlined in Figure 22 continues into box 2. The *MRED Operator* inputs the three types of candidate *Environments* that are available for evaluation. Specifically, the *MRED Operator* notes the γ *Lab Environments* (e_1), the δ *Simulated Environments* (e_2), and the ε *Actual Environments* (e_3). Now that the *Environments* and their counters are input, the three specific steps (A. DEFINE, B. ELIMINATE, C. FILTER) in box 2 are engaged. Equation (11) presents the X_1 matrix. X_2 and X_3 are defined similar to Equation (11).

$$X_1 = \begin{array}{c} \\ \mathrm{c}_1 \\ \vdots \\ \mathrm{c}_\tau \\ \mathrm{p}_1 \\ \vdots \\ \mathrm{p}_\varphi \end{array} \begin{array}{c} \begin{array}{cccc} \mathrm{e}_{1_1} & \mathrm{e}_{1_2} & \cdots & \mathrm{e}_{1_\gamma} \end{array} \\ \begin{bmatrix} x_{1_{1,1}} & x_{1_{1,2}} & \cdots & x_{1_{1,\gamma}} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1_{\tau',1}} & x_{1_{\tau',1}} & \cdots & x_{1_{\tau\prime,\gamma}} \\ x_{1_{\tau'+1,1}} & x_{1_{\tau'+1,2}} & \cdots & x_{1_{\tau'+1,\gamma}} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1_{\tau+\varphi,1}} & x_{1_{\tau+\varphi,2}} & \cdots & x_{1_{\tau+\varphi,\gamma}} \end{bmatrix} \end{array} \qquad (11)$$

Figure 23 presents candidate *Environments* (e_1, e_2, and e_3) and the *TTL – Environment* relationship matrixes (X_1, X_2, and X_3). Figure 23 also reflects the absence of the *Components, Capability* and the *System* that were eliminated due to *Technology State* factors.



Potential Lab Environments - e(sub1)

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

Potential Simulated Environments - e(sub2)

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

Potential Actual Environments - e(sub3)

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

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

	ABC Controls Lab	ABC Robotics Lab	ABC Force/Torque Lab
C4: Rev Joint 3	1	0	1
C5: Pris Joint 2	1	0	1
P1: X Trans	0	1	0
P2: Y Trans	0	1	0
P3: Z Trans	0	1	0

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

	ABC Test Assembly Line	DEF Test Manufacturing Workstation
C4: Rev Joint 3	0	1
C5: Pris Joint 2	0	1
P1: X Trans	1	1
P2: Y Trans	1	1
P3: Z Trans	1	1

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	1	1
P2: Y Trans	1	1	1
P3: Z Trans	1	1	1

Figure 23: Example Interface Showing the Potential Test Environments and the *X* relationship matrices

Once the remaining steps are completed in box 2 of Figure 22, it's time to input and refine the available *Tools*.

4.6.2.4. *Tools*

A process occurs for the *Tools* (shown in box 3 of Figure 22) similar to what was just illustrated for *Environments*. The *MRED Operator* inputs the *Tools* that are available for evaluation in sets d_1 (corresponding to the ζ tools available to support *Technical Performance Metrics*) and d_2 (corresponding to the η *Utility Assessment Metrics*). Now that these inputs are in place, the three step candidate elimination process begins by defining the *Y* relationship matrices between *Metrics* and the available *Tools* (that support the measurement of these *Metrics*). Y_1 is presented in equation (12). Y_2 is defined similar to Equation (12).

$$Y_1 = \begin{array}{c c} & \begin{array}{cccc} d_{1_1} & d_{1_2} & \cdots & d_{1_\zeta} \end{array} \\ \begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_\alpha \end{array} & \begin{bmatrix} y_{1_{1,1}} & y_{1_{1,2}} & \cdots & y_{1_{1,\zeta}} \\ y_{1_{2,1}} & y_{1_{2,2}} & \cdots & y_{1_{2,\zeta}} \\ \vdots & \vdots & \vdots & \vdots \\ y_{1_{\alpha,1}} & y_{1_{\alpha,2}} & \cdots & y_{1_{\alpha,\zeta}} \end{bmatrix} \end{array} \tag{12}$$

Once the steps are complete in box 3 of Figure 22, it is time to input the available *Personnel*. This leads to further eliminating and filtering of the remaining candidate blueprint elements.

4.6.2.5. Personnel

The *MRED Operator* inputs the available *Personnel* and their greatest *Technical* and *Operational Knowledge* levels before moving to the first elimination step (A) in box 4 of Figure 19. Input *Personnel* are captured in the matrix *N* defined in equation (13).

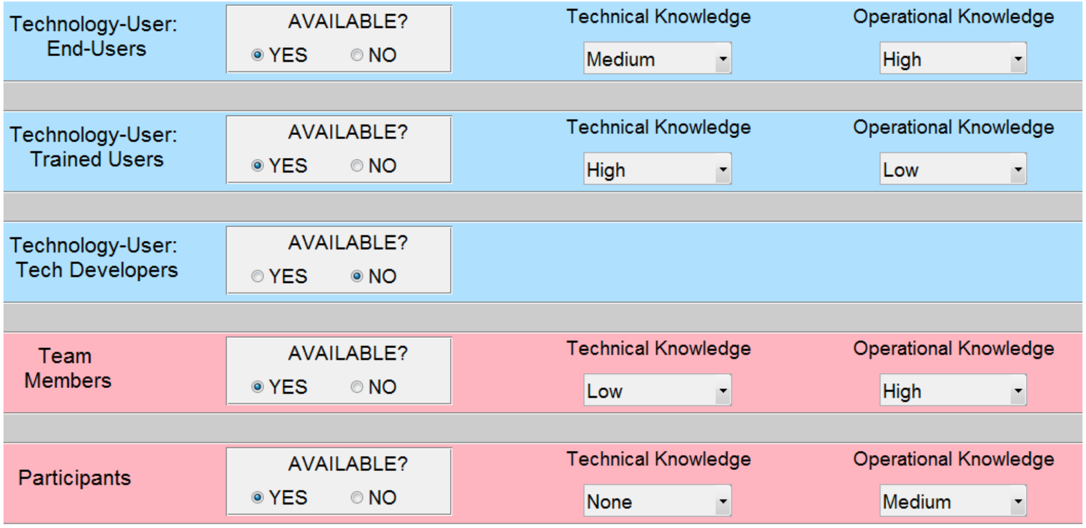
$$N = \begin{bmatrix} n_{1,1} & n_{1,2} & n_{1,3} \\ n_{2,1} & n_{2,2} & n_{2,3} \\ n_{3,1} & n_{3,2} & n_{3,3} \\ n_{4,1} & n_{4,2} & n_{4,3} \\ n_{5,1} & n_{5,2} & n_{5,3} \end{bmatrix} \tag{13}$$

where…

- $\text{Row}_1(N)$ corresponds to *Tech-Users: End-Users*
- $\text{Row}_2(N)$ corresponds to *Tech-Users: Trained Users*
- $\text{Row}_3(N)$ corresponds to *Tech-Users: Tech Developers*

- Row$_4$(N) corresponds to *Team Members*
- Row$_5$(N) corresponds to *Participants*
- Col$_1$(N) corresponds to presence of personnel (0 – Unavailable, 1 – Available)
- Col$_2$(N) corresponds to the greatest level of *Technical Knowledge* required from at least one of the *Personnel* types (0 – None, 1 – Low, 2 – Medium, 3 - High)
- Col$_3$(N) corresponds to the greatest level of *Operational Knowledge* required from at least one of the *Personel* types (0 – None, 1 – Low, 2 – Medium, 3 - High)

Figure 24 shows an example interface that the *MRED Operator* inputs the available *Personnel* and their corresponding *Technical* and *Operational Knowledge Levels*. Matlab automatically saves this data in the *N* matrix format noted above.

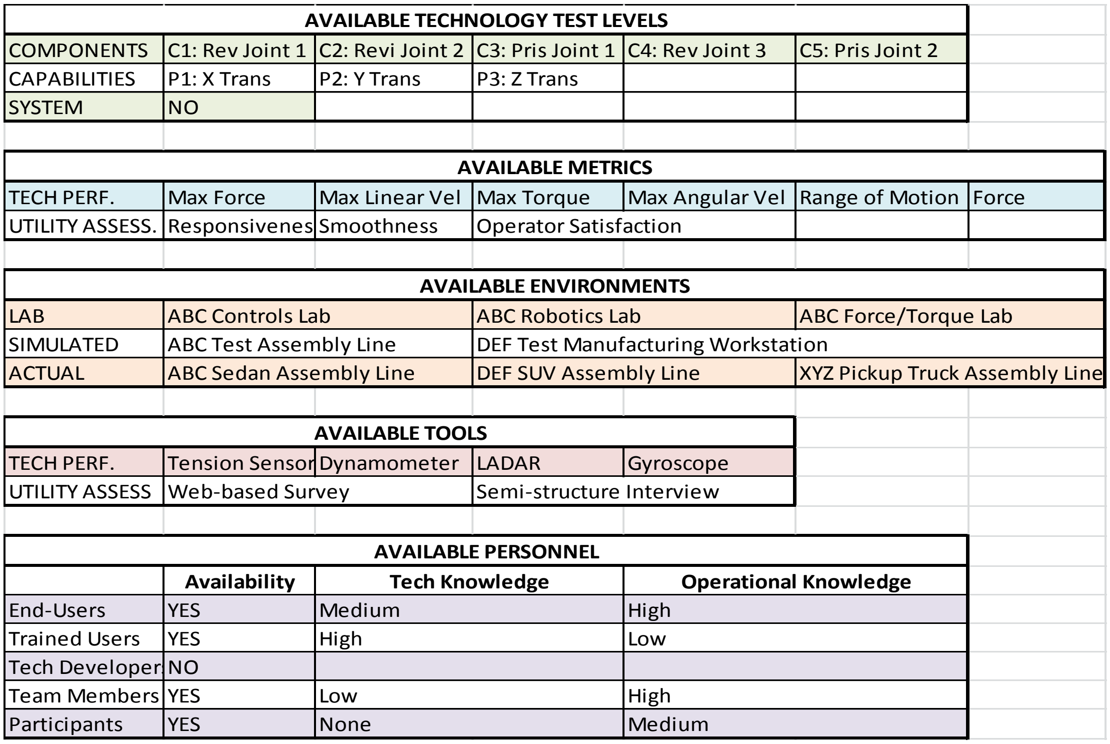


	AVAILABLE?	Technical Knowledge	Operational Knowledge
Technology-User: End-Users	● YES ○ NO	Medium	High
Technology-User: Trained Users	● YES ○ NO	High	Low
Technology-User: Tech Developers	○ YES ● NO		
Team Members	● YES ○ NO	Low	High
Participants	● YES ○ NO	None	Medium

Figure 24: Matlab Interface Showing the Available *Personnel* and their greatest Knowledge Levels

Elimination of *TTLs* and *Metrics* at the next step (B in Box 4 of Figure 22) not only addresses those constraints imposed by the *Personnel* (refer back to Table 11), it also eliminates those *TTLs* and/or *Metrics* that are no longer needed based upon the *Environment(s)* and/or *Tool(s)* that were eliminated in the preceding steps. This creates a domino effect causing further steps to occur. This process concludes at the upper right corner of the box 4 within Figure 22. The remaining feasible candidate test plan elements are presented in Table 19. The next step is to handle *Stakeholder Preferences*. This is discussed in the following section.

Table 19 – Remaining Candidate Test Plan Elements



AVAILABLE TECHNOLOGY TEST LEVELS

COMPONENTS	C1: Rev Joint 1	C2: Revi Joint 2	C3: Pris Joint 1	C4: Rev Joint 3	C5: Pris Joint 2
CAPABILITIES	P1: X Trans	P2: Y Trans	P3: Z Trans		
SYSTEM	NO				

AVAILABLE METRICS

TECH PERF.	Max Force	Max Linear Vel	Max Torque	Max Angular Vel	Range of Motion	Force
UTILITY ASSESS.	Responsivenes	Smoothness	Operator Satisfaction			

AVAILABLE ENVIRONMENTS

LAB	ABC Controls Lab	ABC Robotics Lab	ABC Force/Torque Lab
SIMULATED	ABC Test Assembly Line	DEF Test Manufacturing Workstation	
ACTUAL	ABC Sedan Assembly Line	DEF SUV Assembly Line	XYZ Pickup Truck Assembly Line

AVAILABLE TOOLS

TECH PERF.	Tension Sensor	Dynamometer	LADAR	Gyroscope
UTILITY ASSESS	Web-based Survey		Semi-structure Interview	

AVAILABLE PERSONNEL

	Availability	Tech Knowledge	Operational Knowledge
End-Users	YES	Medium	High
Trained Users	YES	High	Low
Tech Developer	NO		
Team Members	YES	Low	High
Participants	YES	None	Medium

4.6.3. Stakeholder Preference Handling

The next phase of MRED is to capture and handle *Stakeholder Preferences*. An ordinal linguistic scale is devised to capture Stakeholder Preferences based upon prior research (Alwin & Krosnick, 1985; National Opinion Research Center, 1982; O'Brien, 1979). Two different methods are considered to aggregate the Stakeholder Preferences for test plan elements: Majority Voting (Balinski & Laraki, 2007a, 2007b) and Evaluative Voting (Hillinger, 2004). Majority Judgment is selected because it aggregates ordinal preferences in a mathematically valid way. However, Evaluative Voting is also selected because it presents greater level of preference detail as compared to Majority Judgment. Evaluative Voting is proposed based upon the principles of cardinal utility. Cardinal Utility was developed as one of the earliest forms of preference assessment yet is not used as prevalently as ordinal utility. The research community includes proponents of cardinal utility as a means to incorporate an interval scale in preference handling (Fleming, 1952; Harsanyi, 1955; Vasiljev, 2008).

4.6.3.1. Preference Capture Scale

Stakeholder preferences are captured on an ordinal, linguistic scale presented in Figure 25. This scale is symmetric where ratings one through five are negative preferences, rating six is neutral, and ratings seven through eleven are positive preferences. Majority Judgment uses the ordinal data captured on this scale to aggregate preferences. Evaluative Voting uses a corresponding numerical scale (shown in Figure 25) which is considered a cardinal scale (Hillinger, 2007). Majority Judgment is discussed in detail in Section 4.6.3.2 while Evaluative Voting is discussed further in Section 4.6.3.3.

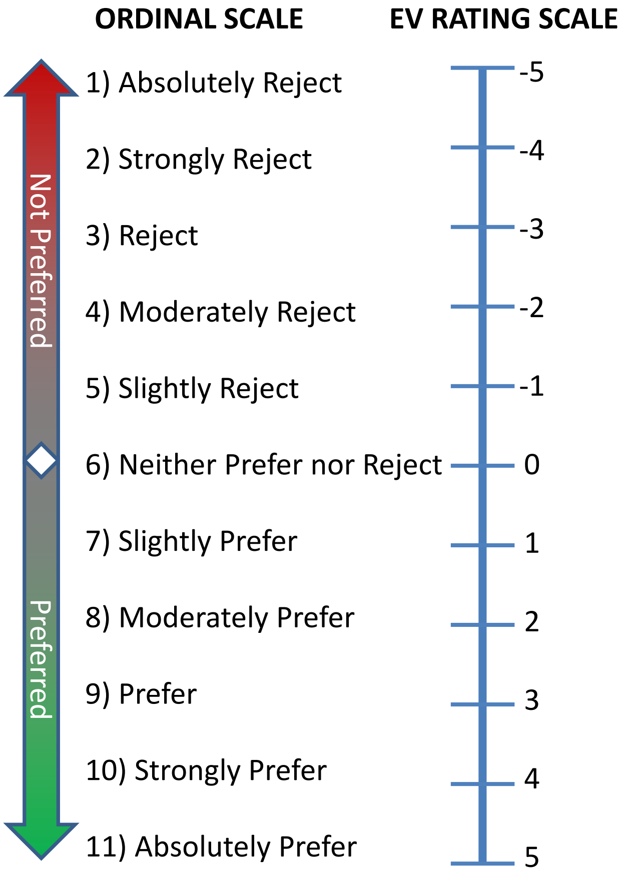


Figure 25: Scales Used for Stakeholder Preference Capture

Stakeholders use this scale to respond to the question of "What is your preference to see this TTL-Metric pair tested or a blueprint element included?"

4.6.3.2. Majority Judgment

Applied to MRED, Majority Voting aggregates the set of stakeholder preferences for a specific blueprint element by identifying the median value of all the stakeholder's preferences. The median is identified under the following conditions (Balinski & Laraki, 2007a):

- The median is the vote in ordered middle of all of the votes, for an odd amount of votes for a specific blueprint element (i.e. median = (n+1)/2 for n total votes)
- The median equals n/2 for n total votes for an even amount of votes

- Ties must be broken for the two or more elements that have the same median value and have achieved the highest ranking. Ties are broken by removing the median value from tying element preference sets. The median is now determined from these new subsets of preferences.
- "No vote" is added to an element's preference set for any stakeholder that chooses to abstain from voting for that specific element. 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.

4.6.3.3. Evaluative Voting

Applied to MRED, Evaluative Voting is a method that calculates the a stakeholder's rating for each blueprint element on an cardinal scale. Figure 25 presents the mapping of a stakeholder's rating to that of a specific value on an 11-point cardinal scale. Numbers -5 to -1 signify a negative preference for an element to be tested, 0 corresponds to a neutral stance to blueprint element being tested, and values 1 to 5 correspond to a positive preference for an element to be tested. A Stakeholder who chooses not to vote will have this lack of preference handled similarly to that of Majority Judgment; "no vote" will be associated with this stakeholder for this blueprint element.

MRED goes beyond Hillinger's original *EV-3* scale to use an 11-point scale originally developed by a German political survey institute when asking survey respondents to rate their satisfaction with politicians (Hillinger, 2004). This scale has a higher resolution, raising the opportunity for each *TTL-Metric* pair to have unique scores further delineating the candidates.

4.6.3.4. Initial Comparison of Majority Judgment to Evaluative Voting

Majority Judgment and Evaluative Voting present similar advantages in their aggregation methods as they are implemented with MRED.

- Capable of providing grades or an order to a set of elements
- Accounts for a *Stakeholder* that chooses not to vote on a specific element in such a manner that does not inflate or deflate an element's score

However, some differences exist between the two methods. Specifically, Majority Judgment exhibits the following behavior:

- Discourages stakeholders from strategically voting
- Emphasizes the middlemost vote

Contrary to Majority Judgment, Evaluative Voting presents the following behavior (Hillinger, 2007):

- 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
- Method is comparable to other judgments expressed on interval scales such as grades (given in schools, universities, etc.) which are often aggregated through averaging

4.6.3.5. MRED's Implementation Preference Capturing and Handling

MRED employs a three-step, iterative process, with one exception (noted below). The three primary steps are QUERY, SCORE, ELIMINATE, with the fourth being GROUP. These steps (highlighted in Figure 26) are defined as:

- QUERY – MRED queries *Stakeholder Preferences* (Absolutely Reject, Strongly Reject,…, Neither Prefer nor Reject,…, Strongly Prefer, Absolutely Prefer) for each available blueprint element (e.g. *TTL-Metric* pair). These are captured in matrices for further use.
- SCORE (MJ) - MRED applies Majority Judgment to produce the grades (or rankings) of each blueprint element.
    - For the case of *TTL-Metric* pairs – ties are acceptable since the intent is to grade these blueprint elements
    - For all other blueprint elements (*Personnel, Environments,* etc.) – ties must be broken for the highest-ranking elements since rankings are desired.
- SCORE (EV) – MRED applies Evaluative Voting by first transforming the linguistic stakeholder votes to the corresponding interval scale presented in Figure 25. Each blueprint element is scored by determining the mean (average) among all of the corresponding stakeholder votes.
- GROUP (*TTL-Metric pairs*, only) – *MRED Operator* groups *TTL-Metric* pairs by *TTLs* or *Metrics*. This is done at the Operator's discretion based upon the specific pairs that score above the set threshold (e.g. > "Neither Prefer nor Reject" or > 0). This step is interactive with the MRED Operator.
- ELIMINATE (for *TTL-Metric* pairs) – MRED eliminates those *TTL-Metric* pairs that score below the pre-determined threshold (and are not grouped with higher-scoring *TTL-Metric* pairs) from further consideration.

- ELIMINATE (applied to all other blueprint elements) - MRED assigns the highest rated blueprint element to the corresponding group of *TTL-Metric* pairs and removes all other candidates from consideration for evaluation with this specific grouping.

MRED’s process of capturing and handling stakeholder preferences is highlighted in Figure 26. This process begins in the upper left box, I. of Figure 26, by determining the preferred *TTL-Metric* pairs.

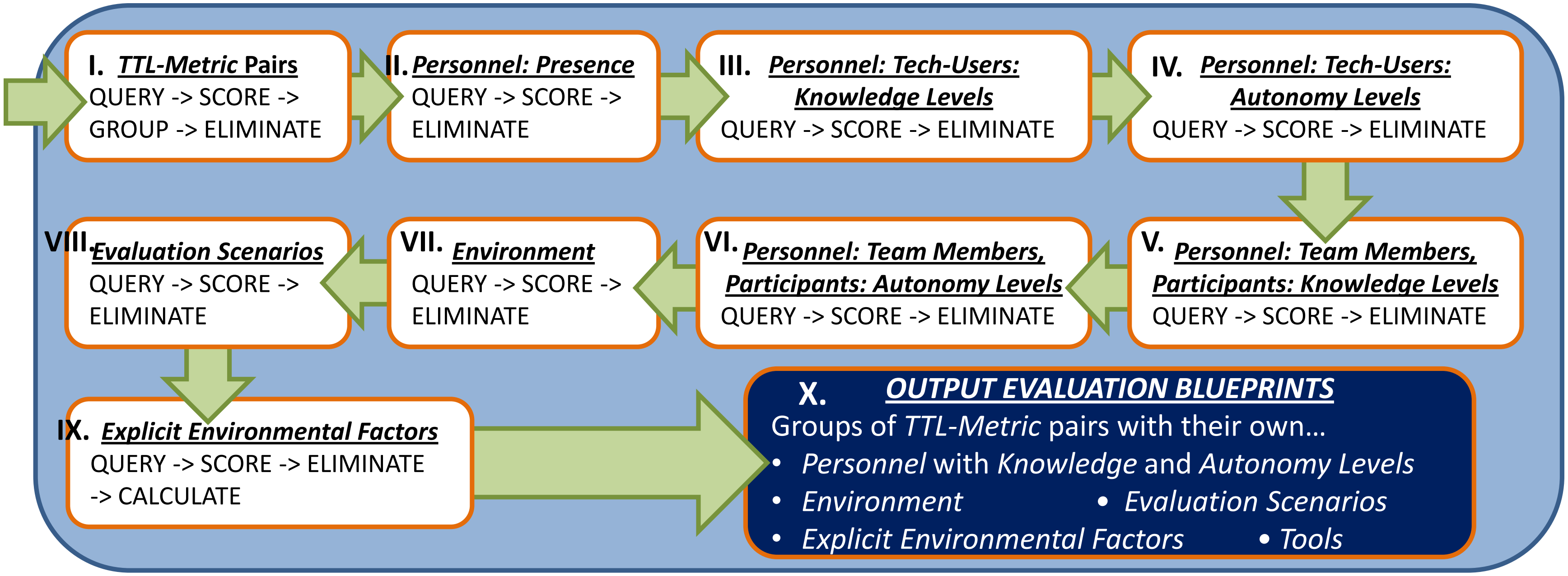

I. TTL-Metric Pairs
QUERY -> SCORE -> GROUP -> ELIMINATE
II. Personnel: Presence
QUERY -> SCORE -> ELIMINATE
III. Personnel: Tech-Users: Knowledge Levels
QUERY -> SCORE -> ELIMINATE
IV. Personnel: Tech-Users: Autonomy Levels
QUERY -> SCORE -> ELIMINATE
V. Personnel: Team Members, Participants: Knowledge Levels
QUERY -> SCORE -> ELIMINATE
VI. Personnel: Team Members, Participants: Autonomy Levels
QUERY -> SCORE -> ELIMINATE
VII. Environment
QUERY -> SCORE -> ELIMINATE
VIII. Evaluation Scenarios
QUERY -> SCORE -> ELIMINATE
IX. Explicit Environmental Factors
QUERY -> SCORE -> ELIMINATE -> CALCULATE
X. OUTPUT EVALUATION BLUEPRINTS
Groups of TTL-Metric pairs with their own…
• Personnel with Knowledge and Autonomy Levels
• Environment
• Evaluation Scenarios
• Explicit Environmental Factors
• Tools

Figure 26: MRED Stakeholder Preference Capture and Handling

Table 20 presents the first step of querying the *Stakeholders* for their specific preferences according to the 11-point linguistic ordinal scale. The dissertation author, acting as the *MRED Operator*, determined the stakeholder preferences in Table 20 based upon reasonable motivations assumed for each *Stakeholder.*

Table 20 - *Stakeholder Preferences* of *TTL-Metric* Pairs

	STAKEHOLDER PREFERENCES				
TTL-Metric Pairs	Buyer	Eval Designer	Sponsor	Tech Dev	User
C_1 - Max Torque	NV	Mod Reject	Slightly Pref	Slightly Rej	NV
C_1 - Max Angular Velocity	NV	Strongly Rej	Neither	Slightly Pref	NV
C_1 - Range of Motion	NV	Slightly Rej	Slightly Pref	Mod Prefer	NV
C_2 - Max Torque	NV	Mod Reject	Slightly Pref	Mod Reject	NV
C_2 - Max Angular Velocity	NV	Strongly Rej	Neither	Slightly Pref	NV
C_2 - Range of Motion	NV	Strongly Rej	Slightly Pref	Mod Prefer	NV
C_3 - Max Force	NV	Strongly Pref	Mod Prefer	Strongly Pref	NV
C_3 - Max Linear Velocity	NV	Strongly Pref	Prefer	Strongly Pref	NV
C_3 - Range of Motion	NV	Abs Prefer	Abs Prefer	Strongly Pref	NV
C_4 - Max Torque	NV	Prefer	Mod Prefer	Abs Prefer	NV
C_4 - Max Angular Velocity	NV	Strongly Pref	Neither	Abs Prefer	NV
C_4 - Range of Motion	NV	Abs Prefer	Abs Prefer	Abs Prefer	NV
C_5 - Max Force	NV	Strongly Pref	Mod Prefer	Abs Prefer	NV
C_5 - Max Linear Velocity	NV	Prefer	Prefer	Abs Prefer	NV
C_5 - Range of Motion	NV	Abs Prefer	Abs Prefer	Abs Prefer	NV
P_1 - Max Force	Mod Prefer	Strongly Pref	Prefer	Prefer	Neither
P_1 - Max Linear Velocity	Slightly Pref	Prefer	Mod Prefer	Prefer	Strongly Pref
P_1 - Range of Motion	Strongly Pref	Prefer	Abs Prefer	Abs Prefer	Abs Prefer
P_1 - Force	Prefer	Slightly Pref	Mod Reject	Prefer	Strongly Pref
P_1 - Responsiveness	Slightly Pref	Prefer	Mod Prefer	Prefer	Strongly Pref
P_1 - Smoothness	Abs Prefer	Mod Prefer	Strongly Pref	Mod Reject	Abs Prefer
P_2 - Max Force	Mod Prefer	Prefer	Prefer	Prefer	Neither
P_2 - Max Linear Velocity	Slightly Pref	Prefer	Mod Prefer	Prefer	Strongly Pref
P_2 - Range of Motion	Strongly Pref	Prefer	Abs Prefer	Abs Prefer	Abs Prefer
P_2 - Force	Prefer	Slightly Pref	Mod Reject	Prefer	Strongly Pref
P_2 - Responsiveness	Abs Prefer	Mod Prefer	Strongly Pref	Mod Reject	Abs Prefer
P_2 - Smoothness	Abs Prefer	Mod Prefer	Strongly Pref	Mod Reject	Abs Prefer
P_3 - Max Force	Abs Prefer	Abs Prefer	Abs Prefer	Strongly Pref	Strongly Pref
P_3 - Max Linear Velocity	Strongly Pref	Strongly Pref	Abs Prefer	Prefer	Strongly Pref
P_3 - Range of Motion	Strongly Pref	Abs Prefer	Abs Prefer	Abs Prefer	Abs Prefer
P_3 - Force	Prefer	Slightly Pref	Mod Reject	Prefer	Strongly Pref
P_3 - Responsiveness	Slightly Pref	Prefer	Mod Prefer	Prefer	Strongly Pref
P_3 - Smoothness	Abs Prefer	Mod Prefer	Strongly Pref	Mod Reject	Abs Prefer

Table 20 shows the *Stakeholder Preferences* while Table 21 presents the average scores and standard deviations of these preferences. In the case of the robot arm, the *MRED Operator* defined the threshold for test consideration to be at 0. This means that any *TTL-Metric* pairs with an average score at or below 0 would be eliminated from further consideration (indicated in Table 21). Standard deviations are shown to present the level of agreement of preference regarding a specific *TTL-Metric* pair. The smaller the deviation on the rating for evaluating a *TTL-Metric*, the stronger the agreement among the *Stakeholders.*

The next step would be to group *TTL-Metric* pairs together to alleviate some of the burden on the *Stakeholders* as they provide their preferences regarding the remaining blueprint elements (*Personnel*, *Environment*, etc.) for each group of *TTL-Metric* pairs. Pairs can either be grouped in three way as follows:

- *TTL* groups (e.g. all of the metrics for P_3 are grouped together so *Stakeholders* only provide a single set of preferences for P_3 – Range of Motion, P_3 – Max Force, etc.),
- *Metric groups* (e.g. all of the *TTLs* required to produce the Range of Motion *Metric* are grouped together), (An exception to grouping by *Metric* would be if the same *Metrics* are to be captured across different types of TTLs, as is the case in this example. Specifically, Range of Motion may be considered an important *Metric* for both *Components* and *Capabilities*.)
- Or a combination of the two methods

This decision is made at the *MRED Operator's* discretion.

Based upon the grouping, the scores, and how expensive it may be to evaluate a specific *TTL* or collect data for a specific *Metric*, the *MRED Operator* may choose to include a *TTL-Metric* pair whose score was below the threshold.

Table 21 – Ratings for *Stakeholder Preferences* for *TTL-Metric* Pairs

EVALUATIVE VOTING				MAJORITY JUDGMENT	
TTL-Metric Pairs	**AVERAGE**	**STD DEV**		**TTL-Metric Pairs**	**MEDIAN**
C_4 - Range of Motion	**5.00**	**0.00**		C_4 - Range of Motion	**Abs Prefer**
C_5 - Range of Motion	**5.00**	**0.00**		C_5 - Range of Motion	**Abs Prefer**
P_3 - Range of Motion	**4.80**	**0.45**		P_3 - Range of Motion	**Abs Prefer**
C_3 - Range of Motion	**4.67**	**0.58**		C_3 - Range of Motion	**Abs Prefer**
P_3 - Max Force	**4.60**	**0.55**		P_3 - Max Force	**Abs Prefer**
P_1 - Range of Motion	**4.40**	**0.89**		P_1 - Range of Motion	**Abs Prefer**
P_2 - Range of Motion	**4.40**	**0.89**		P_2 - Range of Motion	**Abs Prefer**
P_3 - Max Linear Velocity	**4.00**	**0.71**		C3 - Max Linear Velocity	**Strongly Prefer**
C_3 - Max Linear Velocity	**3.67**	**0.58**		C5 - Max Force	**Strongly Prefer**
C_5 - Max Force	**3.67**	**1.53**		C3 - Max Force	**Strongly Prefer**
C_5 - Max Linear Velocity	**3.67**	**1.15**		C4 - Max Angular Velocity	**Strongly Prefer**
C_3 - Max Force	**3.33**	**1.15**		P1 - Smoothness	**Strongly Prefer**
C_4 - Max Torque	**3.33**	**1.53**		P2 - Responsiveness	**Strongly Prefer**
C_4 - Max Angular Velocity	**3.00**	**2.65**		P2 - Smoothness	**Strongly Prefer**
P_1 - Smoothness	**2.80**	**2.95**		P3 - Smoothness	**Strongly Prefer**
P_2 - Responsiveness	**2.80**	**2.95**		P3 - Max Linear Velocity	**Prefer**
P_2 - Smoothness	**2.80**	**2.95**		C5 - Max Linear Velocity	**Prefer**
P_3 - Smoothness	**2.80**	**2.95**		C4 - Max Torque	**Prefer**
P_1 - Max Linear Velocity	**2.60**	**1.14**		P_1 - Max Linear Velocity	**Prefer**
P_1 - Responsiveness	**2.60**	**1.14**		P_1 - Responsiveness	**Prefer**
P_2 - Max Linear Velocity	**2.60**	**1.14**		P_2 - Max Linear Velocity	**Prefer**
P_3 - Responsivenss	**2.60**	**1.14**		P_3 - Responsivenss	**Prefer**
P_1 - Max Force	**2.40**	**1.52**		P_1 - Max Force	**Prefer**
P_2 - Max Force	**2.20**	**1.30**		P_2 - Max Force	**Prefer**
P_1 - Force	**1.80**	**2.39**		P_1 - Force	**Prefer**
P_2 - Force	**1.80**	**2.39**		P_2 - Force	**Prefer**
P_3 - Force	**1.80**	**2.39**		P_3 - Force	**Prefer**
~~C_1 - Range of Motion~~	**0.67**	**1.53**		~~C_1 - Range of Motion~~	~~**Slightly Pref**~~
~~C_2 - Range of Motion~~	**-0.33**	**3.21**	**Negative to Positive**	~~C_2 - Range of Motion~~	~~**Slightly Pref**~~
~~C_1 - Max Torque~~	**-0.67**	**1.53**	**Negative to Neutral**	~~C1 - Max Angular Velocity~~	~~**Neither**~~
~~C_1 - Max Angular Velocity~~	**-1.00**	**2.65**	**Negative to Neutral**	~~C2 - Max Angular Velocity~~	~~**Neither**~~
~~C_2 - Max Torque~~	**-1.00**	**1.73**		~~C1 - Max Torque~~	~~**Slightly Rej**~~
~~C_2 - Max Angular Velocity~~	**-1.00**	**2.65**		~~C2 - Max Torque~~	~~**Mod Reject**~~

The highest rated *TTL-Metric* pairs are consistent for both preference aggregation methods. Several differences are apparent between the Evaluative Voting and Majority Judgment ratings. They include:

- Eleven of the *TTL-Metric* pairs falling on the preferred side of the preferences are rated in different positions (see the green and red arrows in Table 21).
- MJ raises the level of preference as compared to EV for three *TTL-Metric* pairs ranked low; one *TTL-Metric* pair is raised from rejected to preferred while two *TTL-Metric* pairs are raised from rejected to neither reject nor prefer.
- EV is perceived to present greater granulariy in the aggregation of preferences. This perception is conditional upon the acceptance of the validity of the averaging. MJ presents three unique preferred ratings from "Prefer" to "Absolutely Prefer"

Reviewing the data further from Table 21, it's reasonable that the *MRED Operator* could choose to test C_2 – Range of Motion considering that it didn't score much below 0 and Range of Motion Metrics are already being captured for three other *TTLs*. Conversely, an argument can be made not to evaluate C_2 – Range of Motion since there are no C_2 – *Metric* pairs above the 0 threshold for test consideration. Testing this *Component* for a single *Metric* could prove costly and yield little value.

Another situation requiring the MRED Operator's discretion (refer back to Table 14) is if a *TTL – Metric* pair is just above the 0 (neither prefer nor reject) threshold. C_1 – Range of Motion is an example where the *MRED Operator* must use their discretion on eliminating this pair. Although this pair is above the 0 threshold, the *MRED Operator* may choose to eliminate this pair since there are no other *Metrics* being considered for C_1 (i.e. Capturing Range of Motion would be the only *Metric*). In this example, the pair is eliminated. Keeping this *TTL – Metric* available for testing would most likely be an unnecessary cost. The presence of these exceptions influences the grouping decisions by the *MRED Operator* which prevent this from being an automated task within MRED.

MRED provides traceability by capturing and storing all of the *Stakeholders' Preferences* throughout this process. This information can easily be retrieved further into the blueprint development process and beyond, if necessary. This preseveres each *Stakeholder's* individual preference in the event that the *MRED Operator* wanted to review a subset of the *Stakeholder's Preferences* or to apply a weighting factor (discussed further in Section 7.3).

Table 22 presents example groupings of *TTL-Metric* pairs based upon the *Stakeholder Preferences* and scores generated from Majority Judgment and Evaluative Voting. The available *Utility Assessment Metrics* are grouped with several of the *Technical Performance Metrics* for some of the *Capability* testing. This is another judgment by the *MRED Operator* as to what is the most practical and beneficial way to capture these *Utility Assessment Metrics*. Of course, these qualitative *Metrics* could be captured separately from the quantitative *Metrics*, yet this would be an additional cost to generate dedicated tests (especially if the test plans to capture the quantitative *Metrics* are sufficient). The results presented in Table 22 are no different regardless of which method is used.

Table 22 - Groupings of *TTL-Metric* Pairs

	GROUP	TTLs	MAJORITY JUDGMENT	EVALUATIVE VOTING
			Median	Pair Averages
METRIC GROUPINGS	Comp (Rev Joint) - Range of Motion	C4	Abs Prefer	5.00
	Comp (Pris Joint) - Range of Motion	C3	Abs Prefer	4.67
		C5	Abs Prefer	5.00
	Comp - Max Lin Velocity	C3	Strongly Pref	3.67
		C5	Prefer	3.67
	Comp - Max Force	C3	Strongly Pref	3.33
		C5	Strongly Pref	3.67
	Comp - Max Angular Vel	C4	Strongly Pref	3.00
	Comp - Max Torque	C4	Prefer	3.33
	Cap - Max Lin Velocity	P1	Prefer	2.60
		P2	Prefer	2.60
		P3	Prefer	4.00
	Cap - Max Force	P1	Prefer	2.40
		P2	Prefer	2.20
		P3	Abs Prefer	4.60
	Cap - Range of Motion, Smoothness, Responsiveness	P1	Strongly Pref	4.40
		P2	Strongly Pref	4.40
		P3	Strongly Pref	4.80

Once the groupings are in place and the least-preferred *TTL-Metric* pairs are eliminated, the presence of the necessary evaluation *Personnel* is determined by using another QUERY -> SCORE -> ELIMINATE process (Box II in Figure 26).

MRED next requires each *Stakeholder* to provide their *Personnel* preferences for each grouping of *TTL-Metric* pairs. Table 23 provides the Stakeholder Preferences for *Personnel* for the Capability - Range of Motion, Smoothness, Responsive grouping (hereby referred to Cap – ROM for brevity). Similarly, *Stakeholder*

Preferences would be captured and scored for the other groupings presented in Table 22.

Table 23 states that the *Stakeholders* prefer that the *End-Users* be the *Technology Users* during tests to capture Cap – ROM data and that *Trained Users* are less desirable (given both it's lower score and average score being less than 0). *Technology Developers* are not a candidate for consideration since they were listed as unavailable earlier in the MRED process (see Figure 24). Additionally, the *Stakeholders* prefer that both *Team Members* and *Participants* not be involved in the CAP – ROM tests since their respective scores are below 0. The *Stakeholders* collective preferences to have neither category of secondary *Personnel* involved in the test could be a result of the relative immaturity of the technology (less than half of the *Capabilities* are available for evaluation and the *System* is entirely unavailable) and/or the nature of the desired *Metrics* to be captured in this grouping.

The aggregated *Stakeholder's Preference* data is consistent between Majority Judgment and Evaluative Voting. The only difference is that MJ conveys a neutral preference for *Team Members* while EV conveys a positive preference.

Table 23 - Stakeholder Preferences for Personnel for Capability – Range of Motion, Smoothness, Responsiveness Grouping

Stakeholder Preferences	**Capability - Range of Motion, Smoothness, Responsiveness Grouping**					**Majority Judgment**	**Evaluative Voting**	
Personnel	Buyer	Eval	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Tech User: End-User	Strongly Pref	Slightly Pref	Strongly Pref	Mod Pref	Abs Prefer	Strongly Pref	**3.20**	**1.64**
Tech User: Trained User	Strongly Rej	Slightly Rej	Strongly Rej	Reject	Abs Reject	Strongly Rej	**-3.60**	**1.67**
Team Member	Neither	Strongly Rej	Slightly Pref	Strongly Rej	Prefer	Neither	**-0.80**	**3.11**
Participant	Mod Pref	Mod Reject	Abs Rej	Strongly Rej	Mod Pref	Mod Reject	**-1.40**	**3.29**

The desired *Knowledge Levels* of this *Personnel* can now be identified from *Stakeholder Preferences*. Table 24 presents the *Stakeholder Preferences* and the corresponding aggregate ratings for the *Knowledge Levels* of the *End-Users*. Note that each *Stakeholder* provides their preferences for the greatest available *Knowledge Levels* and those below. In consultation with the *Stakeholders*, the *MRED Operator* would have to use their discretion if they thought it practical and necessary to boost *Technical Knowledge* with additional training. This may not be practical given typical initial deployments or training time prior to the evaluation.

Some nuances to point out between the MJ and EV data in Table 24 include:

- MJ scores for *Knowledge Levels – Technical* were based upon a second median calculation since the first resulted in a tie (Prefer). This tie had to be broken since only a single knowledge level should be considered.
- MJ graded the "Low" *Knowledge Levels – Operational* as neutral while EV scored it as rejected. The result is the same in either case.

Table 24 - Stakeholder Preferences for Tech User Knowledge Levels for Capability – Range of Motion, Smoothness, Responsiveness Grouping

Stakeholder Preferences	**Capability - Range of Motion, Smoothness, Responsiveness Grouping**					**Majority Judgment**	**Evaluative Voting**	
Knowledge Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
End-User (Low)	Prefer	Slightly Pref	Strongly Pref	Reject	Strongly Pref	Slightly Pref	**1.80**	**2.95**
End-User (Medium)	Strongly Pref	Prefer	Mod Pref	Abs Pref	Mod Pref	Mod Pref	**3.20**	**1.30**
Knowledge Levels - Operational	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
End-User (Low)	Strongly Rej	Neither	Reject	Slightly Pref	Slightly Pref	Neither	**-1.00**	**2.35**
End-User (Medium)	Abs Pref	Neither	Slightly Pref	Prefer	Prefer	Prefer	**2.40**	**1.95**
End-User (High)	Strongly Pref	Neither	Abs Pref	Abs Pref	Strongly Pref	Strongly Pref	**3.60**	**2.07**

From examining the EV data, Table 24 indicates that the *Stakholders* prefer that the *End-Users* have a "Medium" *Technical Knowledge* and a "High" *Operational Knowledge*. However, it appears the *Stakeholders* would accept a "Low" *Technical Knowledge* and a "Medium" *Operational Knowledge* based upon these average scores being above the threshold of 0. It is apparent that the *Stakeholders* do not prefer "Low" *Operational Knowledge* since this aggregate score is negative. To preserve all preferred candidate options, the *Technical Knowledge Levels* of "L – M" (for Low and Medium) and the *Operational Knowledge Levels* of "M – H" (for Medium and High) will be passed through MRED. This *Knowledge Levels* may be further defined to a single level based upon the *Stakeholder Preferences* for other blueprint elements that impose constraints on *Knowledge Levels* (e.g., *Autonomy Levels*).

The next step in defining the Cap – ROM test plans is to capture and aggregate the *Stakeholder Preferences* for the *Autonomy of the Tech – Users.* These preferences and the preferred *Autonomy Levels* are shown in Table 25.

Table 25 - Stakeholder Preferences for Tech User DM Autonomy Levels for Capability – Range of Motion, Smoothness, Responsiveness Grouping

Stakeholder Preferences	**Capability - Range of Motion, Smoothness, Responsiveness Grouping**					**Majority Judgment**	**Evaluative Voting**	
Autonomy Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
End-User (Medium)	Reject	Neither	Slightly Pref	Abs Prefer	Abs Reject	Neither	**-0.40**	**3.85**
End-User (Low)	Slightly Pref	Strongly Pref	Abs Prefer	Strongly Pref	Slightly Rej	Strongly Pref	**2.60**	**2.51**
End-User (Medium)	Abs Prefer	Prefer	Strongly Pref	Reject	Abs Prefer	Prefer	**2.80**	**3.35**
Autonomy Levels - Environmental	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
End-User (None)	Reject	Neither	Mod Reject	Reject	Abs Reject	Reject	**-2.6**	**1.82**
End-User (Low)	Reject	Prefer	Slightly Rej	Slightly Rej	Slightly Rej	Slightly Reject	**-0.60**	**2.19**
End-User (Medium)	Mod Prefer	Slightly Pref	Strongly Pref	Strongly Pref	Mod Prefer	Mod Prefer	**2.60**	**1.34**
End-User (High)	Abs Prefer	Reject	Prefer	Mod Prefer	Abs Prefer	Prefer	**2.40**	**3.29**

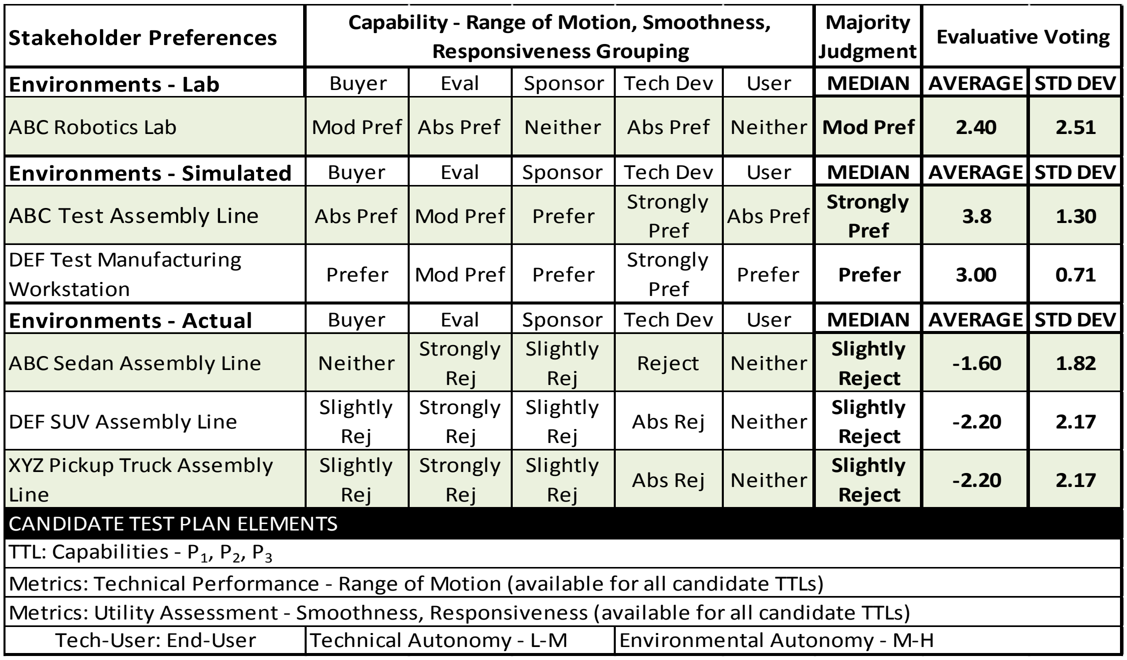
Table 25 indicates that there are two most preferred *Autonomy Levels* for both *Technical* and *Environmental* decisions according to both MJ and EV methods. Before discussing these similarly graded blueprint elements it is important to note that both MJ and EV differ in which blueprint element is graded higher in each category. "Medium" *DM Autonomy – Environmental* is graded "Moderately Prefer" while "High" is graded "Prefer" for MJ. However, EV scores "Medium" at 2.60 while "High" is scored at 2.40. The standard deviation associated with the interval scores is informative; there is greater agreement among the "Medium" score (standard deviation of 1.34) as compared to the "High" score (standard deviation of 3.29).

One limitation of Evaluative Voting is visible in the results of "Low" and "Medium" *DM Autonomy* levels. Both *Technical* and *Environmental* autonomies have their "Low" and "Medium" elements within 0.20 of one another according to EV averages. The MJ rankings also have these same elements closely ranked. This example highlights the difference between these two methods. The closest two elements can be when aggregated by MJ is adjacent preferences (e.g. "Prefer" and "Strongly Prefer"). However, the closest two elements can be when aggregated by EV is a small numerical number allowable by the averages. This could be infinitesimally small based upon the number of the voters. In this sense, MJ will always be more conservative since it does not allow an aggregate preference to be derived.

It is prudent to carry all four of these *Autonomy Levels* (for both *Technical* and *Environmental*) given how close they are to one another. If a *Level* is not clearly identified as the *Stakeholders* provide their preferences for the remaining elements that influence *Knowledge* and *Autonomy Levels,* then the *MRED Operator* can default back to the most preferred by score. The *DM Autonomy Levels* that have negative averages are eliminated from further consideration.

Now that IV in Figure 26 has been addressed, V should be examined. However, both V and VI (of Figure 26) are skipped since *Stakeholder Preferences* in II have determined that no *Team Members* or *Participants* are necessary for this test plan. This leads to VII of Figure 26 which addresses the *Stakeholders Preferences* of the *Environment*. Table 26 presents the *Stakeholder Preferences* for each of the available *Environments*. Both the MJ grades and the EV scores produce similar rankings.

Table 26 - *Stakeholder Preferences* for *Environments* for *Capability* – Range of Motion, Smoothness, Responsiveness Grouping



Stakeholder Preferences	Capability - Range of Motion, Smoothness, Responsiveness Grouping					Majority Judgment	Evaluative Voting	
Environments - Lab	Buyer	Eval	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
ABC Robotics Lab	Mod Pref	Abs Pref	Neither	Abs Pref	Neither	**Mod Pref**	**2.40**	**2.51**
Environments - Simulated	Buyer	Eval	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
ABC Test Assembly Line	Abs Pref	Mod Pref	Prefer	Strongly Pref	Abs Pref	**Strongly Pref**	**3.8**	**1.30**
DEF Test Manufacturing Workstation	Prefer	Mod Pref	Prefer	Strongly Pref	Prefer	**Prefer**	**3.00**	**0.71**
Environments - Actual	Buyer	Eval	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
ABC Sedan Assembly Line	Neither	Strongly Rej	Slightly Rej	Reject	Neither	**Slightly Reject**	**-1.60**	**1.82**
DEF SUV Assembly Line	Slightly Rej	Strongly Rej	Slightly Rej	Abs Rej	Neither	**Slightly Reject**	**-2.20**	**2.17**
XYZ Pickup Truck Assembly Line	Slightly Rej	Strongly Rej	Slightly Rej	Abs Rej	Neither	**Slightly Reject**	**-2.20**	**2.17**
CANDIDATE TEST PLAN ELEMENTS								
TTL: Capabilities - P_1, P_2, P_3								
Metrics: Technical Performance - Range of Motion (available for all candidate TTLs)								
Metrics: Utility Assessment - Smoothness, Responsiveness (available for all candidate TTLs)								
Tech-User: End-User	Technical Autonomy - L-M			Environmental Autonomy - M-H				

Two out of the three *Lab Environments* available for all testing are not candidates for the Cap - ROM pair grouping given the *TTL-Environment* (*X*) relationship matrices presented in Figure 23. As done with preceding test elements, the dissertation author input reasonable preferences for each of the *Stakeholders*. Table 26 shows that none of the *Actual Environments* are preferred for this test plan. This lack of preference could rationaly be explained by the fact that neither the *System* nor all of the *Capabilities* are at least *Functional* so testing in the *Actual Environments* would be premature. The *Simulated Environment* "ABC Test Assembly Line" is the most preferred and is followed by the *Simulated Environment* "DEF Test Manufacturing Workstation" and the *Lab Environment* "ABC Robotics Lab." "ABC Test Assembly Line" is passed through the test plan generator as the VII is completed in Figure 26.

Step VIII in Figure 26 captures and handles *Stakeholder Preferences* for the *Evaluation Scenarios.* Table 27 presents the output *Stakeholder Preferences* and their aggregate scores.

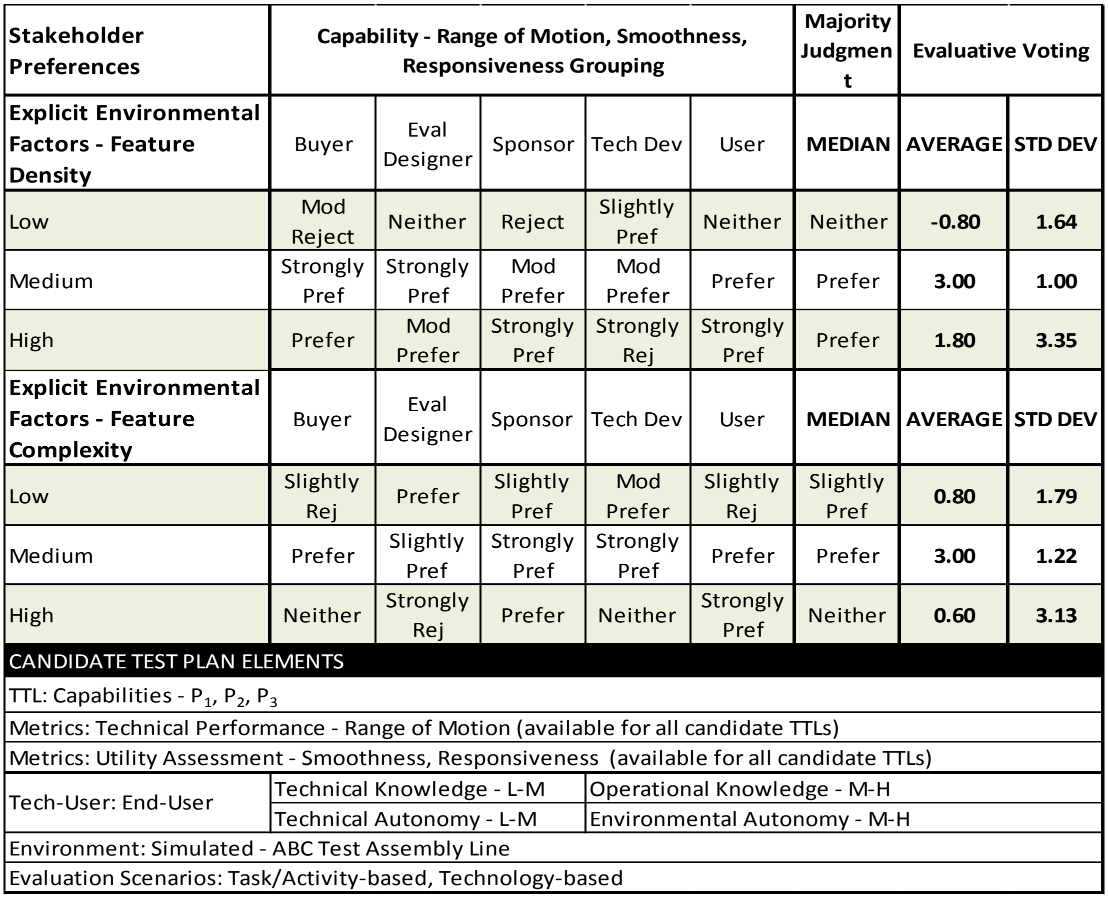
Table 27 - *Stakeholder Preferences* for *Evaluation Scenarios* for *Capability* – Range of Motion, Smoothness, Responsiveness Grouping

Stakeholder Preferences	**Capability - Range of Motion, Smoothness, Responsiveness Grouping**					**Majority Judgment**	**Evaluative Voting**	
Evaluation Scenarios	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Technology-based	Prefer	Abs Pref	Prefer	Abs Pref	Mod Pref	Prefer	**3.60**	**1.34**
Task/Activity-based	Strongly Pref	Strongly Pref	Abs Prefer	Mod Pref	Abs Prefer	Strongly Pref	**4.00**	**1.22**
Environment-based	Mod Reject	Strongly Reject	Reject	Abs Reject	Neither	Reject	**-2.80**	**1.92**
CANDIDATE TEST PLAN ELEMENTS								
TTL: Capabilities - P_1, P_2, P_3								
Metrics: Technical Performance - Range of Motion (available for all candidate TTLs)								
Metrics: Utility Assessment - Smoothness, Responsiveness (available for all candidate TTLs)								
Tech-User: End-User	Technical Knowledge - L-M			Operational Knowledge - M-H				
	Technical Autonomy - L-M			Environmental Autonomy - M-H				
Environment: Simulated - ABC Test Assembly Line								

The MJ grades and EV scores presented in Table 27 are consistent with one another. The *Preferences* indicate that the *Stakeholders* most prefer *Task/Activity-based* scenarios followed closely by *Technology-based* scenarios. Since these two scenario types are close in ranking and score, both will remain candidates for this test plan. It is clear that *Environment-based* scenarios are not preferred given their negative score and median preference of "Reject".

Step VIII in Figure 26 contains the process to capture and handle *Stakeholder Preferences* for the test plan's *Explicit Environmental Factors*. Table 28 presents the *Stakeholder Preferences* and aggregate scores for the *Explicit Enviornmental Factors*.

Table 28 - *Stakeholder Preferences* for *Explicit Environmental Factors* for Capability – Range of Motion, Smoothness, Responsiveness Grouping



Stakeholder Preferences	Capability - Range of Motion, Smoothness, Responsiveness Grouping					Majority Judgment	Evaluative Voting	
Explicit Environmental Factors - Feature Density	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Low	Mod Reject	Neither	Reject	Slightly Pref	Neither	Neither	**-0.80**	**1.64**
Medium	Strongly Pref	Strongly Pref	Mod Prefer	Mod Prefer	Prefer	Prefer	**3.00**	**1.00**
High	Prefer	Mod Prefer	Strongly Pref	Strongly Rej	Strongly Pref	Prefer	**1.80**	**3.35**
Explicit Environmental Factors - Feature Complexity	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Low	Slightly Rej	Prefer	Slightly Pref	Mod Prefer	Slightly Rej	Slightly Pref	**0.80**	**1.79**
Medium	Prefer	Slightly Pref	Strongly Pref	Strongly Pref	Prefer	Prefer	**3.00**	**1.22**
High	Neither	Strongly Rej	Prefer	Neither	Strongly Pref	Neither	**0.60**	**3.13**
CANDIDATE TEST PLAN ELEMENTS								
TTL: Capabilities - P_1, P_2, P_3								
Metrics: Technical Performance - Range of Motion (available for all candidate TTLs)								
Metrics: Utility Assessment - Smoothness, Responsiveness (available for all candidate TTLs)								
Tech-User: End-User	Technical Knowledge - L-M			Operational Knowledge - M-H				
	Technical Autonomy - L-M			Environmental Autonomy - M-H				
Environment: Simulated - ABC Test Assembly Line								
Evaluation Scenarios: Task/Activity-based, Technology-based								

The MJ ranks and EV scores are consistent with one another. One point of discussion is the tie generated by Majority Judgment between "Medium" and "High" *Feature Densities*. Numerous preferences must be removed before one element is ranked above the other.

1. Both elements are graded "Prefer" in the presence of all five of their preferences.

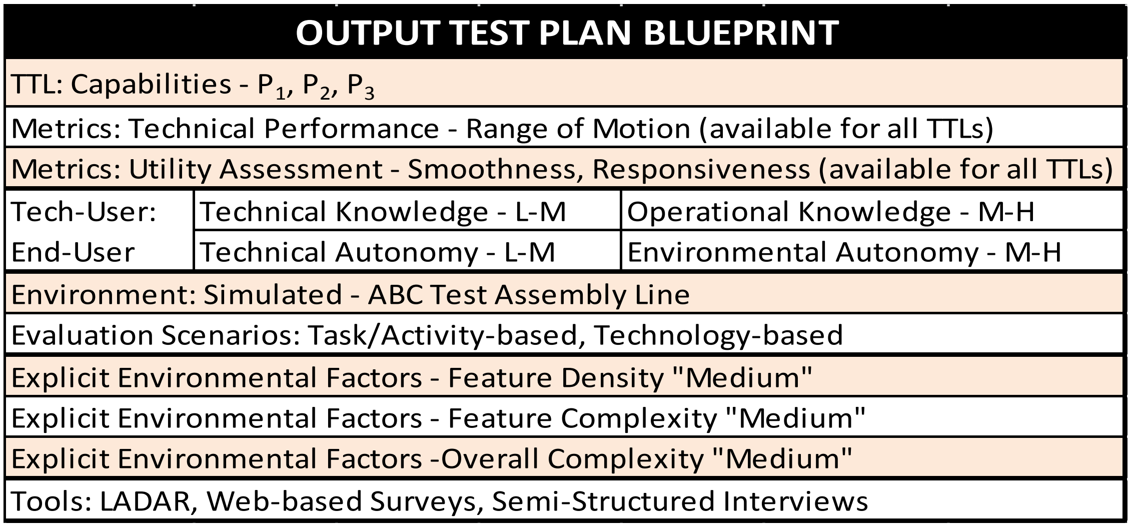
2. Both elements are graded "Moderately Prefer" when a single preference of "Prefer" is removed from both elements.
3. Both elements are graded "Strongly Prefer" when a single preference of "Moderately Prefer" is removed from both elements.
4. "Medium" is ranked "Moderately Prefer" while "High" is ranked "Strongly Rejected" after "Strongly Preferred" is removed. If these final preferences were noted on the spreadsheet, "High" would have been inaccurately ranked lower than "Low." This is because "Low" is originally ranked "Neither" and below the original tied ranks of "Prefer" from "Medium" and "High."

It appears that the *Stakeholders* prefer both "Medium" *Feature Density* and *Feature Complexity* for the test *Environment* – "ABC Test Assembly Line." These *Density* and *Features* lead to an *Overall Complexity* of "Medium." Although the *Overall Complexity* is a dependent value based solely on *Feature Density* and *Feature Complexity*, it is still important to capture. Test plans morph over time where *Stakeholder Preferences* may (and probably should) change with respect to *Feature Density* and *Feature Complexity,* whose values are also changing. Under the conditions the *Overall Complexity* could stay the same (see Figure 18). Likewise, *Overall Complexity* could increase, yet both *Feature Density* and *Feature Complexity* may not (at least *Density* or *Complexity* would have to change).

4.7. Output Blueprints

Table 29 presents MRED's output test plan blueprints for evaluating Range of Motion and capturing user assessments of Smoothness and Responsiveness from the three available *Capabilities* (P_1, P_2, and P_3) on the robot arm. This blueprint is identical regardless of the preference handling method, MJ or EV. The output blueprint also indicates which tools are available and capable of capturing the desired *Metrics.*

Table 29 - Output Test Plan Blueprint for Capability – Range of Motion, Smoothness, Responsiveness Grouping



OUTPUT TEST PLAN BLUEPRINT		
TTL: Capabilities - P_1, P_2, P_3		
Metrics: Technical Performance - Range of Motion (available for all TTLs)		
Metrics: Utility Assessment - Smoothness, Responsiveness (available for all TTLs)		
Tech-User:	Technical Knowledge - L-M	Operational Knowledge - M-H
End-User	Technical Autonomy - L-M	Environmental Autonomy - M-H
Environment: Simulated - ABC Test Assembly Line		
Evaluation Scenarios: Task/Activity-based, Technology-based		
Explicit Environmental Factors - Feature Density "Medium"		
Explicit Environmental Factors - Feature Complexity "Medium"		
Explicit Environmental Factors -Overall Complexity "Medium"		
Tools: LADAR, Web-based Surveys, Semi-Structured Interviews		

Capturing *Stakeholder Preferences* for the other *TTL-Metric* pair groupings presented in Table 22 would yield similar output test plan blueprints to those shown in Table 29. The blueprints provide the *MRED Operator* and the *Evaluation Designer* a guide to define detailed test plan characteristics. Detailed test planning should yield test setup and implementation procedures; specific *End-User* training (based upon stated *Knowledge Levels*); specific *End-User* instructions (based upon stated *Autonomy Levels*); definition and placement of artifacts within the environment (based upon the selected *Environment* and *Explicit Environmental Factors*); specification of evaluation personnel to execute the tests, collect data and maintain a safe environment; etc.

An example test plan, based upon the blueprints shown in Table 29, would include the following details:

- Manufacturing facility workers with less than three months of robot arm experience serve as *End-Users* by running the robot arm technology through "robot obstacle course" situated in the *Simulated* – ABC Test Assembly Line
- The *End-Users* are tasked with two sets of scenarios including: 1) Move the robot arm only in the X/Y/Z on different planes as quickly as possible and 2) Manuever the robot arm around obstacles to touch a goal object only using X, Y, and Z translation.

Additional details, including an evaluation schedule, *Personnel* instructions, *Environment* set up procedures, instrumentation of the test *Environment*, etc. would also have to be specified.

4.8. Majority Judgment v. Evaluative Voting

The robot arm example presents situations where Majority Judgment and Evaluative Voting agree with one another with respect to element rankings. These instances highlight that the same result is achieved whether or not the linguistic preferences are graded (or ranked) ordinally using the median function or whether these preferences are mapped to an interval scale and scored using the mean function. The example also highlights situations where Majority Judgment and Evaluative Voting are not in agreement. This occurs when the two methods highly rank (or score) different elements from the same linguistic preferences. These cases had little impact on the element(s) chosen for the blueprint since the MRED Operator had the discretion to carry forward multiple elements instead of one. Another difference between the two methods is that Evaluative Voting implies a greater level of distinction among scores. This result is an artifact of the EV method.

As an MRED Operator, it may be valuable to understand the relative agreement among the stakeholders. Majority Judgment does not present this level of information in its median calculation. Evaluative Voting preserves the level of agreement among stakeholders in that all voter preferences are averaged. Choosing Evaluative Voting enables the MRED Operator to make decisions based upon more information. The detriment to this method is that only a small portion of the community considers this mathematically rigorous.

4.9. Effort of Algorithms and Stakeholders

The MRED process includes many steps; some which are interactive requiring input from stakeholders and others relying upon algorithms to produce the necessary information. This process is discussed all through Chapter 4 and is further documented in pseudocode presented in Appendix B: Pseudocode. It is important to discuss the estimated computational effort of the algorithms and stakeholders given that MRED is intended for advanced and complex technologies.

The asymptotic or “big-O” notation is a form of estimation to determine the running time of an algorithm that is subject to large input (Sipser, 1997). This estimation is conducted by considering only the highest order term of an expression within an algorithm. As a result the coefficient of the highest order term and all lower order terms are not considered. This notation is formalized as $f(n) = O(n^x)$ where x is the order of the highest term. After reviewing the pseudocode, the highest order of notation is $f(n) = O(n^2)$ and occurs in the following:

- Two nested “For” loops. The first “For” loop is driven by the quantity of *TTLs* and the second “For” loop is driven by the quantity of *Technical Performance Metrics*
- Two nested “For” loops. The first “For” loop is driven by the number of *Capabilities* (plus the *System*, if present) and the second “For” loop is driven by the number of *Utility Assessment Metrics*.

The presence of these two pairs of “For” loops signifies that the quantity of *TTLs* and *Metrics* have the greatest impact on the computational effort on the algorithms. The more *TTLs* and/or *Metrics* that MRED must process, the longer the algorithm will take to run.

MRED requires stakeholders to put forth effort, in addition to the effort expended by the algorithms. The MRED Operator contributes their input at several points during the MRED process. They include:

- *TTLs* and *Metrics* – The targeted *TTLs* and *Metrics* must be input into MRED
- *Technology State* – The *Maturity* of the *Components* must be identified
- *Resources* – The available *Environments, Tools,* and *Personnel* must be identified
- *Relationships* – The relationships between *TTLs, Metrics, Environments,* and *Tools* must be specified in several binary matrices

Likewise, the stakeholders also contribute input in the form of preferences. Preferences are solicited from all stakeholder groups on the following:

- *TTL-Metric* pairs
- *Personnel*
- *Knowledge and Autonomy Levels*

- *Environments*
- *Evaluation Scenarios*
- *Explicit Environmental Factors*

Each of these inputs requires effort by all stakeholder groups to supply MRED with adequate information to formulate one or more feasible and preferred test plan blueprints.

TTLs and *Metrics* should present the greatest demand on the stakeholders as compared to the other elements listed. Specifically, *TTLs* and *Metrics* must be specified; their relationships among themselves and test resources must be identified; preferences must be captured for each *TTL-Metric* pair; and blueprint elements are assigned to one or more grouped *TTL-Metric* pairs. The fewer the *TTLs* and/or *Metrics*, the lesser the burden on the stakeholders. Conversely, the greater the *TTLs* and/or *Metrics*, the heavier the burden on the stakeholders. The discussion of effort highlights the need for the MRED Operator to be experienced with MRED, the technology and its intended operations.

4.10. Summary

MRED provides a formal and systematic method to interactively generate test plans given a specific technology, potential performance metrics, the state of the technology, available resources, and the preferences of the evaluation stakeholders. MRED's output blueprint elements present critical test plan elements that impact each of the detailed characteristics of an evaluation. These blueprints provide a snapshot of the critical evaluation elements that need to be included in the detailed test plan and can enable test plans to be compared to one another over time. Capturing and handling *Stakeholder Preferences* provides traceability so the *Evaluation Designer* can immediately see the impact of existing or changing preferences.

Chapter 5: Application – Speech to Speech Technology

This chapter presents the application of the MRED methodology to design blueprints to evaluate the Speech to Speech Translation Technology (S2S). The dissertation author will focus on the April 2010 S2S evaluation given his direct involvement in the test plan design process (Weiss and Schlenoff, 2011). Presented in Section 1.4, S2S test events evaluated the performance (both quantitative and qualitative) of free-form, two-way, translation systems to enable speakers of different languages to communicate with one another without an interpreter (Weiss and Schlenoff, 2011; Weiss et al., 2008). The April 2010 evaluation focused on communications between English and Pashto speakers where three S2S technologies (produced by three different organizations) operating on smartphones were tested. The dissertation author is serving as the *MRED Operator* in this application based upon his extensive knowledge of the S2S evaluations.

Section 5.1 defines the S2S *TTLs, Metrics* and the MRED-based relationships. Section 5.2 discusses constraint handling and candidate rejection through consideration of the *Technology State, Resources,* and the relationships among these inputs. Section 5.3 presents the *Stakeholder Preferences* that lead to output test plan blueprints. The chapter concludes with a discussion of this application and the validation of MRED using the S2S technology blueprints. The MRED-generated blueprints should provide the *Evaluation Designer* with sufficient guidelines to produce a detailed evaluation. Using MRED's blueprint test plans are presented towards the conclusion of this chapter.

5.1. TTLs, Metrics and Relationships

The first step in applying MRED to the design of an S2S evaluation is to identify the *TTLs* to be considered for testing. Shown in Table 30, the *TTLs* are identified :

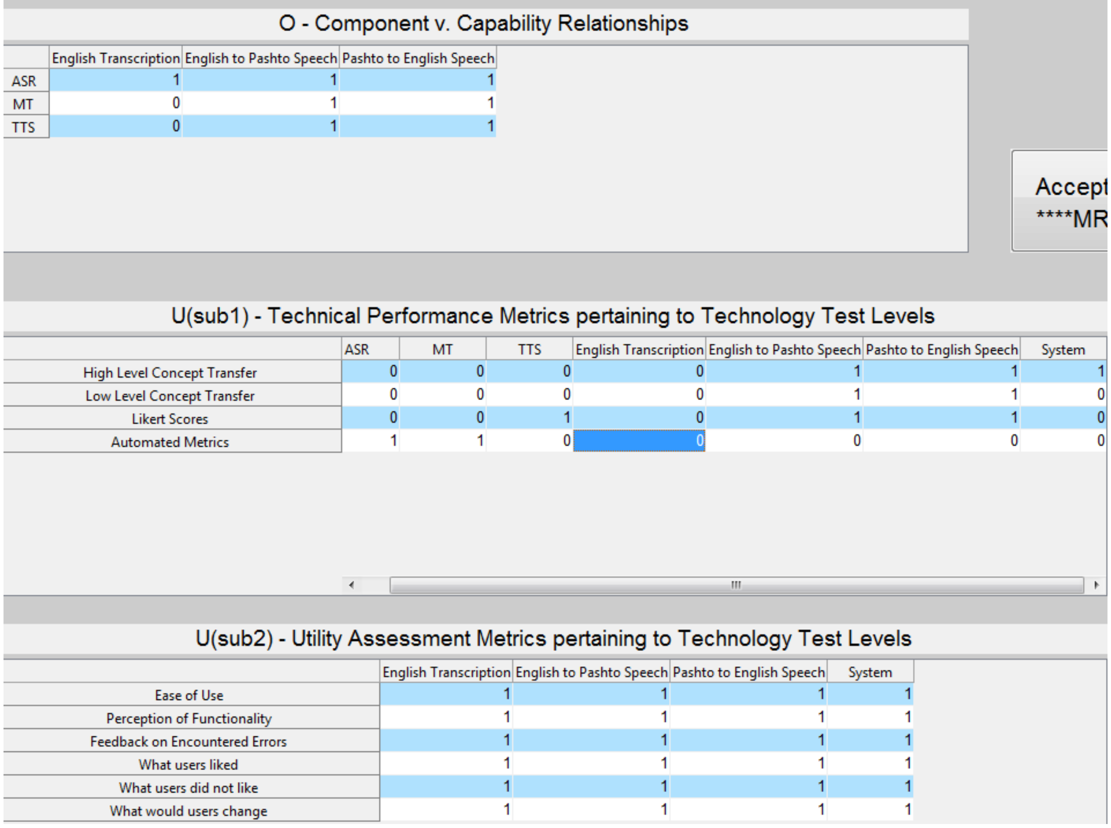
- *Components* – C_1 = Automatic Speech Recognition module (ASR), C_2 = Machine Translation module (MT), and C_3 = Text to Speech module (TTS)
- *Capabilities* – P_1 = English Transcription (technology's ability to present a visual transcription of the input English speech prior to translation), P_2 = English to Pashto Speech translation (technology's ability to translate English speech into Pashto speech), and P_3 = Pashto to English Speech translation (technology's ability to translate Pashto speech into English speech)
- *System* – The technology's full complement of *Capabilities* used to translate a conversation between English and foreign language speakers.

The next step is to identify the pertinent *Metrics* for consideration. The *MRED Operator* identifies both quantitative *Technical Performance Metrics* and qualitative *Utility Assessment Metrics*. These *Metrics* are identified in Table 30.

Table 30 - *TTLs* and *Metrics* Input into MRED

c =	Automatic Speech Recognition (C_1)		Machine Translation (C_2)		Text to Speech (C_3)	
τ = 3						
p =	English Transcription (P_1)		English to Pashto Speech (P_2)		Pashto to English Speech (P_3)	
ϕ = 3						
t =	High Level Concept Transfer		Low Level Concept Transfer		Likert Scores	Set of Automated Metrics
α = 4						
a =	Ease of Use	Perception of Functionality	Feedback on Encountered Errors	What Users Liked	What Users did not Like	What Users would Change
β =6						

Now that the *TTLs* and *Metrics* are defined, the *MRED Operator* can input the data for the Components and *Capabilities* relationship matrix (*O)* and for the *TTLs* and *Metrics* relationship matrices (*U*). These binary matrices are presented in Figure 27.



O - Component v. Capability Relationships

	English Transcription	English to Pashto Speech	Pashto to English Speech
ASR	1	1	1
MT	0	1	1
TTS	0	1	1

U(sub1) - Technical Performance Metrics pertaining to Technology Test Levels

	ASR	MT	TTS	English Transcription	English to Pashto Speech	Pashto to English Speech	System
High Level Concept Transfer	0	0	0	0	1	1	1
Low Level Concept Transfer	0	0	0	0	1	1	0
Likert Scores	0	0	1	0	1	1	0
Automated Metrics	1	1	0	0	0	0	0

U(sub2) - Utility Assessment Metrics pertaining to Technology Test Levels

	English Transcription	English to Pashto Speech	Pashto to English Speech	System
Ease of Use	1	1	1	1
Perception of Functionality	1	1	1	1
Feedback on Encountered Errors	1	1	1	1
What users liked	1	1	1	1
What users did not like	1	1	1	1
What would users change	1	1	1	1

Figure 27: Example Interface of *O* and *U* Relationship Matrices for S2S Evaluation Planning

5.2. *Constraint Handling and Candidate Rejection*

The following subsections present MRED's constraint application process detailed in Figure 22 of Section 4.6.2. Each subprocess enforces constraints imposed by MRED in the following order: *Technology State, Environments, Tools,* and *Personnel*. The overall process includes the definition of technology-specific relationships along with FILTER and ELIMINATE steps (defined in Section 4.6.2).

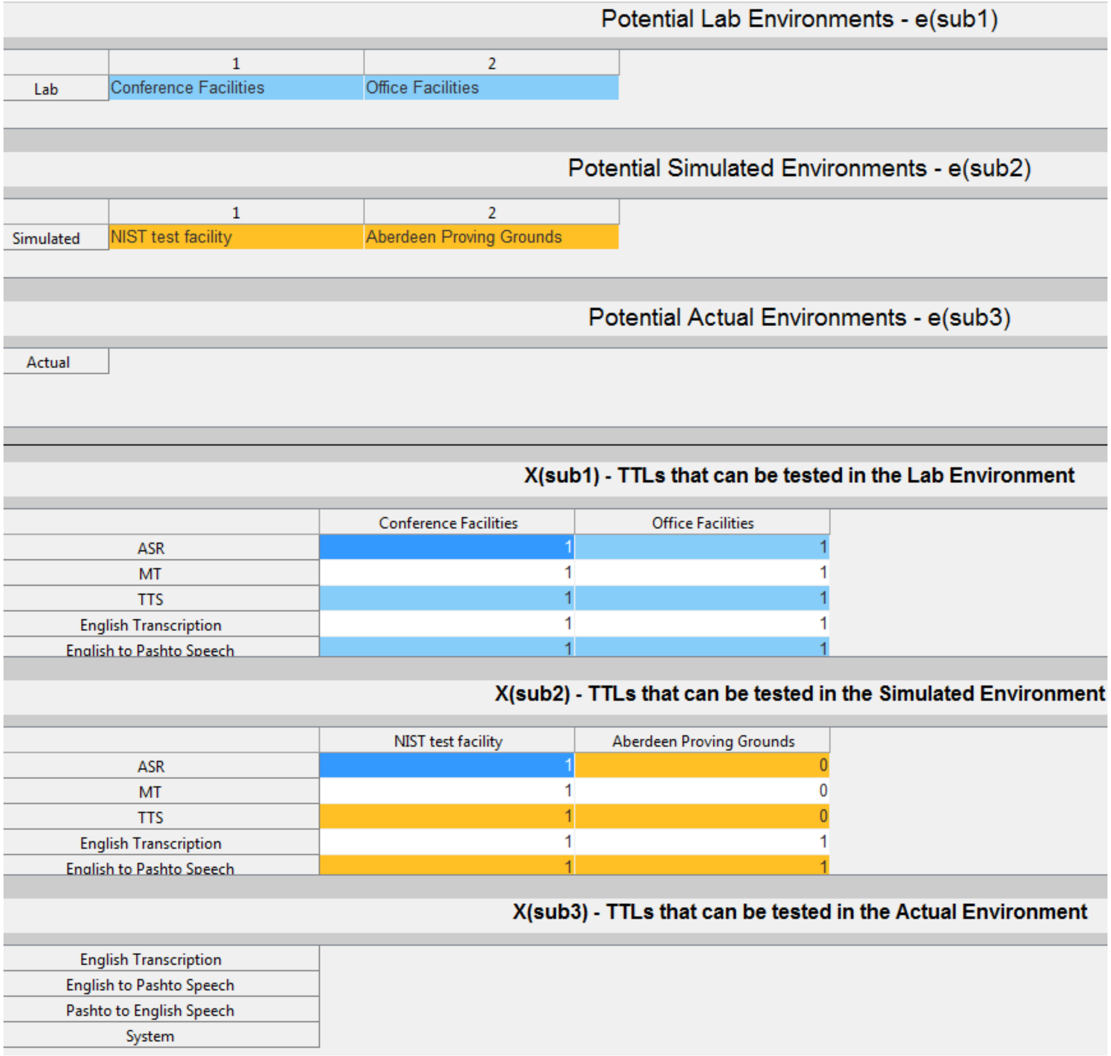
5.2.1. Technology State

The *MRED Operator* now focuses on capturing the *Technology State* data of the *TTLs* to determine which are available for testing. Each of the three *Components* are identified as being Immature. This is appropriate for the scheduled test event; its goal is to provide *Technology Developers* with data to improve their technologies. This *Maturity* information is directly input into MRED.

The MRED Operator now identifies which *Metrics* are still applicable to each of the *TTLs* given the immaturity of each *TTL*. Even with each *TTL* being immature, the MRED Operator declares that all of the corresponding *Metrics* presented in Figure 27 can still be captured.

5.2.2. Environments

The next step is for the *MRED Operator* to input the available test *Environments* and the relationship between the *TTLs* and these *Environments*. A portion of the *environments* and relationships are depicted in Figure 28. There are only two candidate *Lab Environments* and two candidate *Simulated Environments* for testing (no *Actual Environments* are available).



Potential Lab Environments - e(sub1)

	1	2
Lab	Conference Facilities	Office Facilities

Potential Simulated Environments - e(sub2)

	1	2
Simulated	NIST test facility	Aberdeen Proving Grounds

Potential Actual Environments - e(sub3)

Actual

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

	Conference Facilities	Office Facilities
ASR	1	1
MT	1	1
TTS	1	1
English Transcription	1	1
English to Pashto Speech	1	1

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

	NIST test facility	Aberdeen Proving Grounds
ASR	1	0
MT	1	0
TTS	1	0
English Transcription	1	1
English to Pashto Speech	1	1

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

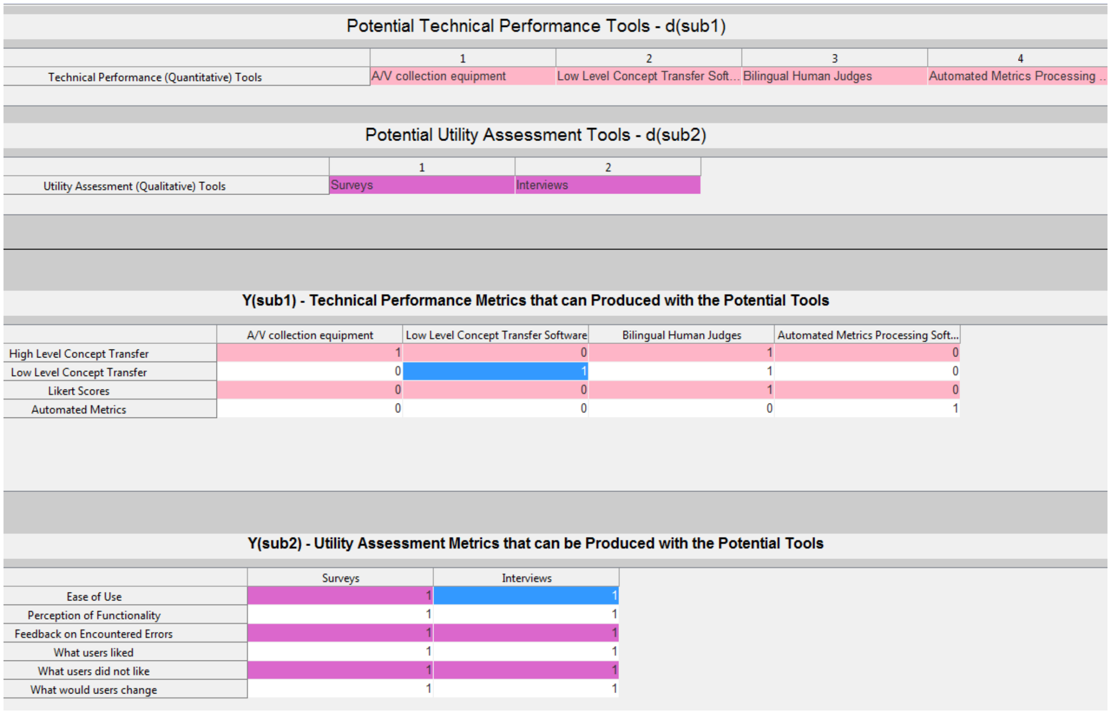
English Transcription
English to Pashto Speech
Pashto to English Speech
System

Figure 28: Matlab Interface of Candidate *Environments* and *X* relationship matrices for S2S Evaluation Planning

After the ELIMINATE and FILTER steps are conducted, it is determined that all of the available *Environments* and candidate *TTLs* are still available for testing.

5.2.3. Tools

The *MRED Operator* now identifies the *Tools* available for testing and the relationship between *Tools* and *Metrics* (*Y* matrices), shown in Figure 29. *Tools* are identified based upon consultation with the *Evaluation Designers* and the *Technology Developers*.



Potential Technical Performance Tools - d(sub1)

	1	2	3	4
Technical Performance (Quantitative) Tools	A/V collection equipment	Low Level Concept Transfer Soft...	Bilingual Human Judges	Automated Metrics Processing ...

Potential Utility Assessment Tools - d(sub2)

	1	2
Utility Assessment (Qualitative) Tools	Surveys	Interviews

Y(sub1) - Technical Performance Metrics that can Produced with the Potential Tools

	A/V collection equipment	Low Level Concept Transfer Software	Bilingual Human Judges	Automated Metrics Processing Soft...
High Level Concept Transfer	1	0	1	0
Low Level Concept Transfer	0	1	1	0
Likert Scores	0	0	1	0
Automated Metrics	0	0	0	1

Y(sub2) - Utility Assessment Metrics that can be Produced with the Potential Tools

	Surveys	Interviews
Ease of Use	1	1
Perception of Functionality	1	1
Feedback on Encountered Errors	1	1
What users liked	1	1
What users did not like	1	1
What would users change	1	1

Figure 29: Example Interface of Candidate *Tools* and *X* relationship matrices for S2S Evaluation Planning

Once the *MRED Operator* identifies *Tools* to support the collection of *Technical Performance* and *Utility Assessment Metrics* and the relationships between these *Tools* and *Metrics*, the ELIMINATE and FILTER steps are completed in Matlab. All *Tools* and *Metrics* are still candidate test plan elements at this point.

5.2.4. Personnel

The next step for the *MRED Operator* is to determine the available *Personnel* and corresponding *Knowledge Levels*. In case of the S2S technologies, *Knowledge Levels* are defined as:

- *Technical Knowledge* – Understanding of the S2S technologies and their related hardware platforms (i.e. smartphones) that is being considered for evaluation. This includes both knowing what the S2S systems can do and how best to use them.

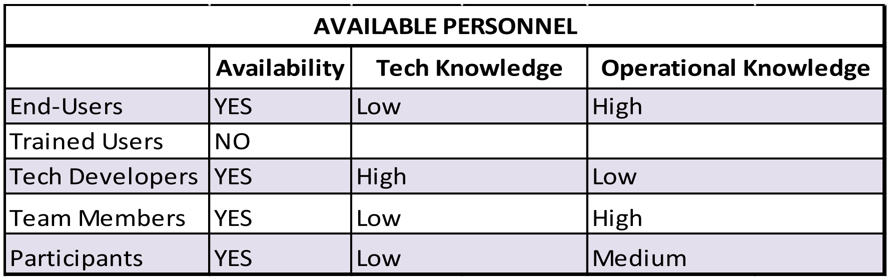
- *Operational Knowledge* – Understanding of the tactical military environments that the S2S technologies are intended.

Each category of *Personnel* is assigned *Knowledge Levels* to be input into MRED.

- *End-User* personnel are identified as US Marines with “High” levels of *Operational Experience* in foreign environments. These available *End-Users* had never received training on speech translation technologies, yet were familiar with smartphone technology so their technical knowledge was deemed “Low.” These *End-Users* are being directly considered to use the S2S technology as English speakers in the evaluations.
- The actual *Technology Developers* are identified as candidate *Tech Users*. Given their thorough knowledge of their own technology, their *Technical Knoweldge* is noted as “High.” However, these *Technology Developers* are greatly limited in their military understanding so their *Operational Knowledge* is listed as “Low.”
- The *MRED Operator* did not identify the need for *Trained Users* to support these test plans. This was confirmed in consultation with the *Stakeholders*.
- *Personnel* in the category of *Team Members* are needed. *Team Members will be* stationed in close proximity to the *End-Users.* In prior evaluations, *Team Members* may be responsible for taking notes on *End-Users* interactions with foreign personnel, providing security, etc. The *MRED Operator* identifies the US Marines (already noted as *End Users*) as *Team Member* candidates and assigns them a “High” level of *Operational Knowledge* and a “Low” level of *Technical Knowledge*.

- *Participants* are identified by the *MRED Operator* as bilingual members of the Pashto-speaking community as *Participants*. These *Participants* have never been exposed to the S2S technology yet they have a small amount of experience with smartphone technology. Their *Technical Knowledge* is assigned a value of "Low." However, their *Operational Knowledge* is considered "Medium" since some of these *Personnel* have experience as interpreters for the US Military. These *Participants* would interact with the *End-Users* through the S2S technologies. The *Participants* do not physically engage with the S2S technology, yet are recipients of the technology's translations and provide the Pashto speech for the technology to translate back to the *Tech User.* Table 31 presents a screenshot of the Matlab interface showing these inputs.

Table 31 - Personnel and Knowledge Levels for S2S Evaluation Planning



AVAILABLE PERSONNEL			
	Availability	**Tech Knowledge**	**Operational Knowledge**
End-Users	YES	Low	High
Trained Users	NO		
Tech Developers	YES	High	Low
Team Members	YES	Low	High
Participants	YES	Low	Medium

For this S2S technology, the relationships defined in the *O, U, X,* and *Y* matrices did not eliminate any of the available test plan elements (i.e. all of the candidate input passed through the planner to this point). This phenomenon occurs because the S2S technology's *TTLs* are developed enough for testing even though they are all classified as imature. Had the S2S technology been less *Mature* leading up to the April 2010 evalaution, it's likely that some *Components* and *Capabilities* would have been eliminated by MRED because of the inability to capture the associated *Metrics*.

5.3. Stakeholder Preference Handling

The next phase of the MRED process is to capture and handle the preferences of the S2S technology *Stakeholders*. For planning the evaluations of S2S technologies, the following *Stakeholders* are identified:

- *Buyers* – No *Buyers* are involved in the evaluation of this technology and therefore did not have any influence in the test plans. The *Buyers'* input for all *Stakeholder Preferences* will be noted as "NV" ("no vote") so as not to influence the aggregate preferences
- *Evaluation Designer* – The NIST evaluation team is identified as the *Evaluation Designer* for these test plans. In addition, these same *Personnel* implemented the testing.
- *Sponsor* – The DARPA program manager is identified as the *Sponsor*
- *Technology Developer* – The collective opinions of the three *Technology Developer* organizations are noted for the *Technology Developer* preferences.
- *User* – Numerous military personnel represented the collective thoughts of the *User* group. This included the opinions of several Marines (who are targeted as the *End-User* population) and a high-ranking Marine officer who has significant experience with S2S technologies for military use.

Capturing and handling *Stakeholder Preferences* centers around the sequential nine steps presented in Figure 26 and discussed in 4.6.3.5.

5.3.1. TTL-Metric Pairs

MRED queries the *Stakeholders' Preferences* using the 11-point linguistic ordinal scale described in Section 4.6.3. MRED uses this data to calculate the Majority

Judgment grades and Evaluative Voting scores for the *TTL-Metric* pairs. For brevity, the following abbreviations are used to indicate specific *Metrics:*

- HLCT – High Level Concept Transfer
- LLCT – Low Level Concept Transfer
- Likert – Likert Scoring of Utterances
- PoF – Perception of Functionality
- FoEE – Feedback on Encountered Errors
- Likes – What users liked
- Dislikes – What users did not like
- Change – What users would change

Table 32 presents the *Stakeholder Preferences* for the *TTL-Metric* pairs using the Evaluative Voting method described in Section 4.6.3. The dissertation author, continuing as the *MRED Operator*, inputting the ratings from the perspective of the *Stakeholders* for Table 32 based upon his detailed knowledge of the actual S2S test planning process.

Table 32 - *Stakeholder Preferences* of *TTL-Metric* Pairs for S2S Evaluation Planning

Stakeholder Preferences	**STAKEHOLDERS**				
TTL-Metric Pairs	Buyer	Eval Designer	Sponsor	Tech Dev	User
ASR - Automated Metrics	NV	Abs Pref	Prefer	Abs Pref	NV
MT - Automated Metrics	NV	Abs Pref	Prefer	Abs Pref	NV
TTS - Likert Scores	NV	Reject	Strongly Rej	Slightly Rej	NV
English Transcription - Ease of Use	NV	Reject	Abs Rej	Neither	Slightly Pref
English Transcription - PoF	NV	Reject	Abs Rej	Neither	Slightly Pref
English Transcription - FoEE	NV	Reject	Abs Rej	Neither	Slightly Pref
English Transcription - Likes	NV	Reject	Abs Rej	Neither	Slightly Pref
English Transcription - Dislikes	NV	Reject	Abs Rej	Neither	Slightly Pref
English Transcription - Change	NV	Reject	Abs Rej	Neither	Slightly Pref
English to Pashto - HLCT	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref
English to Pashto - LLCT	NV	Prefer	Prefer	Abs Pref	Neither
English to Pashto - Likert Scores	NV	Prefer	Strongly Pref	Abs Pref	Neither
English to Pashto - Ease of Use	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
English to Pashto - PoF	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
English to Pashto - FoEE	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
English to Pashto - Likes	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
English to Pashto - Dislikes	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
English to Pashto - Change	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - HLCT	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref
Pashto to English - LLCT	NV	Prefer	Prefer	Abs Pref	Neither
Pashto to English - Likert Scores	NV	Prefer	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - Ease of Use	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - PoF	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - FoEE	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - Likes	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - Dislikes	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
Pashto to English - Change	NV	Abs Pref	Strongly Pref	Abs Pref	Abs Pref
System - HLCT	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref
System - Ease of Use	NV	Abs Pref	Abs Pref	Strongly Pref	Abs Pref
System - PoF	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref
System - FoEE	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref
System - Likes	NV	Abs Pref	Abs Pref	Strongly Pref	Abs Pref
System - Dislikes	NV	Abs Pref	Abs Pref	Strongly Pref	Abs Pref
System - Change	NV	Abs Pref	Abs Pref	Strongly Pref	Abs Pref

Table 33 builds upon Table 32 by presenting the average scores and standard deviations of these preferences. The *MRED Operator* designated the threshold for test consideration to be all *TTL-Metric* pairs graded higher than “Neither” (for MJ) or greater than 0 (for EV). Any *TTL-Metric* pairs below this threshold are eliminated. Only those pairs that fall into one of the exceptions (noted in Section 4.6.3.5) would be further scrutinized for inclusion.

Table 33 - Ordered Grades and Scores *Stakeholder Preferences* for TTL-Metric Pairs for S2S Evaluation Planning

	MAJORITY JUDGMENT	EVALUATIVE VOTING	
TTL-Metric Pairs	MEDIAN	AVERAGE	STD DEV
English to Pashto - HLCT	Abs Pref	**5.00**	**0.00**
Pashto to English - HLCT	Abs Pref	**5.00**	**0.00**
System - HLCT	Abs Pref	**5.00**	**0.00**
System - PoF	Abs Pref	**5.00**	**0.00**
System - FoEE	Abs Pref	**5.00**	**0.00**
English to Pashto - Ease of Use	Abs Pref	**4.75**	**0.50**
English to Pashto - PoF	Abs Pref	**4.75**	**0.50**
English to Pashto - FoEE	Abs Pref	**4.75**	**0.50**
English to Pashto - Likes	Abs Pref	**4.75**	**0.50**
English to Pashto - Dislikes	Abs Pref	**4.75**	**0.50**
English to Pashto - Change	Abs Pref	**4.75**	**0.50**
Pashto to English - Ease of Use	Abs Pref	**4.75**	**0.50**
Pashto to English - PoF	Abs Pref	**4.75**	**0.50**
Pashto to English - FoEE	Abs Pref	**4.75**	**0.50**
Pashto to English - Likes	Abs Pref	**4.75**	**0.50**
Pashto to English - Dislikes	Abs Pref	**4.75**	**0.50**
Pashto to English - Change	Abs Pref	**4.75**	**0.50**
System - Ease of Use	Abs Pref	**4.75**	**0.50**
System - Likes	Abs Pref	**4.75**	**0.50**
System - Dislikes	Abs Pref	**4.75**	**0.50**
System - Change	Abs Pref	**4.75**	**0.50**
ASR - Automated Metrics	Abs Pref	**4.33**	**1.15**
MT - Automated Metrics	Abs Pref	**4.33**	**1.15**
Pashto to English - Likert Scores	Strongly Pref	**4.25**	**0.96**
English to Pashto - Likert Scores	Prefer	**3.00**	**2.16**
English to Pashto - LLCT	Prefer	**2.75**	**2.06**
Pashto to English - LLCT	Prefer	**2.75**	**2.06**
~~English Transcription - Ease of Use~~	Reject	**-1.75**	**2.75**
~~English Transcription - PoF~~	Reject	**-1.75**	**2.75**
~~English Transcription - FoEE~~	Reject	**-1.75**	**2.75**
~~English Transcription - Likes~~	Reject	**-1.75**	**2.75**
~~English Transcription - Dislikes~~	Reject	**-1.75**	**2.75**
~~English Transcription - Change~~	Reject	**-1.75**	**2.75**
~~TTS - Likert Scores~~	Reject	**-2.67**	**1.53**

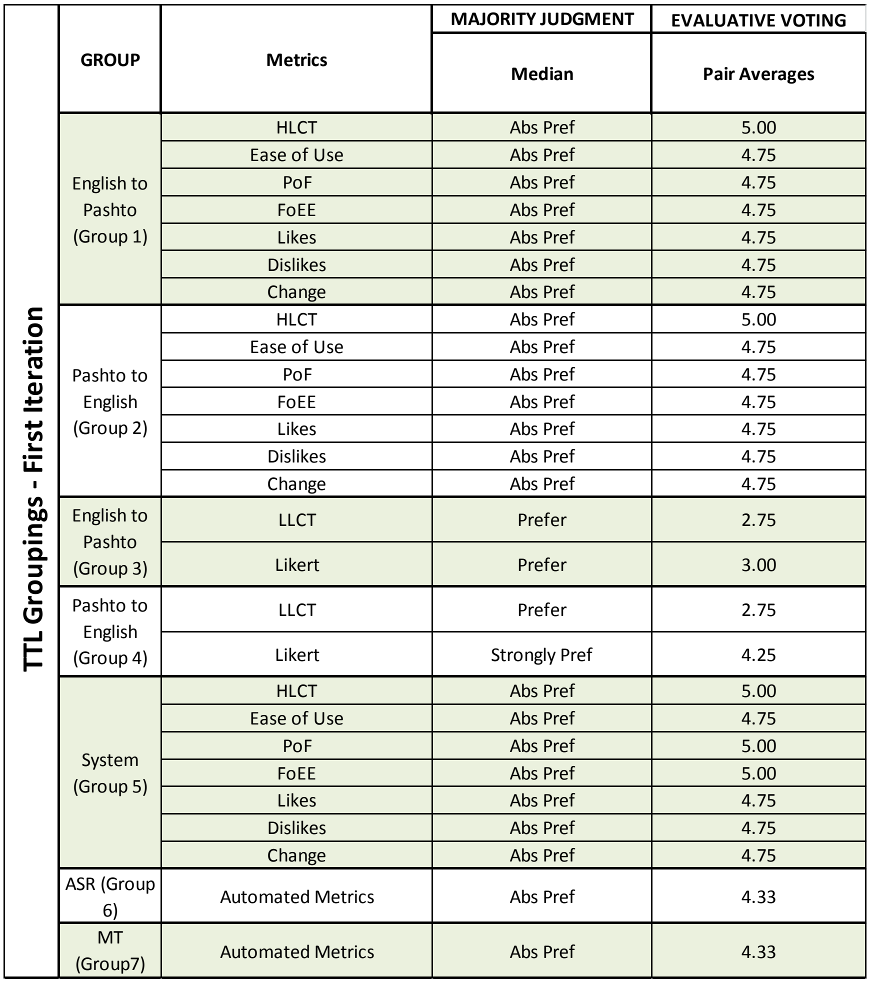
Given their low aggregate scores by the *Stakeholders,* the *Capability* of English Transcription and the *Component* of Text to Speech (TTS) are no longer considered for evaluation.

It is important to highlight that the ordered data from Majority Judgment and Evaluative Voting are in agreement. This is evident both in which *TTL-Metric* pairs are preferred and rejected. In addition, those pairs that are preferred are ordered identically from both methods. Another note is that three pairs are assessed based upon three *Stakeholder Preferences* while the remaining pairs are assessed by four *Stakeholder Preferences*.

Grouping of the remaining candidate *TTL-Metric* pairs is a step unique to identifying candidate *TTL-Metric* pairs. Grouping is the third step in addressing *Stakeholder Preferences* for *TTL-Metric* pairs; the process is QUERY -> SCORE -> GROUP -> ELIMINATE. Table 34 presents the *MRED Operator's* first iteration at grouping the candidate *TTL-Metric* pairs. The pairs were selected to be grouped by *TTL,* yet several *TTLs* (i.e. English to Pashto and Pashto to English) had to be split based upon the *MRED Operator's* experience of how the metrics for these *TTLs* are captured. Those metrics listed within Groups 1 and 2 (identified in Table 34) are captured by processing live speech data. This speech data is captured from English (*Tech Users*) and foreign language speakers (*Participants*) during their interactions with the technology.

However, the metrics listed within Groups 3 and 4 (of Table 34) are generated by processing data that was collected prior to the test event. Noting that all four of these groups could not be evaluated under the same conditions, it's prudent to split them up. The conditions under which *Metrics* are captured for Group 5 are similar to the *Metrics* captured for Groups 1 and 2. Likewise, the *Metrics* evaluation conditions for Groups 3 and 4 are similar to that of Groups 6 and 7.

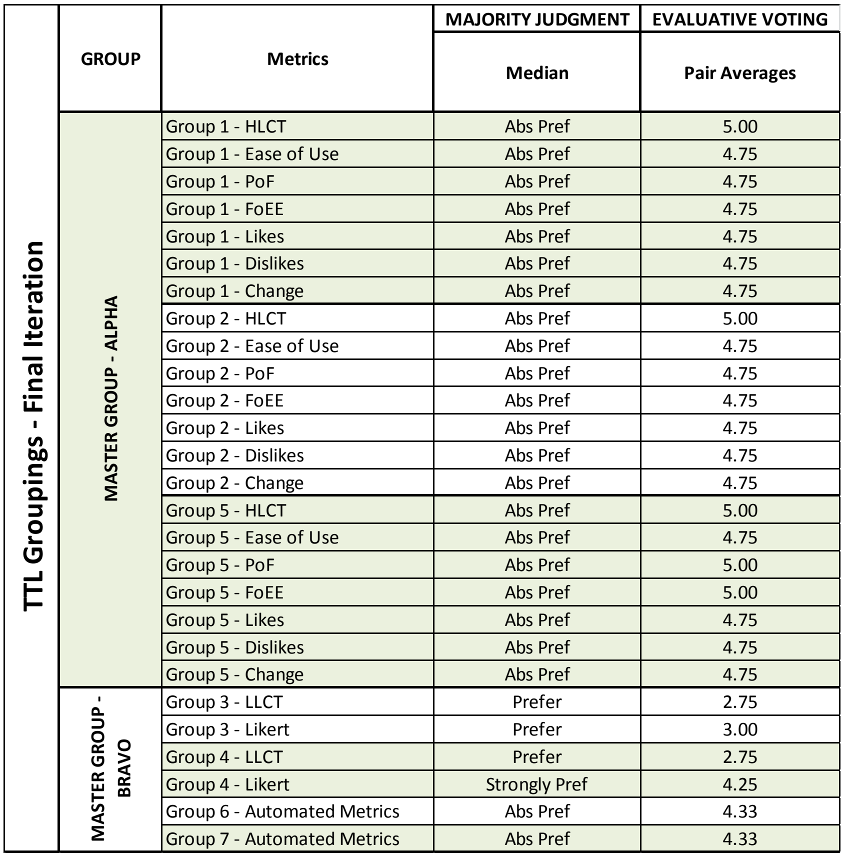
Table 34 – First Iteration of Groupings of *TTL-Metric* Pairs for S2S Evaluation Planning



	GROUP	Metrics	MAJORITY JUDGMENT	EVALUATIVE VOTING
			Median	Pair Averages
TTL Groupings - First Iteration	English to Pashto (Group 1)	HLCT	Abs Pref	5.00
		Ease of Use	Abs Pref	4.75
		PoF	Abs Pref	4.75
		FoEE	Abs Pref	4.75
		Likes	Abs Pref	4.75
		Dislikes	Abs Pref	4.75
		Change	Abs Pref	4.75
	Pashto to English (Group 2)	HLCT	Abs Pref	5.00
		Ease of Use	Abs Pref	4.75
		PoF	Abs Pref	4.75
		FoEE	Abs Pref	4.75
		Likes	Abs Pref	4.75
		Dislikes	Abs Pref	4.75
		Change	Abs Pref	4.75
	English to Pashto (Group 3)	LLCT	Prefer	2.75
		Likert	Prefer	3.00
	Pashto to English (Group 4)	LLCT	Prefer	2.75
		Likert	Strongly Pref	4.25
	System (Group 5)	HLCT	Abs Pref	5.00
		Ease of Use	Abs Pref	4.75
		PoF	Abs Pref	5.00
		FoEE	Abs Pref	5.00
		Likes	Abs Pref	4.75
		Dislikes	Abs Pref	4.75
		Change	Abs Pref	4.75
	ASR (Group 6)	Automated Metrics	Abs Pref	4.33
	MT (Group7)	Automated Metrics	Abs Pref	4.33

To alleviate the burden on the *Stakeholders* in providing preferences for the remaining test plan elements, the *MRED Operator* chooses to partition the seven groups listed in Table 34. The seven groups are consolidated into two master groups, identified as Alpha and Bravo (presented in Table 35). These two master groups lay the foundation for two sets of test plans. The *MRED Operator* arrives at these master groupings given prior input from the *Stakeholders*. It is critical that the *MRED Operator* be informed of the high level motivations of each of the *Stakeholders*. The next step is to identify which evaluation *Personnel* are preferred by the *Stakeholders* for the Alpha and Bravo test plans.

Table 35 - Master Groupings (Final Iteration) of TTL-Metric Pairs for S2S Evaluation Planning

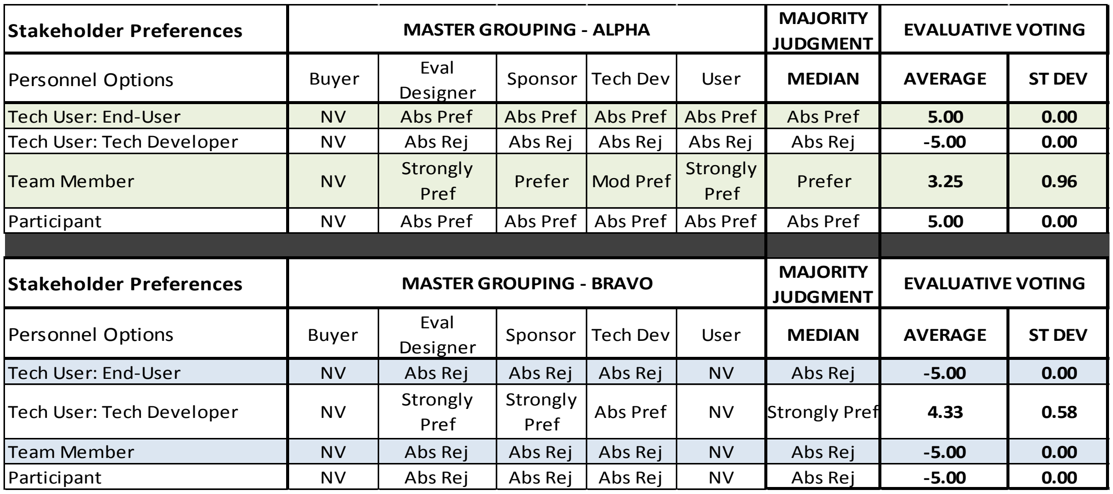


TTL Groupings - Final Iteration	GROUP	Metrics	MAJORITY JUDGMENT	EVALUATIVE VOTING
			Median	Pair Averages
TTL Groupings - Final Iteration	MASTER GROUP - ALPHA	Group 1 - HLCT	Abs Pref	5.00
		Group 1 - Ease of Use	Abs Pref	4.75
		Group 1 - PoF	Abs Pref	4.75
		Group 1 - FoEE	Abs Pref	4.75
		Group 1 - Likes	Abs Pref	4.75
		Group 1 - Dislikes	Abs Pref	4.75
		Group 1 - Change	Abs Pref	4.75
		Group 2 - HLCT	Abs Pref	5.00
		Group 2 - Ease of Use	Abs Pref	4.75
		Group 2 - PoF	Abs Pref	4.75
		Group 2 - FoEE	Abs Pref	4.75
		Group 2 - Likes	Abs Pref	4.75
		Group 2 - Dislikes	Abs Pref	4.75
		Group 2 - Change	Abs Pref	4.75
		Group 5 - HLCT	Abs Pref	5.00
		Group 5 - Ease of Use	Abs Pref	4.75
		Group 5 - PoF	Abs Pref	5.00
		Group 5 - FoEE	Abs Pref	5.00
		Group 5 - Likes	Abs Pref	4.75
		Group 5 - Dislikes	Abs Pref	4.75
		Group 5 - Change	Abs Pref	4.75
	MASTER GROUP - BRAVO	Group 3 - LLCT	Prefer	2.75
		Group 3 - Likert	Prefer	3.00
		Group 4 - LLCT	Prefer	2.75
		Group 4 - Likert	Strongly Pref	4.25
		Group 6 - Automated Metrics	Abs Pref	4.33
		Group 7 - Automated Metrics	Abs Pref	4.33

5.3.2. Personnel: Presence

Now that the two test plan master groups are in place and the least-preferred *TTL-Metric* pairs are eliminated, *Stakeholder Preferences* must be captured to address the evaluation *Personnel.* Table 36 presents the preferences of the *Stakeholders* and the aggregate scores for each of the available *Personnel* that are considered for the Alpha and Bravo test plans. It is evident that the *Stakeholders* strongly prefer the presence of *End-Users* and reject *Tech Developers* to be *Tech Users* for the Alpha test plan. These strong preferences are understandable since a majority of the *Metrics* captured in Alpha are *Utility Assessments*. If this test plan had solely contained *Utility Assessments,* then MRED would eliminate *Tech Developers* as an option for the *Stakeholders* to consider. The *Stakeholders* are also in favor of the *Team Members* and *Participants* being involved in this test plan. Overall, the *Stakeholders* see value in having a *Team Member* observe the evaluation and become another source of qualitative feedback on the technologies' performance. Including *Participants* is a necessity for Alpha group. The *TTLs* included in Alpha are at the *Capability* and *System* levels which require live speakers to converse in English and Pashto.

Table 36 - *Stakeholder Preferences* for *Personnel* for S2S Evaluation Planning



Stakeholder Preferences	**MASTER GROUPING - ALPHA**					**MAJORITY JUDGMENT**	**EVALUATIVE VOTING**	
Personnel Options	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Tech User: End-User	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref	Abs Pref	**5.00**	**0.00**
Tech User: Tech Developer	NV	Abs Rej	Abs Rej	Abs Rej	Abs Rej	Abs Rej	**-5.00**	**0.00**
Team Member	NV	Strongly Pref	Prefer	Mod Pref	Strongly Pref	Prefer	**3.25**	**0.96**
Participant	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref	Abs Pref	**5.00**	**0.00**
Stakeholder Preferences	**MASTER GROUPING - BRAVO**					**MAJORITY JUDGMENT**	**EVALUATIVE VOTING**	
Personnel Options	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Tech User: End-User	NV	Abs Rej	Abs Rej	Abs Rej	NV	Abs Rej	**-5.00**	**0.00**
Tech User: Tech Developer	NV	Strongly Pref	Strongly Pref	Abs Pref	NV	Strongly Pref	**4.33**	**0.58**
Team Member	NV	Abs Rej	Abs Rej	Abs Rej	NV	Abs Rej	**-5.00**	**0.00**
Participant	NV	Abs Rej	Abs Rej	Abs Rej	NV	Abs Rej	**-5.00**	**0.00**

Table 36 presents complete agreement between the ordering of the elements as generated by Majority Judgment and Evaluative Voting. One plausible explanation for this is due to the high level of stakeholder agreement among the elements (seen in the very low standard deviations).

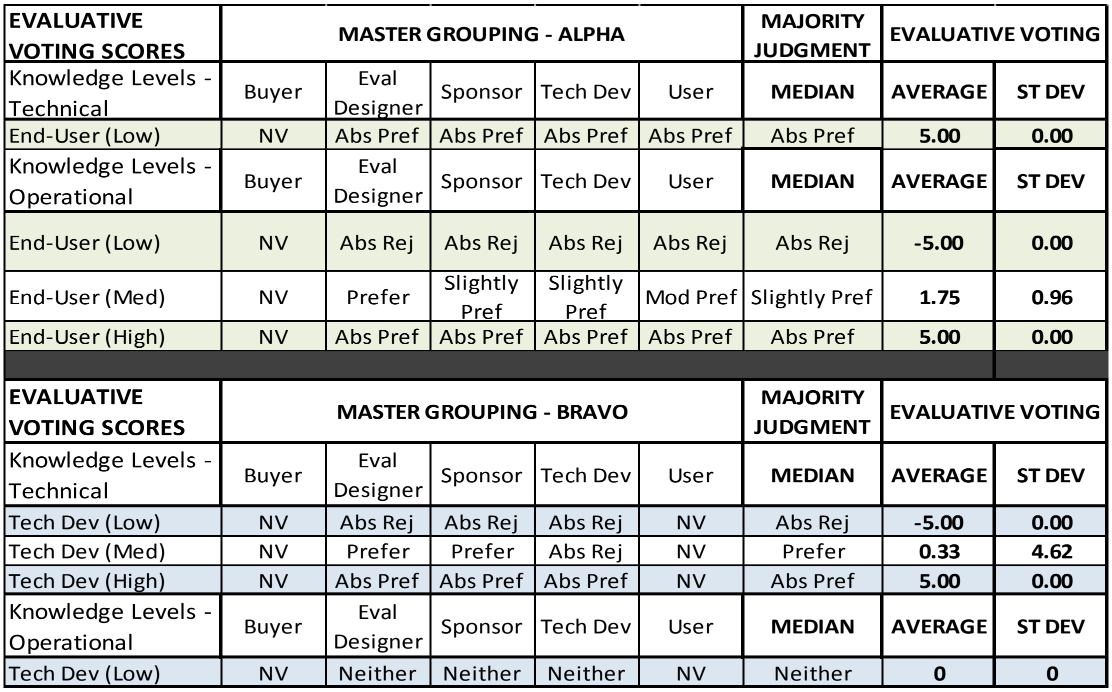
The Bravo test plan is different from Alpha in that neither *End-Users, Team Members,* nor *Participants* are preferred. The *Metrics* noted in Bravo are captured through processing of previously-collected English, foreign speech, and text. A deep understanding of the software is required to capture these *Metrics* so the *Tech Developers* are the ideal *Tech Users* for this test plan.

5.3.3. Personnel: Tech-Users: Knowledge Levels

The next step in the iterative process of capturing and handling *Stakeholder Preferences* is to determine the *Knowledge Levels* of the *Tech Users*. Table 37 presents the *Stakeholder Preferences* and the aggregate scores for the *End-Users* (Alpha) and *Tech Developers* (Bravo). The rationale behind these preferences are:

- A majority of the *Stakeholders* want the technology to be easy to pick-up and learn. So in testing, it's preferred that the *End-Users* have "Low" *Technical Knowledge.*
- It's important that the technology be used with the intended vocabulary (i.e. military jargon) and that the dialogues be in keeping with the types of situations that are commonly encountered in the operational settings.This is the justification behind the *Stakeholders' Preference* for "High" *Operational Knowledge* for the *End-Users.*
- The *Technology Developers* are the only *Personnel* capable of executing the software algorithms to support the generation of the *Metrics* in the Bravo Test Plan. *Technology Developers have a* thorough understanding of the technology and therefore their *Technical Knowledge* is "High."
- The *Tech Developers* are using the technology for the sole purpose of inputting data directly into a computer (without the need for anyone to speak into or listen to the technology). The level of *Operational Knowledge* does not matter and "Low" is chosen since this is the only option.

Table 37 - *Stakeholder Preferences* for *Tech User Knowledge Levels* for S2S Evaluation Planning



EVALUATIVE VOTING SCORES	**MASTER GROUPING - ALPHA**					**MAJORITY JUDGMENT**	**EVALUATIVE VOTING**	
Knowledge Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
End-User (Low)	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref	Abs Pref	**5.00**	**0.00**
Knowledge Levels - Operational	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
End-User (Low)	NV	Abs Rej	Abs Rej	Abs Rej	Abs Rej	Abs Rej	**-5.00**	**0.00**
End-User (Med)	NV	Prefer	Slightly Pref	Slightly Pref	Mod Pref	Slightly Pref	**1.75**	**0.96**
End-User (High)	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref	Abs Pref	**5.00**	**0.00**

EVALUATIVE VOTING SCORES	**MASTER GROUPING - BRAVO**					**MAJORITY JUDGMENT**	**EVALUATIVE VOTING**	
Knowledge Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Tech Dev (Low)	NV	Abs Rej	Abs Rej	Abs Rej	NV	Abs Rej	**-5.00**	**0.00**
Tech Dev (Med)	NV	Prefer	Prefer	Abs Rej	NV	Prefer	**0.33**	**4.62**
Tech Dev (High)	NV	Abs Pref	Abs Pref	Abs Pref	NV	Abs Pref	**5.00**	**0.00**
Knowledge Levels - Operational	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Tech Dev (Low)	NV	Neither	Neither	Neither	NV	Neither	**0**	**0**

Table 37 is another example where there is agreement between Majority Judgment and Evaluative Voting with respect to element ordering.

5.3.4. Personnel: Tech-Users: Autonomy Levels

Stakeholder Preferences are captured and handled to determine the *Autonomy Levels* of the *Tech-Users*. For S2S technologies, *Autonomy Levels* are defined as:

- *Technical Autonomy* – The level of authority a *Personnel* group has in interacting with the S2S technology.
- *Environmental Autonomy* – The level of authority a *Personnel* group has to interact with the test environment (e.g., other *Personnel* groups). *Environmental Autonomy* includes the level or difficulty of speech (or other data input) that is used by the *Personnel* groups to communicate with one another since these test plans are focused on evaluating speech translation technologies.

Table 38 presents the *Stakeholder Preferences* and aggregate scores for the *Autonomy Levels* of the chosen *Tech Users*. In keeping with MRED's constraints, an *Autonomy Level* cannot exceed its corresponding *Knowledge Level* (described earlier in Section 4.4.3). Some of the rationale behind the *Stakeholder Preferences* in Table 38 include:

- Most *Stakeholders* prefer that *End-Users* be given as much *Technical Autonomy* as their *Knowledge Level* allows. *Technology Developers* may feel differently. Although they want the *End-Users* to be comfortable with the technology, *Technology Developers* recognize that the more freedom the *End-Users* are given (to exercise the technology), the more likely the technology will encounter errors.
- The *End-Users* should interact with the *Environment* and other evaluation personnel (*Team Members* and/or *Participants*, if preferred) speaking into the technology as they prefer, yet it is understood that these S2S technologies are still developmental and are not completely polished.
- The *User Stakeholder* category would reasonably provide a "NV" relating to the *Autonomy Levels* for the Bravo test plan. Given that 1) the *TTLs* to be exercised in this test plan do not require any *Tech Users* to speak into and/or listen to the translations of the technology, 2) the evaluation is all done in software, and 3) no *Utility Assessment Metrics* are to be captured it's likely that the *Users* will not concern themselves with this test plan.

Table 38 - *Stakeholder Preferences* for *Tech User Autonomy Levels* for S2S Evaluation Planning

Stakeholder Preferences	MASTER GROUPING - ALPHA					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Autonomy Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	MEDIAN	AVERAGE	ST DEV
End-User (None)	NV	Abs Rej	Abs Rej	Neither	Abs Rej	Abs Rej	-3.75	2.50
End-User (Low)	NV	Slightly Pref	Reject	Strongly Pref	Slightly Rej	Slightly Rej	0.25	2.99
End-User (Med)	NV	Abs Pref	Abs Pref	Prefer	Prefer	Abs Pref	4.00	1.15
End-User (High)	NV	Prefer	Strongly Pref	Neither	Abs Pref	Strongly Pref	3.00	2.16
Stakeholder Preferences	MASTER GROUPING - BRAVO					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Autonomy Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	MEDIAN	AVERAGE	ST DEV
Tech Dev (None)	NV	Slightly Pref	Mod Pref	Abs Rej	NV	Slightly Pref	-0.67	3.79
Tech Dev (Low)	NV	Abs Pref	Abs Pref	Mod Rej	NV	Abs Pref	2.67	4.04
Tech Dev (Med)	NV	Slightly Rej	Neither	Mod Pref	NV	Neither	0.33	1.53
Tech Dev (High)	NV	Reject	Abs Rej	Abs Pref	NV	Reject	-1.00	5.29
Autonomy Levels - Environmental	Buyer	Eval Designer	Sponsor	Tech Dev	User	MEDIAN	AVERAGE	ST DEV
Tech Dev (None)	NV	Neither	Neither	Neither	NV	Neither	0	0
Tech Dev (Low)	NV	Neither	Neither	Neither	NV	Neither	0	0

- A majority of the *Stakeholders* would want the *Tech Developers* to have limited *Technical Autonomy* (i.e. “Low”) to prevent them from inserting bias into the evaluation. It’s plausible that the *Tech Developers* would lobby for greater *Technical Autonomy* here citing they want the flexibility to deal with issues in the evaluation as they arise.
- Lastly, there does not appear to be any need for the *Tech Developers* to have any *Environmental Autonomy* given that *TTLs* to be evaluated will be done so by software.

Some interesting observations can be made with respect to the output Majority Judgment and Evaluative Voting data:

- *Technical Autonomy Stakeholder Preferences* for “Medium” and “High” originally produced a tie of “Prefer” after one iteration of Majority Judgment. The second iteration of Majority Judgment produced “Medium” as ranking higher than “Low.” This finding is consistent with the Evaluative Voting scores.
- The *Technical Autonomy* of “Low” is “Slightly Rejected” by Majority Judgment yet has a very modest preferred score under Evaluative Voting. A plausible explanation for this discrepancy is the disagreement among the individual *Stakeholders* with respect to this element. This is a case where EV presents a different result than MJ based upon their respective methods of aggregation.

5.3.5. Personnel: Team Members and Participants: Knowledge Levels

Stakeholder Preferences relating to the *Knowledge Levels* of the *Team Members* and *Participants* are now captured. Table 39 presents the *Stakeholder Preferences* for the *Knowledge Levels* relating to the Alpha test plan.

Table 39 - *Stakeholder Preferences* for *Team Member* and *Participant Knowledge Levels* for S2S Evaluation Planning

Stakeholder Preferences	MASTER GROUPING - ALPHA					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Knowledge Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	MEDIAN	AVERAGE	ST DEV
Team Member (Low)	NV	Abs Pref	Prefer	Mod Pref	Abs Pref	Prefer	3.75	1.50
Participant (None)	NV	Strongly Pref	Strongly Pref	Slightly Rej	Abs Pref	Strongly Pref	3.00	2.71
Participant (Low)	NV	Slightly Pref	Slightly Rej	Prefer	Slightly Pref	Slightly Pref	1.00	1.63
Knowledge Levels - Operational	Buyer	Eval Designer	Sponsor	Tech Dev	User	MEDIAN	AVERAGE	ST DEV
Team Member (Low)	NV	Mod Rej	Mod Rej	Neither	Abs Rej	Mod Reject	-2.25	2.06
Team Member (Med)	NV	Strongly Pref	Mod Pref	Slightly Pref	Prefer	Mod Prefer	2.50	1.29
Team Member (High)	NV	Abs Pref	Strongly Pref	Prefer	Abs Pref	Strongly Pref	4.25	0.96
Participant (Low)	NV	Mod Pref	Slightly Rej	Neither	Slightly Rej	Slightly Rej	0.00	1.41
Participant (Med)	NV	Abs Pref	Abs Pref	Abs Pref	Abs Pref	Abs Pref	5.00	0.00

The Bravo test plan is not represented in this step since no *Team Members* or *Participants* were identified by the *Stakeholders* to be involved.

Before discussing the rationale behind these *Stakeholder Preferences* it is important to highlight one discrepancy between Majority Judgment and Evaluative Voting. Although these elements are not ranked against one another, the median grade and the average value for *Team Member (Low)* and *Participant (None)* show how the MJ and EV methods produce differing perspectives. *Team Member (Low)* has a median of "Prefer" and an EV score of 3.75 while *Participant (None)* has a lower EV score of 3.00 with a higher median of "Strongly Prefer." If these two elements are compared against one another, the results of MJ and EV would each express a different preference.

Some rationale behind the *Stakeholder Preferences* in Table 39 include:

- The only *Team Members* available have "Low" *Technical Knowledge.* If the *Stakeholders* had a strong negative preference to this lone option, then the *MRED Operator* would be forced to explore other alternatives (beyond his existing knowledge). The *MRED Operator* would contact either the *Evaluation Designer* or the *Sponsor* to obtain more resources.
- The *Stakeholders* prefer the *Participants* to have no ("None") *Technical Knowledge*. This is representative of how the technology would ultimately be deployed in its operational environment; *Tech Users* would attempt to engage foreign nationals (*Participants* for the evaluation).

- The *Operational Knowledge* of *Participants* ("Med") allows these *Personnel* to speak in a manner and with a vocabulary consistent to what the technology would encounter when deployed.

5.3.6. Personnel: Team Members and Participants: Autonomy Levels

Since the *Knowledge Levels* of the *Team Members* and *Participants* have been set, their *Autonomy Levels* can be determined. Table 40 presents the *Stakeholder Preferences* for the *Autonomy Levels* of these *Secondary Personnel*. Similar to Table 39, the Bravo test plan is not shown in Table 40 since it lacks *Team Members* and *Participants*. Some highlights of Table 40 include:

- MRED does not permit *Participants* any *Technical Autonomy* so this option is not presented to the *Stakeholders*
- *Team Members* are afforded little *Autonomy* by the *Stakeholders*
- *Participants* are provided with an *Environmental Autonomy* of "Med" enabling them to speak into the technology in a natural manner. This level of *Environmental Autonomy* also corresponds to the *Participants'* level of the *Operational Knowledge*. If the *Participants* had an *Operational Knowledge* of "High," it's doubtful the *Stakeholders* would have provided them an *Environmental Autonomy* of "High." This conclusion is drawn by the fact that the *Stakeholders* recognize that the technology is still under development and allowing the *Stakeholders* too much freedom in their dialogue could increase the chance of technology error.

Note that the outputs from Majority Judgment and Evaluative Voting from Table 40 are in agreement with respect to ordering of competing elements.

Table 40 - *Stakeholder Preferences* for *Team Member* and *Participant Autonomy Levels* for S2S Evaluation Planning

Stakeholder Preferences	MASTER GROUPING - ALPHA					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Autonomy Levels - Technical	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Team Member (None)	NV	Strongly Pref	Strongly Pref	Abs Pref	Mod Pref	Strongly Pref	**3.75**	**1.26**
Team Member (Low)	NV	Slightly Pref	Prefer	Mod Pref	Strongly Pref	Mod Pref	**2.50**	**1.29**
Participant (None)	NV	Abs Pref	Abs Pref	Mod Pref	Abs Pref	Abs Pref	**4.25**	**1.50**
Autonomy Levels - Environmental	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Team Member (None)	NV	Prefer	Slightly Pref	Neither	Mod Rej	Neither	**0.50**	**2.08**
Team Member (Low)	NV	Abs Pref	Prefer	Neither	Prefer	Prefer	**2.75**	**2.06**
Team Member (Med)	NV	Neither	Slightly Rej	Neither	Strongly Pref	Neither	**0.75**	**2.22**
Team Member (High)	NV	Reject	Reject	Neither	Mod Pref	Reject	**-1.00**	**2.45**
Participant (None)	NV	Strongly Rej	Abs Rej	Reject	Abs Rej	Abs Reject	**-4.25**	**0.96**
Participant (Low)	NV	Mod Pref	Mod Pref	Strongly Pref	Slightly Pref	Mod Pref	**2.25**	**1.26**
Participant (Med)	NV	Strongly Pref	Abs Pref	Slightly Pref	Abs Pref	Strongly Pref	**3.75**	**1.89**

5.3.7. Environment

The next step is to capture and handle the *Stakeholder Preferences* with respect to the test *Environments*. Table 41 presents the *Stakeholder Preferences* of *Environments* for both the Alpha and Bravo test plans. Some notes regarding the information presented in Table 41 include:

- Both the *Lab* and *Simulated Environments* are presented as options to the *Stakeholders* for both the Alpha and Bravo test plans. The Alpha test plan contains the *System* level *TTL*. Normally, if the *System* is to be evaluated without any other *TTLs*, then only the *Simulated* and *Actual Environments* would be candidates. However, the S2S *System* evaluation is coupled with several

Capabilities, so the union of *Environment* options are presented to the *Stakeholders*.

- The *Stakeholders* determine that both a *Lab* and a *Simulated Environment* that should be considered as candidates given their closeness in aggregate scores.
- The *Stakeholders* overwhelmingly reject the option of conducting an evaluation at the Aberdeen Proving Grounds.

Table 41 - ***Stakeholder Preferences*** **for** ***Environments*** **for S2S Evaluation Planning**

Stakeholder Preferences	**MASTER GROUPING - ALPHA**					**MAJORITY JUDGMENT**	**EVALUATIVE VOTING**	
Environments - Lab	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Conference Facilities	NV	Strongly Pref	Prefer	Abs Pref	Mod Pref	Prefer	**3.50**	**1.29**
Office Facilities	NV	Slightly Pref	Slightly Rej	Mod Rej	Strongly Rej	Mod Reject	**-1.50**	**2.08**
Environments - Simulated	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
NIST Test Facilities	NV	Slightly Pref	Abs Pref	Prefer	Prefer	Prefer	**3.75**	**0.96**
Aberdeen Proving Grounds	NV	Abs Rej	Abs Rej	Reject	Abs Pref	Abs Reject	**-2.00**	**4.76**
CANDIDATE TEST PLAN ELEMENTS - ALPHA								
TTL: Capabilities - English to Pashto, Pashto to English; System - YES								
Metrics: Technical Performance - HLCT								
Metrics: Utility Assessment - Ease of Use, PoF, FoEE, Likes, Dislikes, Changes (for all TTLs)								
Tech-User: End-User	Technical Knowledge - L			Operational Knowledge - H				
	Technical Autonomy - L			Environmental Autonomy - M				
Team Member	Technical Knowledge - L			Operational Knowledge - H				
	Technical Autonomy - N			Environmental Autonomy - L				
Participant	Technical Knowledge - L			Operational Knowledge - M				
	Technical Autonomy - N			Environmental Autonomy - M				
Stakeholder Preferences	**MASTER GROUPING - BRAVO**					**MAJORITY JUDGMENT**	**EVALUATIVE VOTING**	
Environments - Lab	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Conference Facilities	NV	Abs Pref	NV	Abs Pref	NV	Abs Pref	**5.00**	**0.00**
Office Facilities	NV	Mod Pref	NV	Neither	NV	Neither	**1.00**	**1.41**
Environments - Simulated	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
NIST Test Facilities	NV	Reject	NV	Strongly Rej	NV	Strongly Reject	**-3.5**	**0.71**
Aberdeen Proving Grounds	NV	Abs Rej	Abs Rej	Abs Rej	NV	Abs Reject	**-5.00**	**0.00**
CANDIDATE TEST PLAN ELEMENTS - BRAVO								
TTL: Components - ASR, MT; Capabilities - English to Pashto, Pashto to English								
Metrics: Technical Performance - Automated Metrics (Components)								
Metrics: Technical Performance - LLCT, Likert Scores (Capabilities)								
Tech-User: Tech-Dev	Technical Knowledge - H			Operational Knowledge - L				
	Technical Autonomy - L			Environmental Autonomy - N-L				

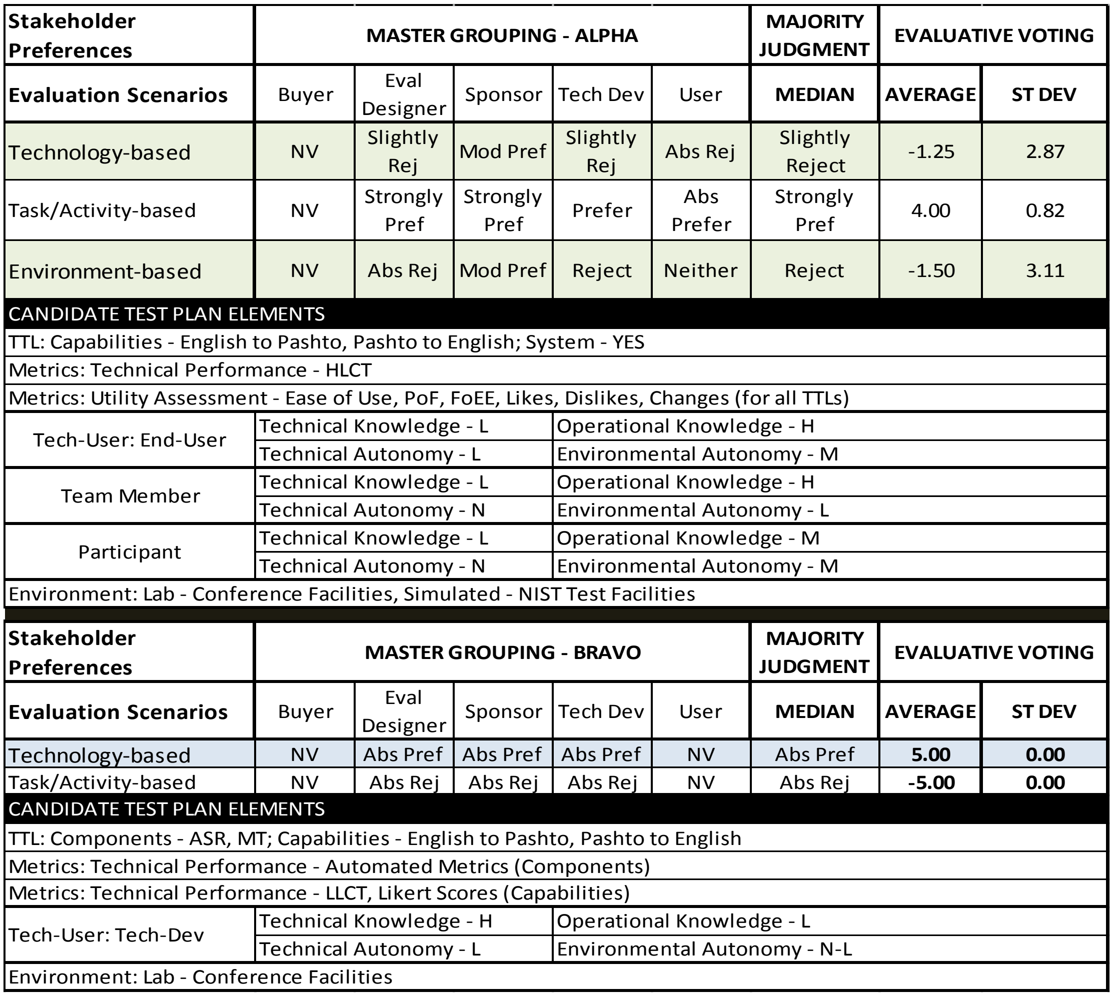
- Only the *Evaluation Designers* and the *Technology Developers* have an interest in determining the *Environment* to capture the data required in the Bravo test plan.
- No ordering discrepancies exist among competing alternatives between the outputs from MJ and EV.

- The median grade of “Absolutely Reject” for Aberdeen Proving Grounds (in *Simulated Environments* within Alpha) is somewhat inflated considering one of the four voting *Stakeholders* graded this element as “Absolutely Prefer.” This inflation is supported by the EV average of -2.00 and a standard deviation of 4.76 (which shows a high disagreement) among the voters.

5.3.8. Evaluation Scenarios

Evaluation Scenarios are determined according to the *Stakeholder Preferences*. Table 42 presents the *Stakeholder Preferences* for *Evaluation Scenarios,* the aggregate preference scores, and the existing state of the Alpha and Bravo test plans.

Table 42 - *Stakeholder Preferences* for *Evaluation Scenarios* for S2S Evaluation Planning



Stakeholder Preferences	MASTER GROUPING - ALPHA					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Evaluation Scenarios	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Technology-based	NV	Slightly Rej	Mod Pref	Slightly Rej	Abs Rej	Slightly Reject	-1.25	2.87
Task/Activity-based	NV	Strongly Pref	Strongly Pref	Prefer	Abs Prefer	Strongly Pref	4.00	0.82
Environment-based	NV	Abs Rej	Mod Pref	Reject	Neither	Reject	-1.50	3.11

CANDIDATE TEST PLAN ELEMENTS		
TTL: Capabilities - English to Pashto, Pashto to English; System - YES		
Metrics: Technical Performance - HLCT		
Metrics: Utility Assessment - Ease of Use, PoF, FoEE, Likes, Dislikes, Changes (for all TTLs)		
Tech-User: End-User	Technical Knowledge - L	Operational Knowledge - H
	Technical Autonomy - L	Environmental Autonomy - M
Team Member	Technical Knowledge - L	Operational Knowledge - H
	Technical Autonomy - N	Environmental Autonomy - L
Participant	Technical Knowledge - L	Operational Knowledge - M
	Technical Autonomy - N	Environmental Autonomy - M
Environment: Lab - Conference Facilities, Simulated - NIST Test Facilities		

Stakeholder Preferences	MASTER GROUPING - BRAVO					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Evaluation Scenarios	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Technology-based	NV	Abs Pref	Abs Pref	Abs Pref	NV	Abs Pref	**5.00**	**0.00**
Task/Activity-based	NV	Abs Rej	Abs Rej	Abs Rej	NV	Abs Rej	**-5.00**	**0.00**

CANDIDATE TEST PLAN ELEMENTS		
TTL: Components - ASR, MT; Capabilities - English to Pashto, Pashto to English		
Metrics: Technical Performance - Automated Metrics (Components)		
Metrics: Technical Performance - LLCT, Likert Scores (Capabilities)		
Tech-User: Tech-Dev	Technical Knowledge - H	Operational Knowledge - L
	Technical Autonomy - L	Environmental Autonomy - N-L
Environment: Lab - Conference Facilities		

In S2S test plans, the *Evaluation Scenario* types are generally defined as follows:

- *Technology-based* – Target a specific *Component* or *Capability* by directly feeding in specific speech or text to the S2S technology
- *Task/Activity-based* – Use the S2S technology to obtain and/or share specific information from a foreign speaker, etc.
- *Environment-based* – Employ the S2S technology as another tool to accomplish an overall mission within the *Environment*.

Pertinent notes relating to the information presented in Table 42 include:

- The *Stakeholders* strongly prefer *Task/Activity-Based* scenarios to *Technology-based* or *Environment-based* scenarios
- *Environment-based* scenarios are not a candidate for the Bravo test plan since a *Lab Environment* has already been designated as the test *Environment* (MRED states that *Environment-based Scenarios* can only be conducted in *Simulated* or *Actual Environments*).

5.3.9. Explicit Environmental Factors

The last step in the process before complete test plan blueprints are output is determining the *Stakeholder Preferences* for *Explicit Environmental Factors*. According to MRED, *Explicit Environmental Factors* are defined for the S2S test plans as the following:

- *Feature Density* – The quantity and distribution of artifacts in the *Environment* that will directly (e.g., sounds such as vehicles idling, weapons being fired, etc.) or indirectly (e.g. visual objects that influence the vocabulary used by the *Tech Users*, etc.)
- *Feature Complexity* – The intricacy of the artifacts in the *Environment*.

Feature Density and *Complexity* also apply to any data that is input into the technology for those test plans with software. The *MRED Operator* defines the *Density* as the overlapping of utterances while *Complexity* is the vocabulary used in and the length of an utterance. It is important to note that the *MRED Operator's* judgment is based upon informative interactions with the other *Stakeholders*. This enables the *MRED Operator* to create relevant test plans with MRED based upon the candidate technology.

Table 43 presents the *Stakeholders Preferences* for *Explicit Environmental Factors.* Some points to be called out from Table 43 include:

- The *Stakeholder Preference* averages are 2.75 for “Low” and 3.00 for “Medium” for *Feature Complexity* in the Alpha test plan. The difference of opinion is that *Technology Developers* want a “Low” *Feature Complexity* while the *Evaluation Designer* and *Sponsor* prefer “Medium.” In this case, the *MRED Operator* chooses to include both levels of *Feature Complexity* in the final Alpha test plan.
- *Feature Complexity* for the Bravo test plan has mixed *Preferences*, as well. The separation (0.67) between “Medium” and “High” causes the *MRED Operator* to consider both moving forward.
- Majority Judgment and Evaluative Voting are in agreement with respect to the ordering of competing elements.
- Majority Judgment and Evaluative Voting differ as to whether or not an element is preferred or rejected with respect to several elements. Note that these results did not impact the output blueprints since both methods identified a more preferred candidate. The difference in results of these two methods is found in:
    - Alpha – *Feature Density* (High) – Majority Judgment grades this element as “Moderately Reject” while Evaluative Voting scores it neutral with 0.
    - Alpha – *Feature Complexity* (High) – Majority Judgment grades this element as “Slightly Reject” while Evaluative Voting scores it positively with 0.50.

Table 43 - *Stakeholder Preferences* for *Explicit Environmental Factors* for S2S Evaluation Planning

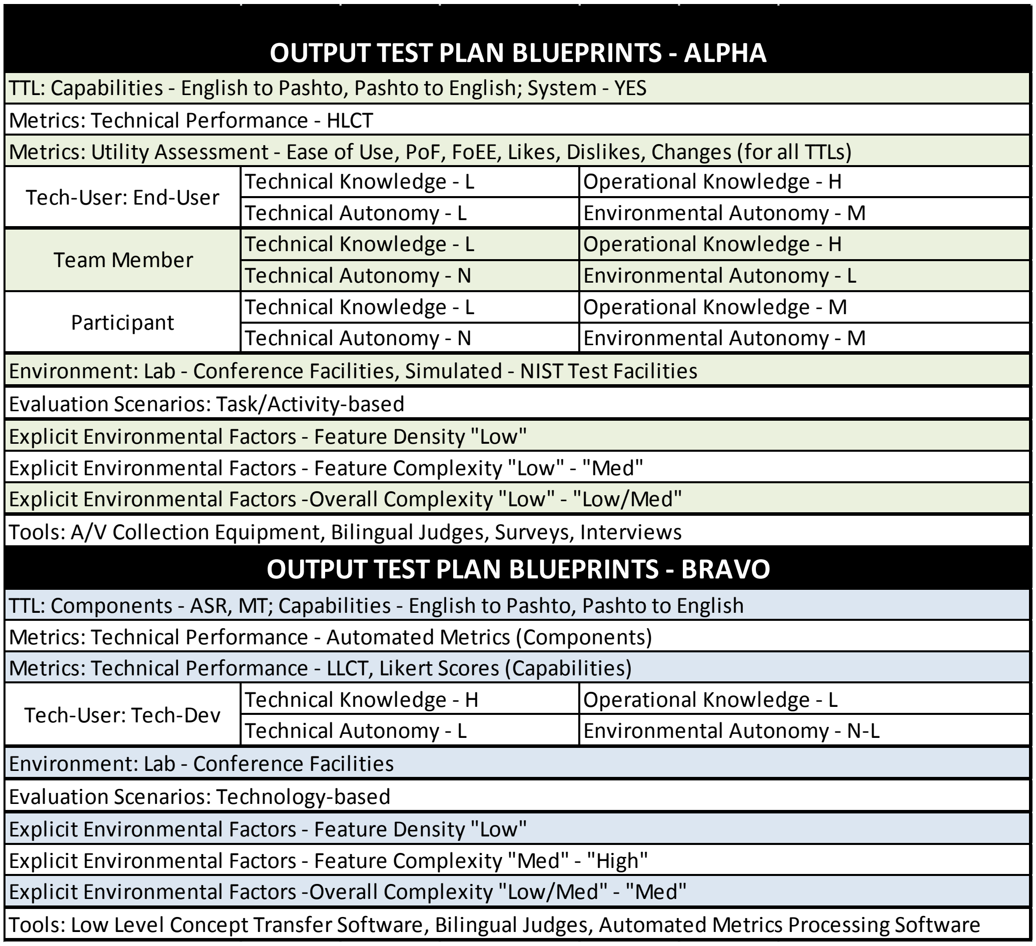
Stakeholder Preferences	MASTER GROUPING - ALPHA					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Explicit Environmental Factors - Feature Density	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Low	NV	Prefer	Abs Pref	Abs Pref	Neither	**Prefer**	**3.25**	**2.36**
Medium	NV	Mod Pref	Mod Pref	Neither	Prefer	**Mod Prefer**	**1.75**	**1.26**
High	NV	Mod Rej	Neither	Mod Rej	Strongly	**Mod Reject**	**0.00**	**2.83**
Explicit Environmental Factors - Feature Complexity	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Low	NV	Strongly Pref	Mod Pref	Abs Pref	Neither	**Mod Prefer**	**2.75**	**2.22**
Medium	NV	Prefer	Strongly Pref	Slightly Pref	Strongly	**Prefer**	**3.00**	**1.41**
High	NV	Slightly Rej	Neither	Mod Rej	Abs Pref	**Slightly Rej**	**0.50**	**3.11**
CANDIDATE TEST PLAN ELEMENTS								
TTL: Capabilities - English to Pashto, Pashto to English; System - YES								
Metrics: Technical Performance - HLCT								
Metrics: Utility Assessment - Ease of Use, PoF, FoEE, Likes, Dislikes, Changes (for all TTLs)								
Tech-User: End-User	Technical Knowledge - L			Operational Knowledge - H				
	Technical Autonomy - L			Environmental Autonomy - M				
Team Member	Technical Knowledge - L			Operational Knowledge - H				
	Technical Autonomy - N			Environmental Autonomy - L				
Participant	Technical Knowledge - L			Operational Knowledge - M				
	Technical Autonomy - N			Environmental Autonomy - M				
Environment: Lab - Conference Facilities, Simulated - NIST Test Facilities								
Evaluation Scenarios: Task/Activity-based								

Stakeholder Preferences	MASTER GROUPING - BRAVO					MAJORITY JUDGMENT	EVALUATIVE VOTING	
Explicit Environmental Factors - Feature Density	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**ST DEV**
Low	NV	Abs Pref	Strongly Pref	Abs Pref	NV	**Abs Prefer**	**4.67**	**0.58**
Medium	NV	Slightly Pref	Mod Pref	Slightly Rej	NV	**Slightly Pref**	**0.67**	**1.53**
High	NV	Abs Rej	Mod Rej	Abs Rej	NV	**Abs Reject**	**-4.00**	**1.73**
Explicit Environmental Factors - Feature Complexity	Buyer	Eval Designer	Sponsor	Tech Dev	User	**MEDIAN**	**AVERAGE**	**STD DEV**
Low	NV	Slightly Rej	Reject	Mod Pref	NV	**Slightly Rej**	**-0.67**	**2.52**
Medium	NV	Strongly Pref	Slightly Pref	Prefer	NV	**Prefer**	**2.67**	**1.53**
High	NV	Mod Pref	Abs Pref	Slightly Rej	NV	**Mod Pref**	**2.00**	**3.00**
CANDIDATE TEST PLAN ELEMENTS								
TTL: Components - ASR, MT; Capabilities - English to Pashto, Pashto to English								
Metrics: Technical Performance - Automated Metrics (Components)								
Metrics: Technical Performance - LLCT, Likert Scores (Capabilities)								
Tech-User: Tech-Dev	Technical Knowledge - H			Operational Knowledge - L				
	Technical Autonomy - L			Environmental Autonomy - N-L				
Environment: Lab - Conference Facilities								
Evaluation Scenarios: Technology-based								

5.3.10. Output Evaluation Blueprints

Table 44 presents the output blueprints. In this example, Majority Judgment and Evaluative Voting both lead to the same sets of blueprints.

Table 44 - Output Alpha and Bravo test plans for S2S technologies



OUTPUT TEST PLAN BLUEPRINTS - ALPHA		
TTL: Capabilities - English to Pashto, Pashto to English; System - YES		
Metrics: Technical Performance - HLCT		
Metrics: Utility Assessment - Ease of Use, PoF, FoEE, Likes, Dislikes, Changes (for all TTLs)		
Tech-User: End-User	Technical Knowledge - L	Operational Knowledge - H
	Technical Autonomy - L	Environmental Autonomy - M
Team Member	Technical Knowledge - L	Operational Knowledge - H
	Technical Autonomy - N	Environmental Autonomy - L
Participant	Technical Knowledge - L	Operational Knowledge - M
	Technical Autonomy - N	Environmental Autonomy - M
Environment: Lab - Conference Facilities, Simulated - NIST Test Facilities		
Evaluation Scenarios: Task/Activity-based		
Explicit Environmental Factors - Feature Density "Low"		
Explicit Environmental Factors - Feature Complexity "Low" - "Med"		
Explicit Environmental Factors -Overall Complexity "Low" - "Low/Med"		
Tools: A/V Collection Equipment, Bilingual Judges, Surveys, Interviews		
OUTPUT TEST PLAN BLUEPRINTS - BRAVO		
TTL: Components - ASR, MT; Capabilities - English to Pashto, Pashto to English		
Metrics: Technical Performance - Automated Metrics (Components)		
Metrics: Technical Performance - LLCT, Likert Scores (Capabilities)		
Tech-User: Tech-Dev	Technical Knowledge - H	Operational Knowledge - L
	Technical Autonomy - L	Environmental Autonomy - N-L
Environment: Lab - Conference Facilities		
Evaluation Scenarios: Technology-based		
Explicit Environmental Factors - Feature Density "Low"		
Explicit Environmental Factors - Feature Complexity "Med" - "High"		
Explicit Environmental Factors -Overall Complexity "Low/Med" - "Med"		
Tools: Low Level Concept Transfer Software, Bilingual Judges, Automated Metrics Processing Software		

These blueprints provide the *MRED Operator* and the *Stakeholders* with specific blueprints to serve as the backbone of detailed testing documentation and happen to align well with the implemented April 2010 test event. That test event can be categorized as having three major evaluation events:

- Offline Evaluations – Previously captured speech and text data that was fed into the S2S technologies to isolate the performance of the ASR and MT. Automated metrics, Low Level Concept Transfer and Likert Scores *Metrics* were captured. This evaluation aligns to the Bravo test plan.

- Lab Evaluations – Live evaluations of the S2S technologies in a controlled *Environment* where English and Pashto speakers conversed back and forth on specific tactical domains. High Level Concept Transfer and *Utility Assessment Metrics* were captured. This evaluation aligns to the Alpha test plan in the *Lab Environment*.
- Field Evaluations – Live evaluations of the S2S technologies in a "semi-controlled" outdoor setting at NIST where English and Pashto speakers conversed back and forth on tactical domains while manuevring around vehicles. High Level Concept Transfer and *Utility Assessment Metrics* were captured. This evaluation aligns to the Alpha test plan in the *Simulated Environment*.

Detailed test documents can be very different even when created from the same blueprints. Handing blueprints to different *Evaluation Designers* could lead to a variety of detailed approaches. Regardless of the approach employed and the detailed test characteristics, MRED provides a collective picture of what elements are available at the time of test consideration and what the *Stakeholders* prefer. This information is of great value as a technology undergoes further testing and/or matures during its development process.

5.4. Alternate Speech to Speech Evaluation Example

The April 2010 speech to speech evaluation is one of seven test events led by NIST personnel. Test plans, and their inputs, from the prior events are reviewed to see if MRED's treatment would be significantly different than that of the April 2010 evaluation. This review indicates that MRED would produce very similar results for prior test events as compared to the April 2010 evaluation. The January 2007 evaluation (NIST's first test event for this program) is used as the basis of comparison. Specifically, Table 45 presents the inputs for the January 2007 test event. These inputs are nearly identical to those presented in Table 30. The exception is that this evaluation focused on the technology developer's capability of translating between English and Iraqi as opposed to English and Pashto. Aside from this language difference, all other inputs are identical.

Table 45 – January 2007 Speech to Speech Evaluation Test Plan Input

Components	c =	Automatic Speech Recognition (C_1)		Machine Translation (C_2)		Text to Speech (C_3)	
# of Comp	τ = 3						
Capabilities	p =	English Transcription (P_1)		English to Iraqi Speech (P_2)		Iraqi to English Speech (P_3)	
# of Cap	ϕ = 3						
Technical Performance Metrics	t =	High Level Concept Transfer		Low Level Concept Transfer		Likert Scores	Automated Metrics
# of Tech	α = 4						
Utility Assessment Metrics	a =	Ease of Use	Perception of Functionality	Feedback on Encountered Errors	What Users Liked	What Users did not Like	What Users would Change
# of Util	β =6						

One other significant difference is that the *Stakeholders* preferred to have the *Technology Users* assess the capability of "English Transcription." This capability would be included in the "Alpha" blueprints where the *Technology Users* would have provided their feedback in the form of the *Utility Assessment Metrics* already noted. Recall that this capability was rejected by the *Stakeholders* for the April 2010 evaluation. Besides the language difference and the inclusion of another capability, MRED's generation of these blueprints would be virtually identical. MRED would have produced very different blueprints had test plans been examined prior to NIST's involvement. Prior to 2007 and before NIST's involvement, another organization led the test events of S2S technologies. The technologies were much less mature at this time and evaluations took a drastically different form. The prior test events were largely based in software and/or feature focused individual assessments of the components and capabilities listed (shown in Table 45) whereas NIST test events were dominated by English and foreign language speakers carrying on conversations in multiple languages.

5.5. Summary

This Chapter successfully validates the MRED approach using an existing technology in accordance with the first research question. Applying MRED to the design of speech to speech translation technology test plans has yielded two unique evaluation blueprints that can reasonably produce the specific test characteristics that defined the April 2010 S2S evaluations. These blueprints represent the available *TTLs* for evaluation, the desired *Metrics,* available *Resources*, *Stakeholder Preferences*, and practical relationships and constraints encountered within test planning. In situations where *Stakeholder Preferences* did not find a clear, preferred option, the MRED Operator used their discretion to pass multiple options through for an evaluation element. This flexibility (e.g., allowing multiple *Environments* to be considered for the Alpha test plan, multiple *Feature Complexity* values to be considered for the Bravo test plan, etc.) ensures that candidate and preferred options are not prematurely discounted.

The MRED process also provides traceability of available test plan elements and *Stakeholder Preferences*. This enables the *MRED Operator* and the other

Stakeholders to track changing availabilities and preferences over time. This is a crucial benefit given that many of these advanced and emerging technologies are evaluated on multiple occasions during their development cycle.

MRED displays tremendous value when applied to the S2S technology with respect to capturing and handling *Stakeholder Preferences*. The fact that MRED did not have a significant impact on the available S2S blueprint elements is explained in that all of the S2S *TTLs* were mature enough to have one or more corresponding *Metrics* produced. Had any of the *TTLs* been too immature to support any *Metrics*, MRED's constraint handling would have eliminated those blueprint elements that were not necessary or unavailable. MRED's use of relationships to handle constraints and eliminate candidate blueprint elements would have been more prominent had an early evaluation of the S2S technologies been explored. Over five formal evaluations of the S2S technologies had occurred prior to the April 2010 test event. However, little data exists on the early evaluation test plans for which not all of the *TTLs* were available for testing.

MRED's diversity is shown in the S2S blueprint generation since the technology is at a different level of fitness as compared to the robot arm example illustrated in Chapter 4. Altogether, MRED is capable of generating test plan blueprints whether a technology is still in its infancy (e.g., very few functional *TTLs*) or if it's close to deployment (e.g., all *TTLs* are at least functional).

Chapter 6: Preference Handling Exploration

Capturing and handling *Stakeholder Preferences* is a non-trivial process. Care must be taken to ensure that the preferences captured are used to produce a meaningful group decision. This chapter is motivated by the differences between Majority Judgment and Evaluative Voting presented in Chapters 4 and 5.

A strategy to addressing preference handling is developed to guide the dissertation author in selecting the most appropriate method for integration with MRED. The strategy focuses on satisficing the following:

- *No Voter Quantity Restrictions* – Account for the preferences of multiple *Stakeholders*, yet not be locked into a method that restricts the quantity of *Stakeholders.*
- *Independent of Irrelevant Alternatives* - Capture preferences of alternatives such that they are still valid if other alternatives are added and/or subtracted.
- *Preference Longevity* – Capture preferences of alternatives such that comparisons can be made of preferences of the same alternative from one evaluation to the next.
- *Strength of Preference* – Delineate strength of preference.
- *Abstention* – Enable *Stakeholders* to abstain from voting.

This chapter will present evidence on how Majority Judgment and Evaluative Voting have satisfied these five factors as compared to several other preference capture and handling methods mentioned in Section 2.3. Several instances of disagreement between Majority Judgment and Evaluative Voting will also be discussed. An ordinal ranking method, the Borda Count, will be examined to see how their methods and results compare to that of Majority Judgment and Evaluative Voting. In addition, these methods will also be reviewed against the five points outlined in the strategy.

The first section of this chapter highlights some differences between Majority Judgment and Evaluative Voting. This section also presents the responses of these two methods to the five criteria outlined above. The second section outlines the implementation of the Borda Count with MRED. The third section will apply the Borda Count to capturing and handling *Stakeholder Preferences* for the S2S technologies where the results will be compared to the output from Evaluative Voting. The fourth section will apply the 5-point Evaluative Voting to see how its results compare to the 11-point scale used in MRED. The fifth section will conclude the chapter.

6.1. Majority Judgment v. Evaluative Voting

Table 46 presents an example where five voters express their preferences for two alternatives. Both Majority Judgment (MJ) and Evaluative Voting (EV) produce the same ranking; Alternative B is preferred over Alternative A. The minimum (red highlighting) and maximum (green highlighting) preferences for each alternative are identified (in Table 46) in addition to the median, mean, and standard deviation. The min and max values are included to augment the median data thereby providing the MRED Operator with richer information to make their decision.

Table 46 - Majority Judgment (Median) vs. Evaluative Voting (Mean) with level of Stakeholder Agreement (Standard Deviation) – Practical Example 1

ALTERNATIVE	STAKEHOLDER PREFERENCES							
	Voter #1	Voter #2	Voter #3	Voter #4	Voter #5	MEDIAN	MEAN	ST DEV
Alternative A	Neither	Abs Pref	Prefer	Mod Pref	Abs Pref	**Prefer**	**3.00**	**2.12**
Alternative B	Abs Pref	Strongly Pref	Strongly Pref	Slightly Reject	Abs Pref	**Strongly Pref**	**3.40**	**2.51**

Table 47 highlights how two alternatives are ranked differently according to Majority Judgment and Evaluative Voting. The median scores with Majority Voting show that Alternative B is equally preferred to Alternative A. However, the maximum preference for both alternatives is “Absolutely Prefer.” The minimum preference varies between the two alternatives; the lowest preference for Alternative A is “Moderately Prefer” while the lowest preference for Alternative B is “Slightly Reject” (which also happens to be the only negative preference). Each of the voters has a relatively positive impression of Alternative A; the same cannot be said for Alternative B. The mean and standard deviation data presented from Evaluative Voting show a different situation; this method indicates that Alternative A is preferred over Alternative B. The preference of A over B is clear (according to Evaluative Voting) in that the mean is greater (3.75 for A as compared to 3.00 for B) and there is more agreement among the voters (standard deviation of A is 1.50 while the standard deviation of B is 2.71).

Table 47 - Majority Judgment (Median) vs. Evaluative Voting (Mean) with level of Stakeholder Agreement (Standard Deviation) – Practical Example 2

ALTERNATIVE	STAKEHOLDER PREFERENCES							
	Voter #1	Voter #2	Voter #3	Voter #4	Voter #5	MEDIAN	MEAN	ST DEV
Alternative A	NV	Abs Pref	Prefer	Mod Pref	Abs Pref	**Prefer**	**3.75**	**1.50**
Alternative B	NV	Strongly Pref	Strongly Pref	Slightly Reject	Abs Pref	**Strongly Pref**	**3.00**	**2.71**

Now suppose that Voter #1 felt they were now more informed on the alternatives and decided to vote. Table 48 presents the output of this change.

Table 48 - Majority Judgment (Median) vs. Evaluative Voting (Mean) with level of Stakeholder Agreement (Standard Deviation) – Practical Example 3

ALTERNATIVE	STAKEHOLDER PREFERENCES							
	Voter #1	Voter #2	Voter #3	Voter #4	Voter #5	MEDIAN	MEAN	ST DEV
Alternative A	Abs Reject	Abs Pref	Prefer	Mod Pref	Abs Pref	**Prefer**	**2.00**	**4.12**
Alternative B	Mod Reject	Strongly Pref	Strongly Pref	Slightly Reject	Abs Pref	**Strongly Pref**	**2.00**	**3.24**

The output from Majority Judgment does not change; Alternative B is still preferred to Alternative A. The minimum and maximum data offers some insight. Although both Alternatives have the same maximum, Alternative A's minimum is much lower than Alternative B's ("Absolutely Reject" v. "Moderaly Reject"). Evaluative Voting tells a different story where the means are equal, indicating that these two alternatives are equally preferred. The standard deviation proves useful and highlights that the stakeholders are in greater agreement as to their preference of Alternative B as compared to Alternative A since B has a lower standard deivation than A.

Now suppose that Alternative C is introduced and each voter is allowed to convey their preferences (see Table 49). All five voters provide positive grades for Alternative C; whereas Alternatives A and B each continue to receive one or more negative grades. Alternative C is considerd the most preferred according to Evaluative Voting (greatest mean) and displays the most voter agreement (lowest standard deviation). However, Alternative C is graded the lowest by Majority Judgment. Again, the minimum data offers some value while the maximum data does not. Alternative A has the lowest minimum with "Absolutely Reject" while

Alternative C has the highest minimum of “Neither.” All of the maximum data is equal.

Table 49 - Majority Judgment (Median) vs. Evaluative Voting (Mean) with level of Stakeholder Agreement (Standard Deviation) – Practical Example 4

ALTERNATIVE	STAKEHOLDER PREFERENCES							
	Voter #1	Voter #2	Voter #3	Voter #4	Voter #5	MEDIAN	MEAN	ST DEV
Alternative A	Abs Reject	Abs Pref	Prefer	Mod Pref	Abs Pref	**Prefer**	**2.00**	**4.12**
Alternative B	Mod Reject	Strongly Pref	Strongly Pref	Slightly Reject	Abs Pref	**Strongly Pref**	**2.00**	**3.24**
Alternative C	Mod Pref	Neither	Mod Pref	Abs Pref	Abs Pref	**Mod Pref**	**2.80**	**2.17**

The example presented in Table 46 has shown the two methods (MJ and EV) yielding similar results (whether the same candidates are above a certain threshold or whether the same candidate is most preferred among its peers). Likewise, instances are also documented (see examples presented in Table 47, Table 48, and Table 49) where these two methods present conflicting results. Majority Judgment provides the MRED Operator with a single piece of information, the median, for when a decision must be made with respect to two or more candidates that score “close” to one another. Identifying the minimum and maximum data offers the MRED Operator a richer information set with which to base their decisions, yet this appears insufficient in some instances. Evaluative voting offers the MRED Operator the mean and standard deviation. This gives the MRED Operator an understanding of the overall preference (mean) for an alternative in addition to the relative level of agreement (standard deviation) among the stakeholders.

The five criteria that were presented at the onset of this Chapter are discussed with respect to Majority Judgment and Evaluative Voting:

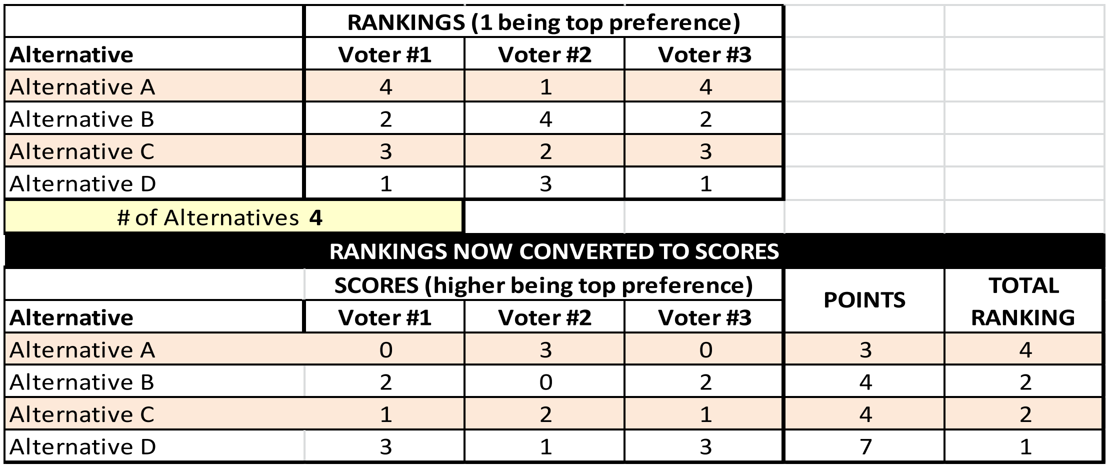
- *No Voter Quantity Restrictions* – Neither Majority Judgment nor Evaluative Voting restricts the quantity of voters that can express their preferences on a given set of Alternatives
- *Independence of Irrelevant Alternatives* – Majority Judgment and Evaluative Voting support new alternatives being added or existing alternatives subtracted without impacting the preferences of the original (or remaining) alternatives.
- *Preference Longevity* – Both methods support the comparison of an alternative's preferences across multiple test events or points in time. Additional information is not required to make these comparisons. This is supported by the *Independence of Irrelevant Alternatives*.
- *Strength of Preference* – Both aggregation methods of Majority Judgment and Evaluative Voting support strength of preference; strength of preference allows each stakeholder to express varying degrees of preference or rejection for an alternative.
- *Abstention* – Both aggregation methods support a voter's ability to not vote for an alternative without negatively or positively impacting its overall aggregate preference.

6.2. Borda Count Implementation

The Borda Count enables each decision-maker to rank (or vote) on each candidate alternative (Saari, 1990; Saari, 2006). Suppose the Borda Count method were used in place of the 11-point ordinal scale to capture *Stakeholder Preferences*. Likewise, the Borda Count would be used to aggregate preferences instead of

Majority Judgment or Evaluative Voting. Instead of asking the *Stakeholders* to rate each test plan element on a scale from "Absolutely Reject" to "Absolutely Prefer", they are asked to rank each element from 1 to the total number of elements. *Stakeholders* would not be allowed to waive ranking any element. Table 50 applies the Borda Count to four alternative *TTL-Metric* pairs. For simplicity in explaining this method, only three *Stakeholders* cast votes. A *Stakeholder* only has to rank each alternative once (i.e. if there are 4 alternatives, the *Stakeholder* ranks them from one to four). Since there are four total alternatives, the *Stakeholders* rate each *TTL-Metric* pair with their most preferred pair given a "1," the next most preferred given a "2," etc. Once all of the *Stakeholders* have completed their rankings, these numbers are turned into scores by subtracting each ranking from the total # of alternatives. For example, the Alternative A is rated "4" by the *Evaluation Designer*. The corresponding score would be 4 (total # of *TTL-Metric* pairs) – 4 (specific *TTL-Metric* pair score) = 0. This process is repeated for all ratings and the ratings of each *TTL-Metric* pair are summed. These sums are "Points" and the *TTL-Metric* pair with the greatest points is the "most preferred." The scores shown in Table 50 indicate the order of preference to be D ≻ B = C ≻A (i.e., D is preferred to B; B and C are equally preferred; and C is preferred to A). What this table does not show is which of the *TTL-Metric* pairs the *Stakeholders* prefer to be evaluated and which pairs they feel should not be evaluated.

Table 50 – Implementation of Borda Count example

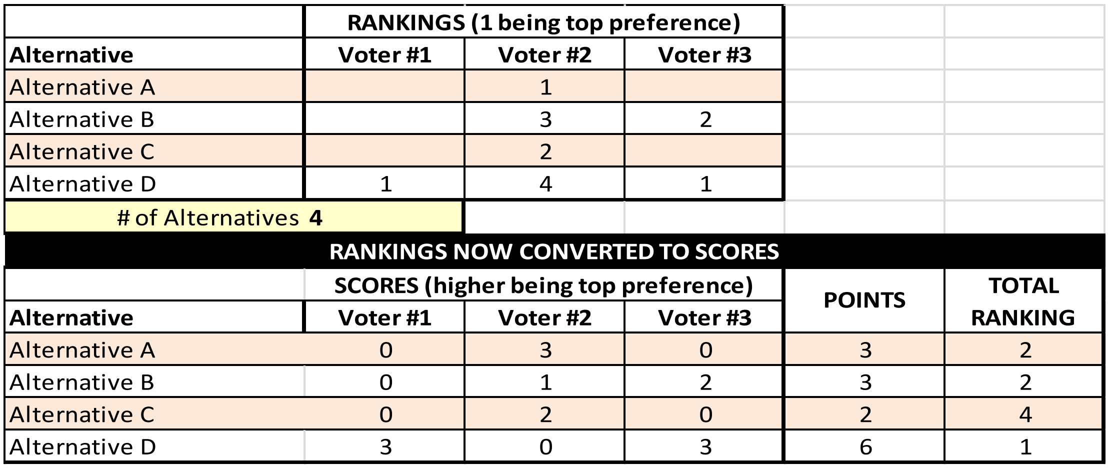


	RANKINGS (1 being top preference)				
Alternative	Voter #1	Voter #2	Voter #3		
Alternative A	4	1	4		
Alternative B	2	4	2		
Alternative C	3	2	3		
Alternative D	1	3	1		
# of Alternatives **4**					
RANKINGS NOW CONVERTED TO SCORES					
	SCORES (higher being top preference)			POINTS	TOTAL RANKING
Alternative	Voter #1	Voter #2	Voter #3		
Alternative A	0	3	0	3	4
Alternative B	2	0	2	4	2
Alternative C	1	2	1	4	2
Alternative D	3	1	3	7	1

A significant piece of information that is lacking from the Borda scores in Table 50 is that it is unknown which alternatives the *Stakeholders* prefer to see evaluated and which they do not. Alternative D may have been ranked low either because the *Stakeholders* are against evaluating this alternative; or the *Stakeholders* believe it should be evaluated, yet not at the priority level of the other alternatives. It's possible that any or a combination of the *Stakeholders* prefer ALL of the alternatives be evaluated or a subset of them. Likewise, the strength of preference among alternatives is unknown. The *Evaluation Designer* ranks the alternatives in preference order D ≻ B ≻ C ≻A in Table 50. However, it's plausible that the *Evaluation Designer* feels B = C, yet was unable to express this equality. Another scenario could be that the *Evaluation Designer* believes D ≻≻ B. Unfortunately, the Borda Count does not provide the *Stakeholders the* ability to express this strength of preference.

Now suppose that not all of the *Stakeholders* rated all of the *TTL-Metric* pairs as is allowed by MRED. Some *Stakeholders* might feel they did not have enough information to make an informed decision on these *TTL-Metric* pairs. Or, there were so many alternatives to choose from, *Stakeholders* only ranked those that they believed to be the most important. Table 51 presents partial rankings where not all of the *TTL-Metric* pairs are rated by each *Stakeholder*. Specifically, one *Stakeholder* rated only one alternative, another *Stakeholder* rated two alternatives while the remaining *Stakeholder* rated all four pairs.

Table 51 – Implementation of Borda Count example with Partial Rankings

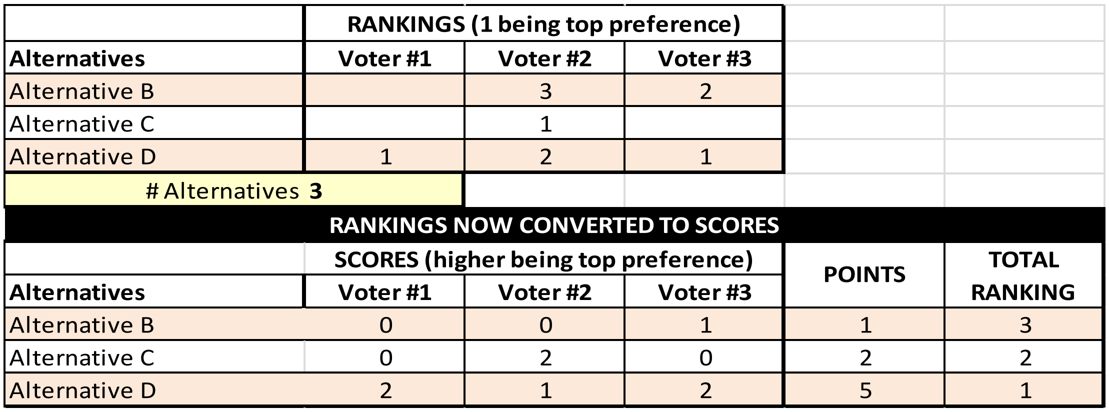


	RANKINGS (1 being top preference)				
Alternative	**Voter #1**	**Voter #2**	**Voter #3**		
Alternative A		1			
Alternative B		3	2		
Alternative C		2			
Alternative D	1	4	1		
# of Alternatives **4**					
RANKINGS NOW CONVERTED TO SCORES					
	SCORES (higher being top preference)			**POINTS**	**TOTAL RANKING**
Alternative	**Voter #1**	**Voter #2**	**Voter #3**		
Alternative A	0	3	0	3	2
Alternative B	0	1	2	3	2
Alternative C	0	2	0	2	4
Alternative D	3	0	3	6	1

Comparing the results of Table 50 and Table 51 shows a change in preference ordering. What is originally D ≻ B = C ≻ A in Table 50 is now D ≻ A = B ≻ C in Table 51. A *Stakeholder's* decision to abstain from voting in the Borda Count, for whatever reason, can impact the preference ordering; the example in Table 51 demonstrated that an alternative tied for second can move to fourth (last).

Now remove Alternative A from consideration in Table 51. The result of removing Alternative A is D ≻ C ≻ B, shown in Table 52. Alternative D has always been the most preferred choice, yet C and B continue to switch places. First, B = C; then B ≻ C when partial ratings were allowed; now C ≻ B when an alternative is removed.

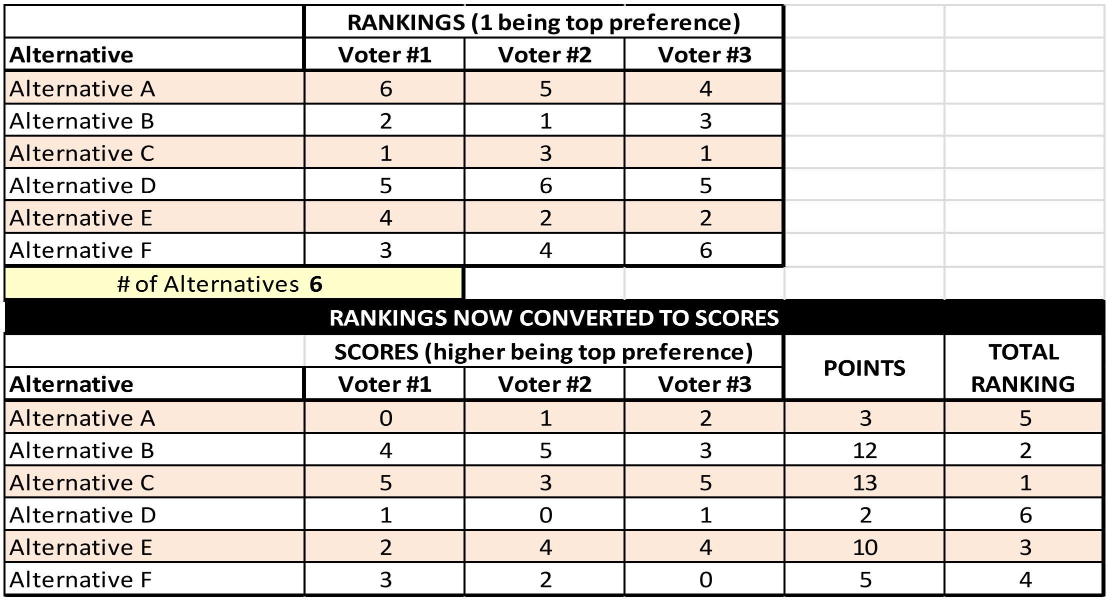
Table 52 - Implementation of Borda Count example with Partial Rankings and a Removed Alternative



	RANKINGS (1 being top preference)				
Alternatives	**Voter #1**	**Voter #2**	**Voter #3**		
Alternative B		3	2		
Alternative C		1			
Alternative D	1	2	1		
# Alternatives **3**					
RANKINGS NOW CONVERTED TO SCORES					
	SCORES (higher being top preference)			**POINTS**	**TOTAL**
Alternatives	**Voter #1**	**Voter #2**	**Voter #3**		**RANKING**
Alternative B	0	0	1	1	3
Alternative C	0	2	0	2	2
Alternative D	2	1	2	5	1

Let's go back to the original Borda rankings (D ≻ B = C ≻A) for Alternatives A, B, C, and D where all *Stakeholders* voted (Table 50) and assume these are the *TTL-Metric* pairs being considered for evaluation in year one of a program. Now, explore the case where an evaluation is being considered for year two where Alternatives A, B, C, and D are available and E and F are added. Table 53 shows the Borda Rankings and Scores for these six alternatives.

Table 53 - Implementation of Borda Count example with Partial Rankings and Added Alternatives



	RANKINGS (1 being top preference)				
Alternative	**Voter #1**	**Voter #2**	**Voter #3**		
Alternative A	6	5	4		
Alternative B	2	1	3		
Alternative C	1	3	1		
Alternative D	5	6	5		
Alternative E	4	2	2		
Alternative F	3	4	6		
# of Alternatives **6**					
RANKINGS NOW CONVERTED TO SCORES					
	SCORES (higher being top preference)			**POINTS**	**TOTAL**
Alternative	**Voter #1**	**Voter #2**	**Voter #3**		**RANKING**
Alternative A	0	1	2	3	5
Alternative B	4	5	3	12	2
Alternative C	5	3	5	13	1
Alternative D	1	0	1	2	6
Alternative E	2	4	4	10	3
Alternative F	3	2	0	5	4

Recall that Table 50 produces the preference ordering of D ≻ B = C ≻A. However, Table 53 presents the preference ordering of C ≻ B ≻ E ≻ F ≻ A ≻ D. This table not only highlights the *Stakeholders* change in preferences from year one to year two of a technology development cycle, it also indicates that *Stakeholder Preference* data collected during year one cannot be compared to the data collected during year two. *Stakeholder Preference* data would remain valid in this situation when Evaluative Voting is used.

This simple example highlights many concerns with implementing the Borda Count within MRED as the means to capture and handle *Stakeholder Preferences*. Specifically, the following statements can be made about the Borda Count as it relates to the five strategy:

- *No Voter Quantity Restrictions* – Enables an infinite amount of voters to express their preferences.
- *Independent of Irrelevant Alternatives* – Does NOT maintain the validity of previously-collected alternative preferences if other alternatives are added and/or subtracted
- *Preference Longevity* – Does NOT enable comparison of preferences of the same alternative from one evaluation to the next
- *Strength of Preference* – Does NOT delineate strength of preference
- *Abstention* – Enables *Stakeholders* to abstain from voting

The Borda Count does not satisfy all of the criteria after the examination conducted in this section. An additional concern with Borda Count is that it is a ranking method. This is in contrast to the ordinal linguistic scale that is used to capture preferences which can be classified as ratings. Further research has shown that "…rankings may impose a somewhat artificial contrast on the data…" (Alwin & Krosnick, 1985). This is supported by ranking methods forcing negatively-correlated preferences as opposed to rating approaches which foster positive correlations. Ranking requires increased cognitive demand because a decision-maker must compare each alternative to every other alternative, while rating an alternative is done using a scale (regardless of the other alternatives). The added mental effort is another reason that the Borda Count is not a reasonable candidate for integration with MRED.

6.3. Alternative Scales

Both Majority Judgment and Evaluative Voting can be performed on scales other than the chosen 11-point linguistic ordinal scale. This section presents a brief study of implementing the 5-point scale for the sake of reducing *Stakeholder* effort. A five point scale has been used in several prior efforts with success (Alwin & Krosnick, 1985; Hillinger, 2007; National Opinion Research Center, 1982). The 5-point scale is defined as (Absolutely Reject, Reject, Neither Prefer nor Reject, Prefer, Absolutely Prefer). Let's use the same *TTL-Metric* pair *Stakeholder Preferences* noted in Chapter 5 and convert them to the 5-point scale. For consistency in the conversion, the following mapping of ratings is devised in Table 54.

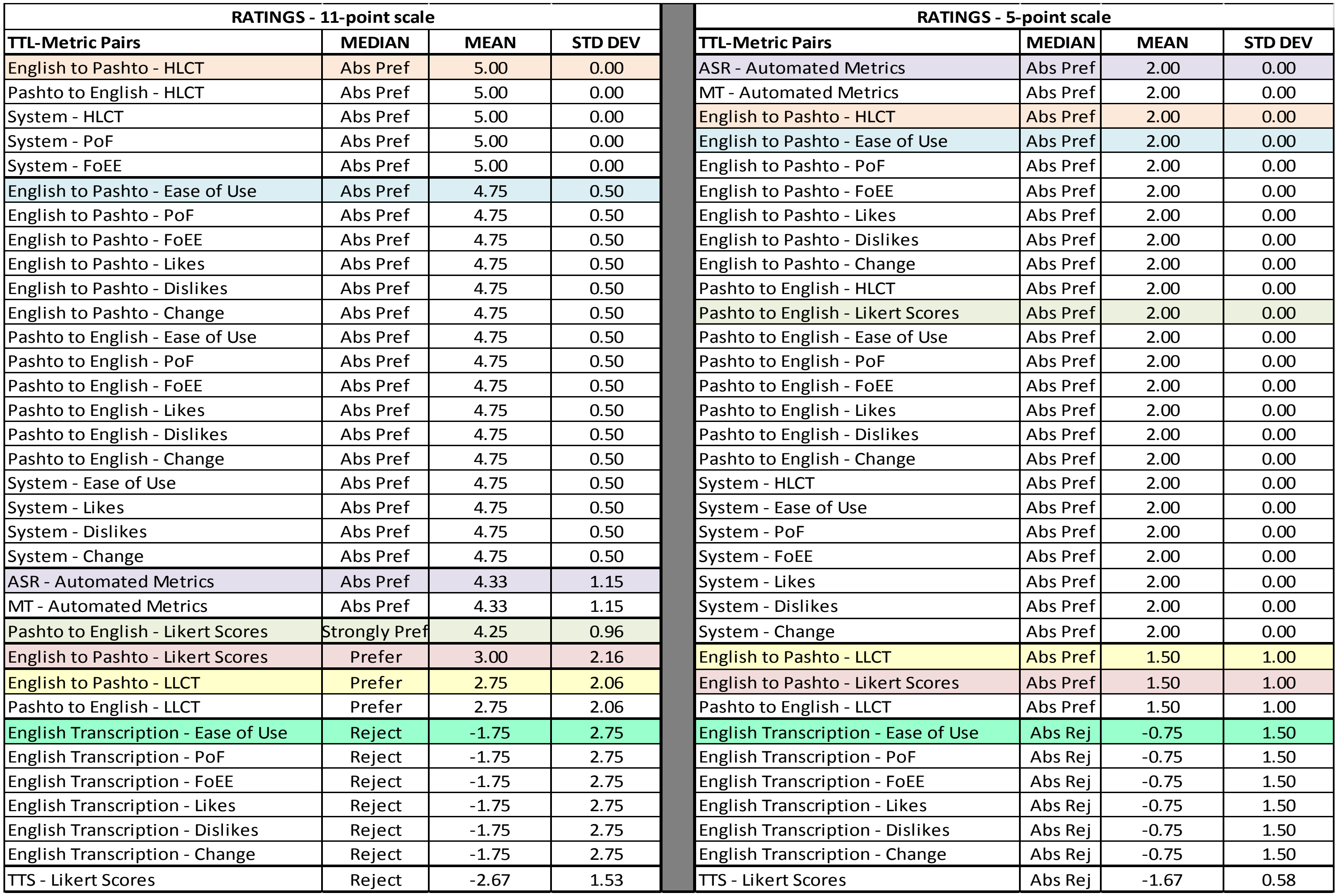
Table 54 – 11-point to 5-point Scale Mapping

LINGUISTIC SCALE MAPPING (Interval Transformation)	
11-point Scale	**5-point Scale**
Absolutely Reject (-5)	Absolutely Reject (-2)
Strongly Reject (-4)	Absolutely Reject (-2)
Reject (-3)	Absolutely Reject (-2)
Moderately Reject (-2)	Reject (-1)
Slightly Reject (-1)	Reject (-1)
Neither Prefer nor Reject (0)	Neither Prefer nor Reject (0)
Slightly Prefer (1)	Prefer (1)
Moderately Prefer (2)	Prefer (1)
Prefer (3)	Absolutely Prefer (2)
Strongly Prefer (4)	Absolutely Prefer (2)
Absolutely Prefer (5)	Absolutely Prefer (2)

Table 55 presents Evaluative Voting scores of the *Stakeholder Preferences* for the S2S candidate *TTL-Metric* pairs presented in Chapter 5. These scores are presented on both the 11-point and 5-point scales (according to the mapping defined in Table 54). Several pairs have been colored in Table 55 to highlight the difference in overall ordering according to the total scores according to Evaluative Voting. The median data produced by Majority Judgment does not present any shift in the overall orderings.

Note that the coarser, 5-point scale has much less dispersion among the pairs. This is particulary evident in the median data. Likewise, there is much less disagreement among the *Stakeholders* in the 5-point scale (lower standard deviations). Like the 11-point scale, the 5-point scale does indicate which *TTL-Metric* pairs should be considered for evaluation and which should not. The *MRED Operator* has a more challenging decision to make with the data with the 5-point scale; here 24 pairs are ranked in the top tier. However, the *MRED Operator* can easily see that 5 pairs are ranked above the other candidates on the 11-point scale.

Table 55 - 11-point and 5-point scale comparison with S2S TTL-Metric pairs



RATINGS - 11-point scale				RATINGS - 5-point scale			
TTL-Metric Pairs	MEDIAN	MEAN	STD DEV	TTL-Metric Pairs	MEDIAN	MEAN	STD DEV
English to Pashto - HLCT	Abs Pref	5.00	0.00	ASR - Automated Metrics	Abs Pref	2.00	0.00
Pashto to English - HLCT	Abs Pref	5.00	0.00	MT - Automated Metrics	Abs Pref	2.00	0.00
System - HLCT	Abs Pref	5.00	0.00	English to Pashto - HLCT	Abs Pref	2.00	0.00
System - PoF	Abs Pref	5.00	0.00	English to Pashto - Ease of Use	Abs Pref	2.00	0.00
System - FoEE	Abs Pref	5.00	0.00	English to Pashto - PoF	Abs Pref	2.00	0.00
English to Pashto - Ease of Use	Abs Pref	4.75	0.50	English to Pashto - FoEE	Abs Pref	2.00	0.00
English to Pashto - PoF	Abs Pref	4.75	0.50	English to Pashto - Likes	Abs Pref	2.00	0.00
English to Pashto - FoEE	Abs Pref	4.75	0.50	English to Pashto - Dislikes	Abs Pref	2.00	0.00
English to Pashto - Likes	Abs Pref	4.75	0.50	English to Pashto - Change	Abs Pref	2.00	0.00
English to Pashto - Dislikes	Abs Pref	4.75	0.50	Pashto to English - HLCT	Abs Pref	2.00	0.00
English to Pashto - Change	Abs Pref	4.75	0.50	Pashto to English - Likert Scores	Abs Pref	2.00	0.00
Pashto to English - Ease of Use	Abs Pref	4.75	0.50	Pashto to English - Ease of Use	Abs Pref	2.00	0.00
Pashto to English - PoF	Abs Pref	4.75	0.50	Pashto to English - PoF	Abs Pref	2.00	0.00
Pashto to English - FoEE	Abs Pref	4.75	0.50	Pashto to English - FoEE	Abs Pref	2.00	0.00
Pashto to English - Likes	Abs Pref	4.75	0.50	Pashto to English - Likes	Abs Pref	2.00	0.00
Pashto to English - Dislikes	Abs Pref	4.75	0.50	Pashto to English - Dislikes	Abs Pref	2.00	0.00
Pashto to English - Change	Abs Pref	4.75	0.50	Pashto to English - Change	Abs Pref	2.00	0.00
System - Ease of Use	Abs Pref	4.75	0.50	System - HLCT	Abs Pref	2.00	0.00
System - Likes	Abs Pref	4.75	0.50	System - Ease of Use	Abs Pref	2.00	0.00
System - Dislikes	Abs Pref	4.75	0.50	System - PoF	Abs Pref	2.00	0.00
System - Change	Abs Pref	4.75	0.50	System - FoEE	Abs Pref	2.00	0.00
ASR - Automated Metrics	Abs Pref	4.33	1.15	System - Likes	Abs Pref	2.00	0.00
MT - Automated Metrics	Abs Pref	4.33	1.15	System - Dislikes	Abs Pref	2.00	0.00
Pashto to English - Likert Scores	Strongly Pref	4.25	0.96	System - Change	Abs Pref	2.00	0.00
English to Pashto - Likert Scores	Prefer	3.00	2.16	English to Pashto - LLCT	Abs Pref	1.50	1.00
English to Pashto - LLCT	Prefer	2.75	2.06	English to Pashto - Likert Scores	Abs Pref	1.50	1.00
Pashto to English - LLCT	Prefer	2.75	2.06	Pashto to English - LLCT	Abs Pref	1.50	1.00
English Transcription - Ease of Use	Reject	-1.75	2.75	English Transcription - Ease of Use	Abs Rej	-0.75	1.50
English Transcription - PoF	Reject	-1.75	2.75	English Transcription - PoF	Abs Rej	-0.75	1.50
English Transcription - FoEE	Reject	-1.75	2.75	English Transcription - FoEE	Abs Rej	-0.75	1.50
English Transcription - Likes	Reject	-1.75	2.75	English Transcription - Likes	Abs Rej	-0.75	1.50
English Transcription - Dislikes	Reject	-1.75	2.75	English Transcription - Dislikes	Abs Rej	-0.75	1.50
English Transcription - Change	Reject	-1.75	2.75	English Transcription - Change	Abs Rej	-0.75	1.50
TTS - Likert Scores	Reject	-2.67	1.53	TTS - Likert Scores	Abs Rej	-1.67	0.58

Each of the five strategic points is examined with respect to Evaluative Voting to promote the benefits of its implementation with MRED. Both the 5-point and 11-point scales satisfy all five strategic points identified earlier in this chapter.

The clear advantage of the 11-point scale is that it provides the *Stakeholder* with a richer rating scale than the 5-point scale. This is beneficial since the *Stakeholders* are provided with more granularity for their ratings and the *MRED Operator* can better understand the dispersion of preferences.

6.4. Summary

The Borda Count is exposed as an infeasiable candidate for *Stakeholder Preferences* within MRED. The ranking nature of the Borda Count restricts the flexibility of both the *Stakeholders*, in specifying their preferences, and the *MRED Operator*, in interpreting the data. An increased cognitive burden is also placed on the *Stakeholders* in ranking alternatives with the Borda Count. The ordinality of the Borda Count method inhibits valid aggregation of preferences captured using either method (refer to Section 2 for greater detail). Exploration of both the Borda Count and Pairwise Comparison further justified the decision to implement Evaluative Voting in MRED.

Implementing the 5-point scale revealed that it has nearly the same qualities as the 11-point scale, yet significantly reduced the granularity at which *Stakeholders* can provide preferences. The lack of granularity in the 5-point scale appears to collectively inhibit the *Stakeholders* from differentiating their “preferred” alternatives from their “strongly preferred” alternatives. In addition, it makes it more challenging for the *MRED Operator* to differentiate the levels of preference of each alternative.

Majority Judgment and Evaluative Voting are explored in greater detail. Majority Judgment is demonstrated as a method proven in the literature to determine a group’s preference based upon a linguistic ordinal rating scale. The validity of Evaluative Voting is an active topic of discussion since it’s based upon aggregation on a cardinal scale. However, EV has provided value to the MRED Operator by giving a measure of variance. Majority Judgment’s aggregate median values are restricted to the terms defined on the voting scale; 11 terms in the case of MRED. This restriction can be a disadvantage when working with a relatively small quantity of voters. The MRED Operator would like to know what the average preference value and the level of agreement among the stakeholders. Therefore, the MRED Operator would like a rating method that can aggregate numerical data by providing a mean and standard deviation in addition to being valid.

Chapter 7: Conclusions

MRED is an interactive test plan blueprint generator that takes input from several relevant sources and outputs relevant test plan blueprints. MRED contains: 1) an interactive process to identify candidate evaluation elements and eliminate those that are infeasiable or unnecessary given relationships among these elements and 2) a method to capture and handle *Stakeholder Preferences* of evaluation elements while minimizing the burden on these *Stakeholders*. MRED draws on the Systems Engineering model to break a technology down into its constituent elements. While Systems Engineering decomposes a technology into constituent requirements and then builds it back up during the realization process, MRED specifies test plan blueprints for these constituent elements in support of verification and validation. MRED inputs and outputs are verified against a robot arm example. Majority Judgment and Evaluative Voting are selected and implemented with MRED to handle stakeholder preferences prior to outputting final test plan blueprints. MRED's model and process are validated against known test plans for a speech translation technology that was successfully evaluated. MRED's ability to eliminate blueprint elements based upon constraints and relationships is highlighted for technologies who have some *TTLs* that are not mature and reliable enough for testing.

The first section acknowledges the work that is performed to address each research question. The second section discusses the contributions and research impact of this dissertation effort. Lastly, the third section presents areas of future research relating to the development of MRED.

7.1. Addressing Research Questions

7.1.1. Research Question 1

How should an evaluation test plan generator be modeled to exploit the relationships among multiple deterministic inputs and output test blueprints?

MRED is devised as the evaluation test plan generator for this research effort. MRED uses hierarchical input of a technology's physical structure and functional performance to exploit the relationships among inputs and outputs. MRED interprets and applies inherent and technological constraints between technology test levels through matrix manipulation and linear algebra. MRED takes a modular approach to preference capture and handling. MRED generates blueprints providing *Evaluation Designers* the key characteristic test plan information to move forward on detailed evaluation design. A blueprint contains key test plan characteristics that drive specific test plan development. These key characteristics include the target technology test level(s), metrics, personnel, personnel knowledge and autonomy levels, environment(s), evaluation scenarios, explicit environmental factors, and tools.

7.1.2. Research Question 2

How should MRED integrate stakeholder preferences into the design of test plans?

Preference capture and handling becomes the focal point of MRED's functionality for the *TTLs*. MRED integrates stakeholder preferences by capturing the preferences on an ordinal, linguistic scale. Majority Judgment and Evaluative Voting are two preference capture and handling methods that are implemented into MRED to determine the most preferred blueprint elements. These methods are implemented in both the test plan blueprint generation for the robot arm and speech translation technologies. Each method produced identical sets of blueprints. However, examples presented in Chapter 6 show where the two methods would yield different results.

The challenge with recommending either Majority Judgment and Evaluative Voting for use in MRED is there are differing opinions as to what methods are appropriate for aggregating multiple stakeholder preferences to determine a single most preferred alternative. Knowing that the methods can produce conflicting results, it is difficult recommend one method over the other. Further studies should be conducted that examine the results of both methods as the number of alternatives and the number of voters are increased. The dissertation author appreciates the variance information provided by Evaluative Voting as a test planner with over 15 years of experience designing and implementing evaluations of advanced technologies.

7.1.3. Research Question 3

How can the chosen preference handling method be validated?

Preference handling strategies, including Majority Judgment, Evaluative Voting, and the Borda Count, are explored to see their impact on MRED blueprints and compared against a list of criteria. Multiple linguistic ordinal rating scales are examined to understand their influence on the capture and handling of *Stakeholder Preferences*.

7.2. Contributions and Research Impact

The development of the MRED automatic blueprint generator enables the *MRED Operator* (e.g., *Evaluation Designer, Sponsor*) to formally define and document test plan blueprints. This also includes the documentation of *Stakeholder*

Preferences. This offers tangible benefits enabling evaluation personnel the opportunity to:

- Formalize, and potentially standardize, the development of evaluation blueprints.
- Provide traceability of stakeholder preferences within a test plan, across multiple test plans and across multiple test events.
- Enable blueprints to be altered more rapidly while imposing a minimal burden on the *Stakeholders.*
- Demonstrate the use of Majority Judgment and Evaluative Voting to capture and aggregate *Stakeholder Preferences* in this application.

This research has great potential to speed the development of emerging technologies by streamlining the time it takes to develop comprehensive evaluation test plans. Evaluation test planning can be done more efficiently, thereby saving time, when test designers have blueprints to build upon. It's more cost effective and time-saving to recreate blueprints using MRED in the face of changing requirements. Requirements frequently change in development efforts and it's critical for evaluators to be responsive to fluctuating objectives of the program and functions of the technology. MRED chronicles the pertinent blueprint elements and corresponding stakeholder preferences each time a set of blueprints are generated. This chronicle offers the MRED Operator the ease of recreating blueprints in MRED if one or more preferences are changed or elements are added and/or subtracted. Chronicling blueprints according to their pertinent elements offers the stakeholders an opportunity to compare the focus of one evaluation to the next in like terms. Stakeholders can speak more intelligently on the goals of individual evaluations and the differences among multiple evaluations by understanding the blueprints.

Blueprint generation will enhance the efficiency of producing test plans for a single technology at a given moment in time. Additionally, continuous blueprint generation will highlight the evolution of an evaluation focus for a specific technology over the course of multiple test events. Likewise, relevant comparisons could be drawn between blueprints of similar technologies. Ultimately, blueprint generation and regeneration will save time in test plan development leading to speedier technology development.

7.3. Future Work

The dissertation author is in various stages of collaboration with other evaluation personnel regarding the extension of this work. Currently, the dissertation author is exploring MRED as a tool to develop test plan blueprints for information exchange standards in manufacturing with other personnel in NIST’s Intelligent Systems Division. This has the potential to extend MRED to apply to the production of validation, conformance and/or performance test plan blueprints. Likewise, personnel from another division at NIST have expressed interest in exploring MRED’s potential to develop test plan blueprints to support the development of performance metrics for physical security modeling.

There are other potential research extensions for this work including:

- Explore other methods to capture and handle *Stakeholder Preferences* (e.g., weighting methods, etc.) to reflect those *Stakeholders* that have more or less influence in the test plan development process.

- Develop a calibration method for use in MRED to help enable the valid aggregation of stakeholder preferences.
- Explore cost calculations corresponding to specific test plan blueprints. This extension would determine an approach to produce cost estimates for each blueprint better informing the *Stakeholders* as to which blueprint(s) should be implemented.
- Automate reporting of missing or insufficient resources. This would enable the *MRED Operator* to determine if output blueprints will be limited at stage within MRED prior to the capture and handling of *Stakeholder Preferences*.
- Account for uncertainty in: 1) the availability of blueprint elements, 2) degree of a technology's *Maturity* and 3) *Stakeholder Preferences* and mitigate its propagation through the MRED process.

Test plan development is expected to remain a challenge to the test and evaluation community as more technologies are developed and integrated with robotic elements. The MRED blueprint generator is envisioned to be a tool to further speed the development of technology through more efficient test planning.

Appendix A: Technology Decomposition

This appendix introduces an alternate strategy to decoupling a technology's physical and functional elements as compared to the strategy presented in Section 3.5.3. The goal of this strategy is to symbolically decouple the components and capabilities to arrive at one-to-one relationships. It is determined that the strategy presented in 3.5.3 is more suited for integration into MRED while the strategy presented in this appendix could prove beneficial in the future.

Section 3.5.3 presents different types of relationships that can exist between components and capabilities; a single component can support a single capability; a single component can support multiple capabilities; multiple components can support a single capability; and multiple components can support multiple capabilities. It is desirable to have each component support a single capability where every capability is supported by only one component; this produces a natural one-to-one mapping. This occurrence is the exception and not the norm so the complex relationships described above need to be explored.

Figure 30 presents an example complex relationship where Component 3 influences Capability 2 and Capability 3. The strength of the relationship between this component and capabilities is hard to assess. Questions include: "Does this component equally impact both capabilities?"; "If not, how much greater does this component impact one capability as compared to another?"; and "Can the component be at a development state where it fulfills its contribution to one capability and not the other?" Figure 30 represents each of the components that support multiple capabilities as symbolically-separate entities.

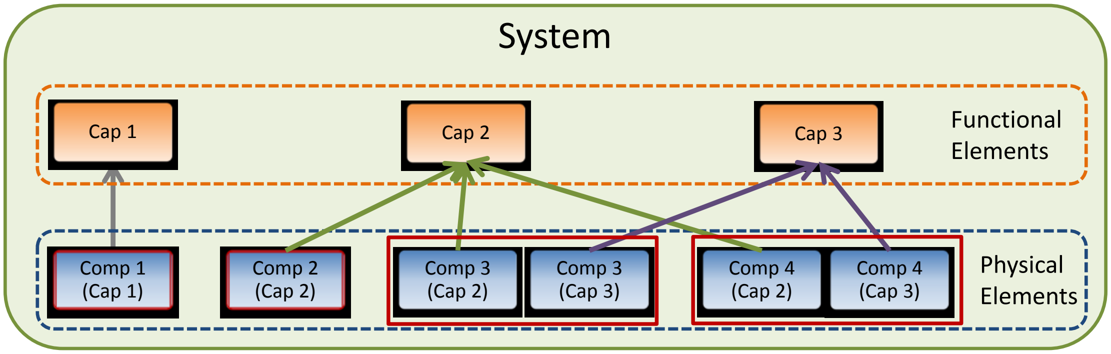


Figure 30: Relationships between "Symbollically-Decoupled" Components and Capabilities

Figure 30 represents Component 3 into two elements; one that supports Capability 2 and another that supports Capability 3.

This symbolic representation of the components enables a component's technology state (i.e. maturity) to be decoupled based upon the capabilities it supports. Multiple maturity states can be defined for a single component that has been symbolically-decoupled. Instead of having to determine a single state of maturity for Component 3 different maturity states can be defined for each symbolically-separate element of Component 3. For example, Component 3 (Cap 2) may be fully mature if Component 3 fulfills its contribution to Capability 2 while Component 3 (Cap 3) may not be mature if Component 3 has yet to fulfill its contribution to Capability 3. Table 54 presents these symbolically-decoupled relationships in a matrix format. Table 54 makes it apparent that each component element only influences a single capability since each row is filled with one "X." Thus the assignment of technology maturities is more direct and relevant to test plan blueprint generation.

Table 56 - Relationship Matrix between Components and Capabilities corresponding to Figure 30

	CAPABILITIES		
COMPONENTS	1	2	3
1 (Cap 1)	X		
2 (Cap 2)		X	
3 (Cap 2)		X	
3 (Cap 3)			X
4 (Cap 2)		X	
4 (Cap 3)			X

The relationships between components and capabilities are further simplified into one-to-one relationships to accommodate the complexity of multiple components supporting a single capability. Figure 31 shows the result of the symbolic separation of both the components and capabilities. The result is a symbolic one-to-one mapping of the originally complex relationships identified in Figure 30 and expanded in Figure 31.

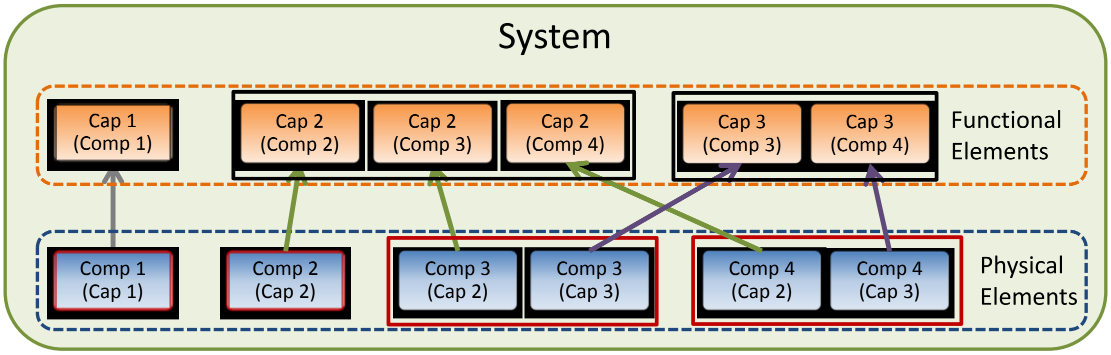


Figure 31: One-to-one Symbolic Mapping of Components and Capabilities

Table 57 presents the expanded matrix corresponding to Table 56. This matrix presents each capability and component in one-to-one relationships as evidenced by the presence of a single "X" in every row and column.

Table 57 - Relationship Matrix between Components and Capabilities corresponding to Figure 31

	CAPABILITIES					
COMPONENTS	1 (Comp 1)	2 (Comp 2)	2 (Comp 3)	2 (Comp 4)	3 (Comp 3)	3 (Comp4)
1 (Cap 1)	**X**					
2 (Cap 2)		**X**				
3 (Cap 2)			**X**			
3 (Cap 3)					**X**	
4 (Cap 2)				**X**		
4 (Cap 3)						**X**

Table 57 is easily manipulated to become an identity matrix. This effort lays the foundation for future work. The symbolic decoupling has great potential to streamline the determination of component and capability technology states. This potential could be realized if the decoupled elements are paired with a method that captures maturity in more granular values, not simply binary terms.

Appendix B: Pseudocode

```
Input Component names
Input Capability names
Input SystemPresence equal to one

Input Technical Performance Metric names
Input Utility Assessment Metric names

Set number of TTL types to three
Set number of Metric types to two
Set number of Available Personnel Characteristics to three
Set number of Environment types to three
Set number of Stakeholder types to five
Set number of Preferred Personnel Characteristics to five
Set number of Types of Evaluation Personnel to five
Set number of Evaluation Scenario Types to three
Set number of Explicit Environmental Factors to two
Set number of Goal Types to five

Input "O" - Binary Relationship Matrix between Components (rows) and Capabilities (columns)
Input "U1" - Binary Relationship Matrix between (Components, Capabilities, System) (columns) and
Technical Performance Metric Types (rows)
Input "U2" - Binary Relationship Matrix between (Capabilities, System) (columns) and Utility
Assessment Metric Types (rows)

Total TTLs = Number of Component names + Number of Capability names + one
TTLs for Utility = Number of Capability names + one

Input Component Maturity (Fully-Developed = one, Immature = zero)
Capability Maturity = Component Maturity times "O" divided by the number of components that
influence each capability
System Maturity = Product of all Capability Maturities

For all columns of Component Maturity
    If a Component Maturity is zero
        Update/Revise "U1" to indicate which Metric Types cannot be tested

For all columns of Capability Maturity
    If a Capability Maturity is less than one
        Update/Revise "U1" to eliminate Metric Types cannot be captured
        Update/Revise "U2" to eliminate Metric Types cannot be captured

If a System Maturity is less than one
    Update/Revise "U1" to eliminate Metric Types cannot be captured
    Update/Revise "U2" to eliminate Metric Types cannot be captured

For all rows in "U1"
    If sum of row equals zero
        Remove row from "U1"
        Remove corresponding metric from Technical Performance Metric names

For all rows in "U2"
    If sum of row equals zero
```

```
                    Remove row from "U2"
                    Remove corresponding metric from Utility Assessment Metric names

For all columns of "U1"
          If sum of a column equals zero
                    Remove column from "U1"
                    Subtract one from Total TTLs
                    If column corresponds to Component
                              Remove corresponding Component row from "O"
                              Remove corresponding Component name
                    If column corresponds to Capability
                              For corresponding Capability column in "U2"
                                        If sum of column equals zero
                                                  Remove corresponding Capability column from "O"
                                                  Remove corresponding Capability column from "U2"
                                                  Subtract one from TTLs for Utility
                                                  Remove corresponding Capability name
                    Else
                              For corresponding System column in "U2"
                                        If sum of column equals zero
                                                  Remove corresponding System column from "U2"
                                                  Subtract one from TTLs for Utility
                                                  Subtract one from SystemPresence

Input Available Lab Environment names
Input Available Simulated Environment names
Input Available Actual Environment names

Input "X1" - Binary Relationship Matrix between Lab Environments (columns) and Components, Capabilities (rows)
Input "X2" - Binary Relationship Matrix between Simulated Environments (columns) and all TTLs (rows)
Input "X3" - Binary Relationship Matrix between Actual Environment (columns) and Capabilities, System (rows)

For all columns in "X1"
          If sum of column equals zero
                    Remove column from "X1"
                    Remove corresponding environment from Lab Environment names

For all columns in "X2"
          If sum of column equals zero
                    Remove column from "X2"
                    Remove corresponding environment from Simulated Environment names

For all columns in "X3"
          If sum of column equals zero
                    Remove column from "X3"
                    Remove corresponding environment from Actual Environment names

Input Technical Performance Metric Tool names
Input Utility Assessment Tool names

Input "Y1" - Binary Relationship Matrix between Technical Performance Metric Tools (columns) and TTLs (rows)
Input "Y2" - Binary Relationship Matrix between Utility Assessment Tools (columns) and Capabilities, System (rows)


For all columns in "Y1"
    If sum of column equals zero
        Remove column from "Y1"
        Remove corresponding tool from Technical Performance Metric Tool names

For all columns in "Y2"
    If sum of column equals zero
        Remove column from "Y2"
        Remove corresponding tool from Utility Assessment Tool names

Input Presence of End-Users (Binary - zero or one)
Input Presence of Trained Users
Input Presence of Technology Developers
Input Presence of Team Members
Input Presence of Participants

If End-Users are Present
    Input Technical Knowledge of End-Users (None, Low, Med, High)
    Input Operational Knowledge of End-Users (None, Low, Med, High)

If Trained Users are Present
    Input Technical Knowledge of Trained Users
    Input Operational Knowledge of Trained Users

If Technology Developers are Present
    Input Technical Knowledge of Technology Developers
    Input Operational Knowledge of Technology Developers

If Team Members are Present
    Input Technical Knowledge of Team Members
    Input Operational Knowledge of Team Members

If Participants are Present
    Input Technical Knowledge of Participants
    Input Operational Knowledge of Participants

If Capability Names equals zero AND SystemPresence equals zero AND Presence of End-Users equals one
    Presence of End-Users equals zero

If Technical Performance Metric names equals zero AND Presence of Technology Developers equals one
    Presence of Technology Developers equals zero

If Presence of Trained Users equals zero AND Presence of Technology Developers equals zero
    Total TTLs equals Total TTLs minus Number of Components
    Set Component names equal to 0
    Remove Component columns from "U1"
    Remove Component rows in "X1"
    Remove Component rows in "X2"

        For all rows in "U1"
```

```
            If sum of row equals zero
                Remove row from "U1"
                Remove corresponding metric from Technical Performance Metric names

        For all columns in "X1"
            If sum of column equals zero
                Remove column from "X1"
                Remove corresponding environment from Lab Environment names

        For all columns in "X2"
            If sum of column equals zero
                Remove column from "X2"
                Remove corresponding environment from Simulated Environment names

If Presence of End-Users equals zero AND Presence of Trained users equals zero
    Set Utility Assessment names equal to 0
    Set Utility Assessment Tool names equal to 0
    Set "Y2" equal to 0
    Set "U2" equal to 0

For all columns of "U1"
    If sum of a column equals zero
        If column corresponds to Component
            Remove corresponding Component row from "O"
            Remove corresponding Component name
            Remove Component row in "X1"
            Remove Component row in "X2"
            Remove Component column from "U1"
            Subtract one from Total TTLs

        If column corresponds to Capability
            For corresponding Capability column in "U2"
                If sum of column equals zero
                    Remove corresponding Capability column from "O"
                    Remove corresponding Capability column from "U2"
                    Subtract one from TTLs for Utility
                    Remove corresponding Capability name
                    Remove Capability row in "X1"
                    Remove Capability row in "X2"
                    Remove Capability row in "X3"
                    Remove Capability column from "U1"
                    Subtract one from Total TTLs
        Else
            For corresponding System column in "U2"
                If sum of column equals zero
                    Remove corresponding System column from "U2"
                    Subtract one from TTLs for Utility
                    Subtract one from SystemPresence
                    Remove System row in "X2"
                    Remove System row in "X3"
                    Remove System column from "U1"
                    Subtract one from Total TTLs

For all rows in "U1"
```

```
        If sum of row equals zero
                Remove row from "U1"
                Remove corresponding metric from Technical Performance Metric names

For all rows in "U2"
        If sum of row equals zero
                Remove row from "U2"
                Remove corresponding metric from Utility Assessment Metric names

For all rows in "Y1"
        If sum of row equals zero
                Remove row from "Y1"
                Remove corresponding Technical Performance Metric row in "U1"
                Remove corresponding metric from Technical Performance Metric names

For all rows in "Y2"
        If sum of row equals zero
                Remove row from "Y2"
                Remove corresponding Utility Assessment Metric row from "U2"
                Remove corresponding metric from Utility Assessment Metric names


For all columns in "Y1"
        If sum of column equals zero
                Remove column from "Y1"
                Remove corresponding tool from Technical Performance Metric Tool names

For all columns in "Y2"
        If sum of column equals zero
                Remove column from "Y2"
                Remove corresponding tool from Utility Assessment Tool names


For all columns in "X1"
        If sum of column equals zero
                Remove column from "X1"
                Remove corresponding environment from Lab Environment names

For all columns in "X2"
        If sum of column equals zero
                Remove column from "X2"
                Remove corresponding environment from Simulated Environment names

For all columns in "X3"
        If sum of column equals zero
                Remove column from "X3"
                Remove corresponding environment from Actual Environment names

For all rows in "X1"
        If sum of row equals zero AND
                If row corresponds to Component
                        If sum of corresponding row in "X2" equals zero
                                Remove Component row in "X1"
                                Remove Component row in "X2"
                                Remove corresponding Component row from "O"
                                Remove corresponding Component name
                                Remove corresponding Component column from "U1"

                If row corresponds to a Capability
                        If sum of corresponding row in "X2" equals 0 AND sum of corresponding
                        row in "X3" equals zero
                                Remove Capability row in "X1"
                                Remove Capability row in "X2"
                                Remove Capability row in "X3"
                                Remove corresponding Capability column from "O"
                                Remove corresponding Capability name
                                Remove corresponding Capability column from "U1"
                                Remove corresponding Capability column from "U2"
                                Subtract one from TTLs in Utility

If SystemPresence equals one
        For corresponding System row in "X2"
                If sum of row equals zero and sum of corresponding System row in "X3" equals zero
                        Remove System row in "X2"
                        Remove System row in "X3"
                        Remove corresponding System column from "U1"
                        Remove corresponding System column from "U2"
                        SystemPresence equals zero
                        Subtract one from TTLs in Utility

Input Presence of Stakeholder-Buyer
Input Presence of Stakeholder-Evaluation Designer
Input Presence of Stakeholder-Sponsor
Input Presence of Stakeholder-Technology Developer
Input Presence of Stakeholder-User

Counter equals one
If "U1" does not equal zero
        For all columns in "U1"
                For each row in "U1"
                        If "U1"(row, column) equals one
                                TTL_Metric_Pair(counter) equals text string of corresponding TTL
                                and Technical Performance Metric
                                counter equals counter plus one

If "U2" does not equal zero
        For each column "U2"
                For each row in "U2"
                        If "U2"(row, column) equals one
                                TTL_Metric_Pair(counter) equals text string of corresponding TTL
                                and Utility Assessment Metric
                                counter equals counter plus one

For all TTL_Metric_Pair
        Input Linguistic Preference of TTL_Metric_Pair from Stakeholder-Buyer
        Input Linguistic Preference of TTL_Metric_Pair from Stakeholder-Evaluation Designer
        Input Linguistic Preference of TTL_Metric_Pair from Stakeholder-Sponsor
        Input Linguistic Preference of TTL_Metric_Pair from Stakeholder-Technology Developer
        Input Linguistic Preference of TTL_Metric_Pair from Stakeholder-User

For all TTL_Metric_Pair
        Calculate Median for all preferences for each TTL_Metric_Pair
```

```
For all TTL_Metric_Pair Linguistic Preferences
    Convert TTL_Metric_Pair Linguistic Preferences to TTL_Metric_Pair Interval Values

For all TTL_Metric_Pair Interval Values
    Calculate Mean for all of the values for each TTL_Metric_Pair
    Calculate Standard Deviation for all of the values for each TTL_Metric_Pair

Display TTL_Metric_Pair Median data
Display TTL_Metric_Pair Mean data
Display TTL_Metric_Pair Standard Deviation data

Group TTL_Metric_Pairs
Remove lowest scoring TTL_Metric_Pairs

For all TTL_Metric_Pair Groups
    Input Linguistic Preference of Tech-User Presence from Stakeholder-Buyer
    Input Linguistic Preference of Tech-User Presence from Stakeholder-Evaluation Designer
    Input Linguistic Preference of Tech-User Presence from Stakeholder-Sponsor
    Input Linguistic Preference of Tech-User Presence from Stakeholder-Technology Developer
    Input Linguistic Preference of Tech-User Presence from Stakeholder-User

For all TTL_Metric_Pair Groups
    For all Tech-Users
        Calculate Median for all preferences for Tech-User Presence

For all TTL_Metric_Pair Groups
    For all Tech-Users
        Convert Tech-User Presence Linguistic Preferences to Tech-User Presence Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Tech-User
    Calculate Standard Deviation for all of the values for each Tech-User

Display Tech-User Presence Median data
Display Tech-User Presence Mean data
Display Tech-User Presence Standard Deviation data

For all TTL_Metric_Pair Groups
    Remove all but highest scoring Tech-User(s)

For all TTL_Metric_Pair Groups
    For all Remaing Tech-Users
        Input Linguistic Preference of Tech-User Knowledge Levels from Stakeholder-Buyer
        Input Linguistic Preference of Tech-User Knowledge Levels from Stakeholder-Evaluation Designer
        Input Linguistic Preference of Tech-User Knowledge Levels from Stakeholder-Sponsor
        Input Linguistic Preference of Tech-User Knowledge Levels from Stakeholder-Technology Developer
        Input Linguistic Preference of Tech-User Knowledge Levels from Stakeholder-User

For all TTL_Metric_Pair Groups
    For all Remaining Tech-Users
```

```
        Calculate Median for all preferences for Tech-User Knowledge Levels

For all TTL_Metric_Pair Groups
    For all Remaining Tech-Users
        Convert Tech-User Knowledge Levels Linguistic Preferences to Tech-User Knowledge Levels Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Tech-User Knowledge Level
    Calculate Standard Deviation for all of the values for each Tech-User Knowledge Level

Display Tech-User Knowledge Level Median data
Display Tech-User Knowledge Level Mean data
Display Tech-User Knowledge Level Standard Deviation data

For all TTL_Metric_Pair Groups
    Remove all but highest scoring Tech-User Knowledge Level(s)

For all TTL_Metric_Pair Groups
    For all Remaing Tech-Users
        Input Linguistic Preference of Tech-User Autonomy Levels from Stakeholder-Buyer
        Input Linguistic Preference of Tech-User Autonomy Levels from Stakeholder-Evaluation Designer
        Input Linguistic Preference of Tech-User Autonomy Levels from Stakeholder-Sponsor
        Input Linguistic Preference of Tech-User Autonomy Levels from Stakeholder-Technology Developer
        Input Linguistic Preference of Tech-User Autonomy Levels from Stakeholder-User

For all TTL_Metric_Pair Groups
    For all Remaining Tech-Users
        Calculate Median for all preferences for Tech-User Autonomy Levels

For all TTL_Metric_Pair Groups
    For all Remaining Tech-Users
        Convert Tech-User Autonomy Levels Linguistic Preferences to Tech-User Autonomy Levels Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Tech-User Autonomy Level
    Calculate Standard Deviation for all of the values for each Tech-User Autonomy Level

Display Tech-User Autonomy Level Median data
Display Tech-User Autonomy Level Mean data
Display Tech-User Autonomy Level Standard Deviation data

For all TTL_Metric_Pair Groups
    Remove all but highest scoring Tech-User Autonomy Level(s)

For all TTL_Metric_Pair Groups
    Input Linguistic Preferences of Team Member and Participant Presence from Stakeholder-Buyer
    Input Linguistic Preferences of Team Member and Participant Presence from Stakeholder-Evaluation Designer
    Input Linguistic Preferences of Team Member and Participant Presence from Stakeholder-Sponsor
    Input Linguistic Preferences of Team Member and Participant Presence from Stakeholder-Technology Developer
    Input Linguistic Preferences of Team Member and Participant Presence from Stakeholder-User

For all TTL_Metric_Pair Groups
    For all Team Member and Participant
        Calculate Median for all preferences for Team Member and Participant Presence

For all TTL_Metric_Pair Groups
    For all Team Member and Participant
        Convert Team Member and Participant Presence Linguistic Preferences to Team Member and Participant Presence Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Team Member and Participant
    Calculate Standard Deviation for all of the values for each Team Member and Participant

Display Team Member and Participant Presence Median data
Display Team Member and Participant Presence Mean data
Display Team Member and Participant Presence Standard Deviation data

For all TTL_Metric_Pair Groups
    Remove all but highest scoring Team Member and Participant(s)

For all TTL_Metric_Pair Groups
    For all Remaing Team Members and Participants
        Input Linguistic Preferences of Team Member and Participant Knowledge Levels from Stakeholder-Buyer
        Input Linguistic Preferences of Team Member and Participant Knowledge Levels from Stakeholder-Evaluation Designer
        Input Linguistic Preferences of Team Member and Participant Knowledge Levels from Stakeholder-Sponsor
        Input Linguistic Preferences of Team Member and Participant Knowledge Levels from Stakeholder-Technology Developer
        Input Linguistic Preferences of Team Member and Participant Knowledge Levels from Stakeholder-User

For all TTL_Metric_Pair Groups
    For all Remaining Team Members and Participants
        Calculate Median for all preferences for Team Member and Participant Knowledge Levels

For all TTL_Metric_Pair Groups
    For all Remaining Team Members and Participants
        Convert Team Member and Participant Knowledge Levels Linguistic Preferences to Team Member and Participant Knowledge Levels Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Team Member and Participant Knowledge Level
    Calculate Standard Deviation for all of the values for each Team Member and Participant Knowledge Level

Display Team Member and Participant Knowledge Level Median data
Display Team Member and Participant Knowledge Level Mean data
Display Team Member and Participant Knowledge Level Standard Deviation data
```

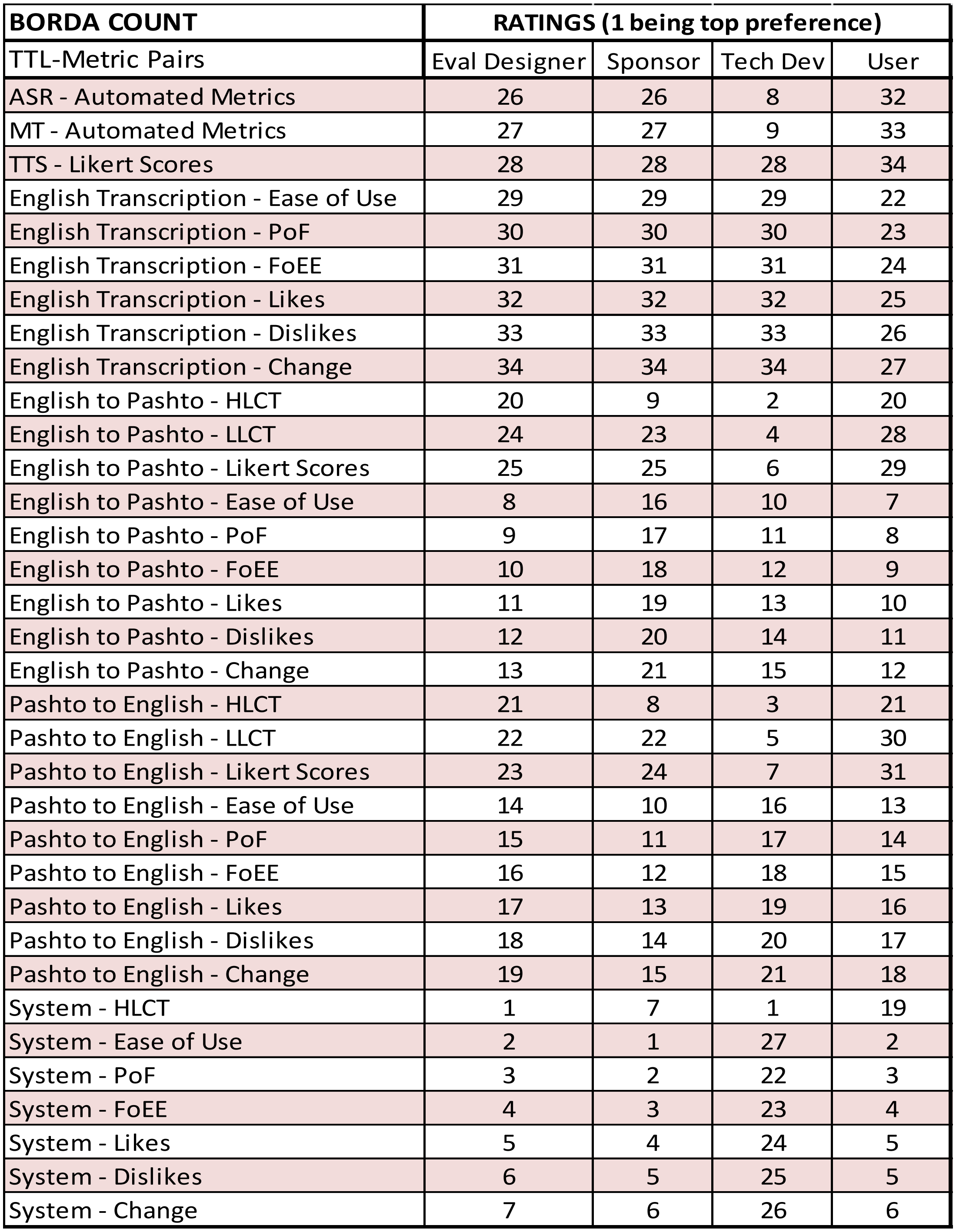
```
For all TTL_Metric_Pair Groups
    Remove all but highest scoring Team Member and Participant Knowledge Level(s)

For all TTL_Metric_Pair Groups
    Input Linguistic Preferences of Environments from Stakeholder-Buyer
    Input Linguistic Preferences of Environments from Stakeholder-Evaluation Designer
    Input Linguistic Preferences of Environments from Stakeholder-Sponsor
    Input Linguistic Preferences of Environments from Stakeholder-Technology Developer
    Input Linguistic Preferences of Environments from Stakeholder-User

For all TTL_Metric_Pair Groups
    Calculate Median for all preferences for Environments

For all TTL_Metric_Pair Groups
    Convert Environment Linguistic Preferences to Environment Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Environment
    Calculate Standard Deviation for all of the values for each Environment

Display Environment Median data
Display Environment Mean data
Display Environment Standard Deviation data

For all TTL_Metric_Pair Groups
    Remove all but highest scoring Environment(s)

For all TTL_Metric_Pair Groups
    Input Linguistic Preferences of Evaluation Scenarios from Stakeholder-Buyer
    Input Linguistic Preferences of Evaluation Scenarios from Stakeholder-Evaluation Designer
    Input Linguistic Preferences of Evaluation Scenarios from Stakeholder-Sponsor
    Input Linguistic Preferences of Evaluation Scenarios from Stakeholder-Technology
    Developer
    Input Linguistic Preferences of Evaluation Scenarios from Stakeholder-User

For all TTL_Metric_Pair Groups
    Calculate Median for all preferences for Evaluation Scenarios

For all TTL_Metric_Pair Groups
    Convert Evaluation Scenario Linguistic Preferences to Evaluation Scenario Interval Values

For all TTL_Metric_Pair Groups
    Calculate Mean for all of the values for each Evaluation Scenario
    Calculate Standard Deviation for all of the values for each Evaluation Scenario

Display Evaluation Scenario Median data
Display Evaluation Scenario Mean data
Display Evaluation Scenario Standard Deviation data

For all TTL_Metric_Pair Groups
    Remove all but highest scoring Evaluation Scenario(s)

For all TTL_Metric_Pair Groups
    Input Linguistic Preferences of Environmental Factors from Stakeholder-Buyer
    Input Linguistic Preferences of Environmental Factors from Stakeholder-Evaluation Designer
```

```
          Input Linguistic Preferences of Environmental Factors from Stakeholder-Sponsor
          Input Linguistic Preferences of Environmental Factors from Stakeholder-Technology
          Developer
          Input Linguistic Preferences of Environmental Factors from Stakeholder-User

For all TTL_Metric_Pair Groups
          Calculate Median for all preferences for Environmental Factors

For all TTL_Metric_Pair Groups
          Convert Environmental Factors Linguistic Preferences to Environmental Factors Interval
          Values

For all TTL_Metric_Pair Groups
          Calculate Mean for all of the values for each Environmental Factor
          Calculate Standard Deviation for all of the values for each Environmental Factor

Display Environmental Factors Median data
Display Environmental Factors Mean data
Display Environmental Factors Standard Deviation data

For all TTL_Metric_Pair Groups
          Remove all but highest scoring Environmental Factor(s)

For all TTL_Metric_Pair Groups
          Calculate Overall Environmental Complexity from Chosen Environmental Factors

For all TTL_Metric_Pair Groups
          Display Blueprint Elements
```

Appendix C: Borda Count Application to S2S

Building upon the example presented in Section 6.2, the Borda Count is applied to capturing and handling *Stakeholder Preferences* for the S2S technology *TTL-Metric* pairs. The *MRED Operator* (and author of this work)*,* from his knowledge of the *Stakeholders*, generates *Stakeholder Preferences* for Borda Ratings for all 34 of the *TTL-Metric* pairs. The raw ratings are documented in Table 58 while the Borda scores (generated from the raw ratings) are shown in Table 59. These Borda scores are consistent with the ratings included in Table 33.

Table 58 - Borda Count Ratings - Full Ratings with Borda Scores



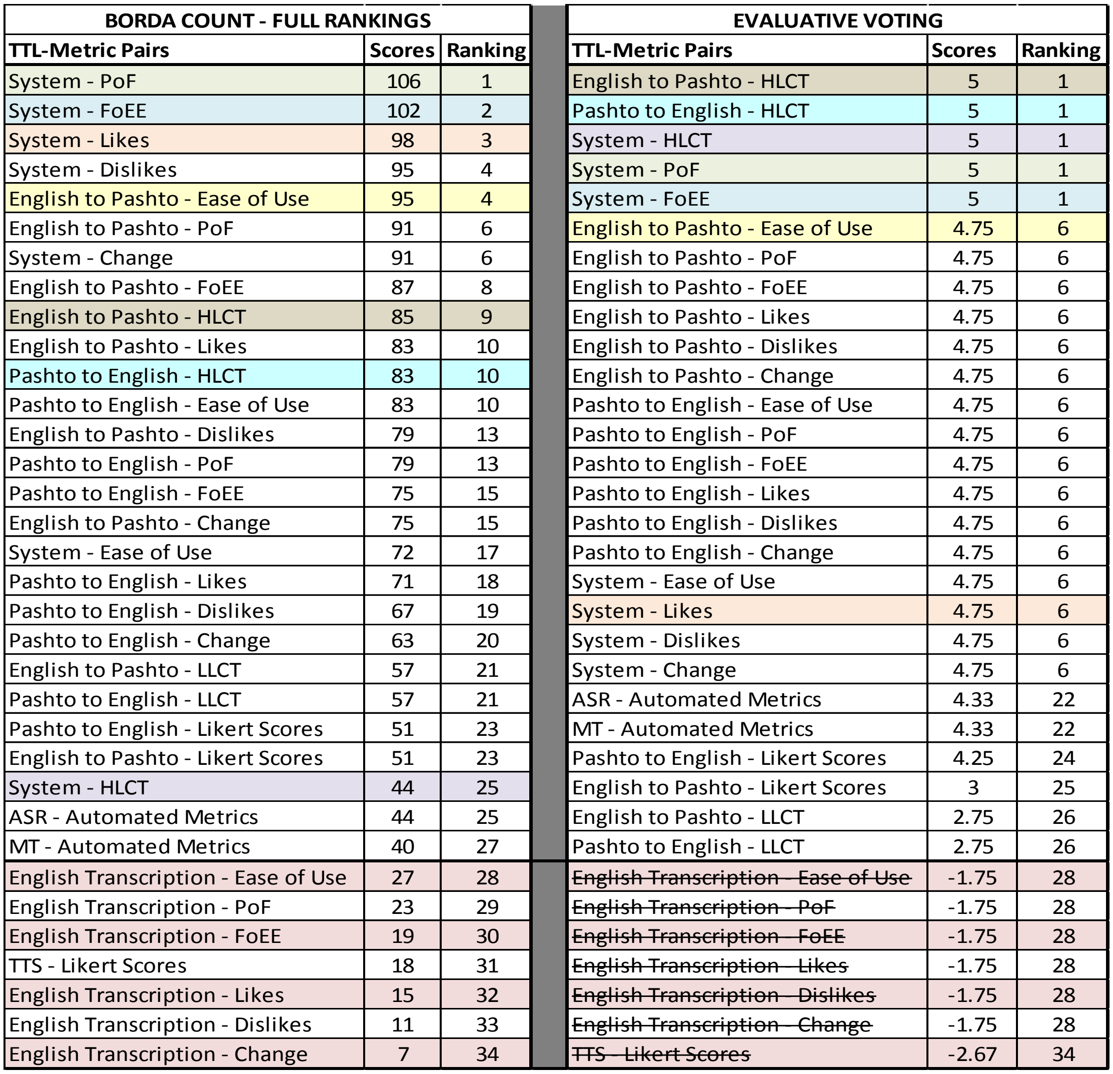
BORDA COUNT	RATINGS (1 being top preference)			
TTL-Metric Pairs	Eval Designer	Sponsor	Tech Dev	User
ASR - Automated Metrics	26	26	8	32
MT - Automated Metrics	27	27	9	33
TTS - Likert Scores	28	28	28	34
English Transcription - Ease of Use	29	29	29	22
English Transcription - PoF	30	30	30	23
English Transcription - FoEE	31	31	31	24
English Transcription - Likes	32	32	32	25
English Transcription - Dislikes	33	33	33	26
English Transcription - Change	34	34	34	27
English to Pashto - HLCT	20	9	2	20
English to Pashto - LLCT	24	23	4	28
English to Pashto - Likert Scores	25	25	6	29
English to Pashto - Ease of Use	8	16	10	7
English to Pashto - PoF	9	17	11	8
English to Pashto - FoEE	10	18	12	9
English to Pashto - Likes	11	19	13	10
English to Pashto - Dislikes	12	20	14	11
English to Pashto - Change	13	21	15	12
Pashto to English - HLCT	21	8	3	21
Pashto to English - LLCT	22	22	5	30
Pashto to English - Likert Scores	23	24	7	31
Pashto to English - Ease of Use	14	10	16	13
Pashto to English - PoF	15	11	17	14
Pashto to English - FoEE	16	12	18	15
Pashto to English - Likes	17	13	19	16
Pashto to English - Dislikes	18	14	20	17
Pashto to English - Change	19	15	21	18
System - HLCT	1	7	1	19
System - Ease of Use	2	1	27	2
System - PoF	3	2	22	3
System - FoEE	4	3	23	4
System - Likes	5	4	24	5
System - Dislikes	6	5	25	5
System - Change	7	6	26	6

Table 59 – Borda Scores for *TTL-Metric* pairs for S2S test planning

SCORES CONVERTED FROM RATINGS					
BORDA COUNT	SCORES (Higher being top preference)				POINTS
TTL-Metric Pairs	Eval Designer	Sponsor	Tech Dev	User	
ASR - Automated Metrics	8	8	26	2	44
MT - Automated Metrics	7	7	25	1	40
TTS - Likert Scores	6	6	6	0	18
English Transcription - Ease of Use	5	5	5	12	27
English Transcription - PoF	4	4	4	11	23
English Transcription - FoEE	3	3	3	10	19
English Transcription - Likes	2	2	2	9	15
English Transcription - Dislikes	1	1	1	8	11
English Transcription - Change	0	0	0	7	7
English to Pashto - HLCT	14	25	32	14	85
English to Pashto - LLCT	10	11	30	6	57
English to Pashto - Likert Scores	9	9	28	5	51
English to Pashto - Ease of Use	26	18	24	27	95
English to Pashto - PoF	25	17	23	26	91
English to Pashto - FoEE	24	16	22	25	87
English to Pashto - Likes	23	15	21	24	83
English to Pashto - Dislikes	22	14	20	23	79
English to Pashto - Change	21	13	19	22	75
Pashto to English - HLCT	13	26	31	13	83
Pashto to English - LLCT	12	12	29	4	57
Pashto to English - Likert Scores	11	10	27	3	51
Pashto to English - Ease of Use	20	24	18	21	83
Pashto to English - PoF	19	23	17	20	79
Pashto to English - FoEE	18	22	16	19	75
Pashto to English - Likes	17	21	15	18	71
Pashto to English - Dislikes	16	20	14	17	67
Pashto to English - Change	15	19	13	16	63
System - HLCT	1	27	1	15	44
System - Ease of Use	32	1	7	32	72
System - PoF	31	32	12	31	106
System - FoEE	30	31	11	30	102
System - Likes	29	30	10	29	98
System - Dislikes	28	29	9	29	95
System - Change	27	28	8	28	91

The Borda scores generated in Table 60 are compared to the Evaluative Voting scores (produced in Section 5.3.1) in Table 33. The Borda Scores are not compared to the Majority Voting grades given that the Evaluative Voting scores present more granularity as compared to the Majority Voting grades.

Table 60 - Comparison of Borda Count Rankings to Evaluative Voting Ratings for S2S *TTL-Metric* Pairs



BORDA COUNT - FULL RANKINGS			EVALUATIVE VOTING		
TTL-Metric Pairs	Scores	Ranking	TTL-Metric Pairs	Scores	Ranking
System - PoF	106	1	English to Pashto - HLCT	5	1
System - FoEE	102	2	Pashto to English - HLCT	5	1
System - Likes	98	3	System - HLCT	5	1
System - Dislikes	95	4	System - PoF	5	1
English to Pashto - Ease of Use	95	4	System - FoEE	5	1
English to Pashto - PoF	91	6	English to Pashto - Ease of Use	4.75	6
System - Change	91	6	English to Pashto - PoF	4.75	6
English to Pashto - FoEE	87	8	English to Pashto - FoEE	4.75	6
English to Pashto - HLCT	85	9	English to Pashto - Likes	4.75	6
English to Pashto - Likes	83	10	English to Pashto - Dislikes	4.75	6
Pashto to English - HLCT	83	10	English to Pashto - Change	4.75	6
Pashto to English - Ease of Use	83	10	Pashto to English - Ease of Use	4.75	6
English to Pashto - Dislikes	79	13	Pashto to English - PoF	4.75	6
Pashto to English - PoF	79	13	Pashto to English - FoEE	4.75	6
Pashto to English - FoEE	75	15	Pashto to English - Likes	4.75	6
English to Pashto - Change	75	15	Pashto to English - Dislikes	4.75	6
System - Ease of Use	72	17	Pashto to English - Change	4.75	6
Pashto to English - Likes	71	18	System - Ease of Use	4.75	6
Pashto to English - Dislikes	67	19	System - Likes	4.75	6
Pashto to English - Change	63	20	System - Dislikes	4.75	6
English to Pashto - LLCT	57	21	System - Change	4.75	6
Pashto to English - LLCT	57	21	ASR - Automated Metrics	4.33	22
Pashto to English - Likert Scores	51	23	MT - Automated Metrics	4.33	22
English to Pashto - Likert Scores	51	23	Pashto to English - Likert Scores	4.25	24
System - HLCT	44	25	English to Pashto - Likert Scores	3	25
ASR - Automated Metrics	44	25	English to Pashto - LLCT	2.75	26
MT - Automated Metrics	40	27	Pashto to English - LLCT	2.75	26
English Transcription - Ease of Use	27	28	~~English Transcription - Ease of Use~~	-1.75	28
English Transcription - PoF	23	29	~~English Transcription - PoF~~	-1.75	28
English Transcription - FoEE	19	30	~~English Transcription - FoEE~~	-1.75	28
TTS - Likert Scores	18	31	~~English Transcription - Likes~~	-1.75	28
English Transcription - Likes	15	32	~~English Transcription - Dislikes~~	-1.75	28
English Transcription - Dislikes	11	33	~~English Transcription - Change~~	-1.75	28
English Transcription - Change	7	34	~~TTS - Likert Scores~~	-2.67	34

Several observations can be made about the data presented in Table 60. They include:

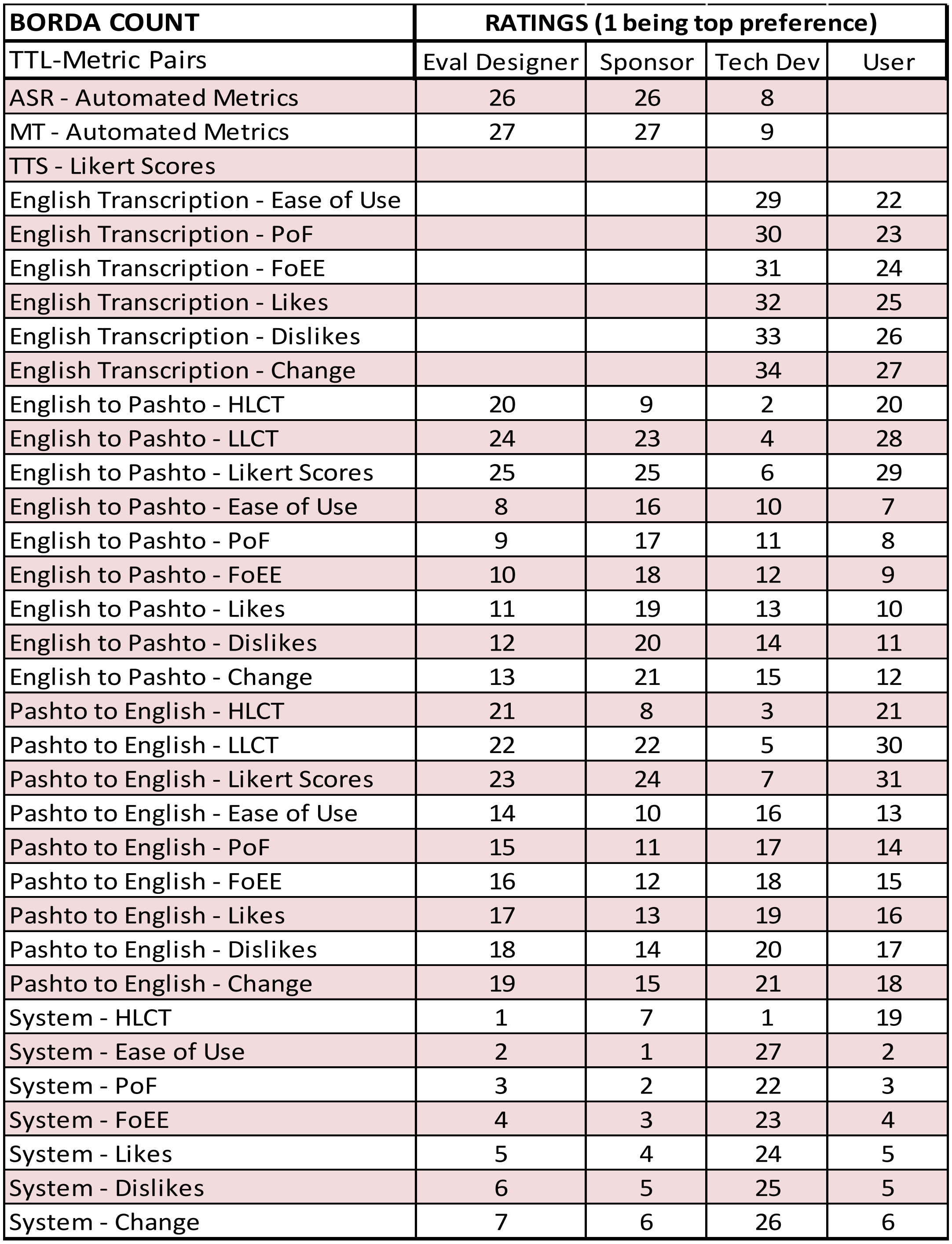
- Only two of the top five alternatives from Evaluative Voting remained in the top five when the Borda Count is applied.
- The other three candidates in the top five in Evaluative Voting have a greater shift in their ranking under the Borda Count.

- English to Pashto – HLCT moved from tied 1st (Evaluative Voting) to 9th (Borda Count).
- Pashto to English – HLCT moved from tied 1st (Evaluative Voting) to tied 10th (Borda Count).
- System – HLCT moved from tied 1st (Evaluative Voting) to tied 25th (Borda Count). This is the most drastic shift and very concerning.

In addition to the five strategic points noted earlier in this Chapter, the Borda Count exhibits a significant problem; There is no indication as to the threshold for a *TTL-Metric* pair to be evaluated or discarded. Of the 34 *TTL-Metric* pairs presented in , how many points would a *TTL-Metric* pair have to acquire to be considered? Or what is the lowest ranking a *TTL-Metric* pair could get and still be considered for evaluation? Evaluative Voting addresses this concern where the *MRED Operator* naturally sets the scale at 0 (where negative values are "against" evaluation and the positive values are "for" evaluation).

The *Stakeholders* are now provided with the option to abstain from ranking a specific alternative using the Borda Count. Any alternative that a *Stakeholder* chooses not to rank is automatically given a score of "0" in the Borda Count. For the sake of this exploration, let every "NV" or negative rating in the Evaluative Voting scheme translate into an abstained ranking in the Borda Count. Table 61 presents the raw ratings while Table 62 presents the Borda Scores.

Table 61 - Borda Count Ratings – Partial Ratings



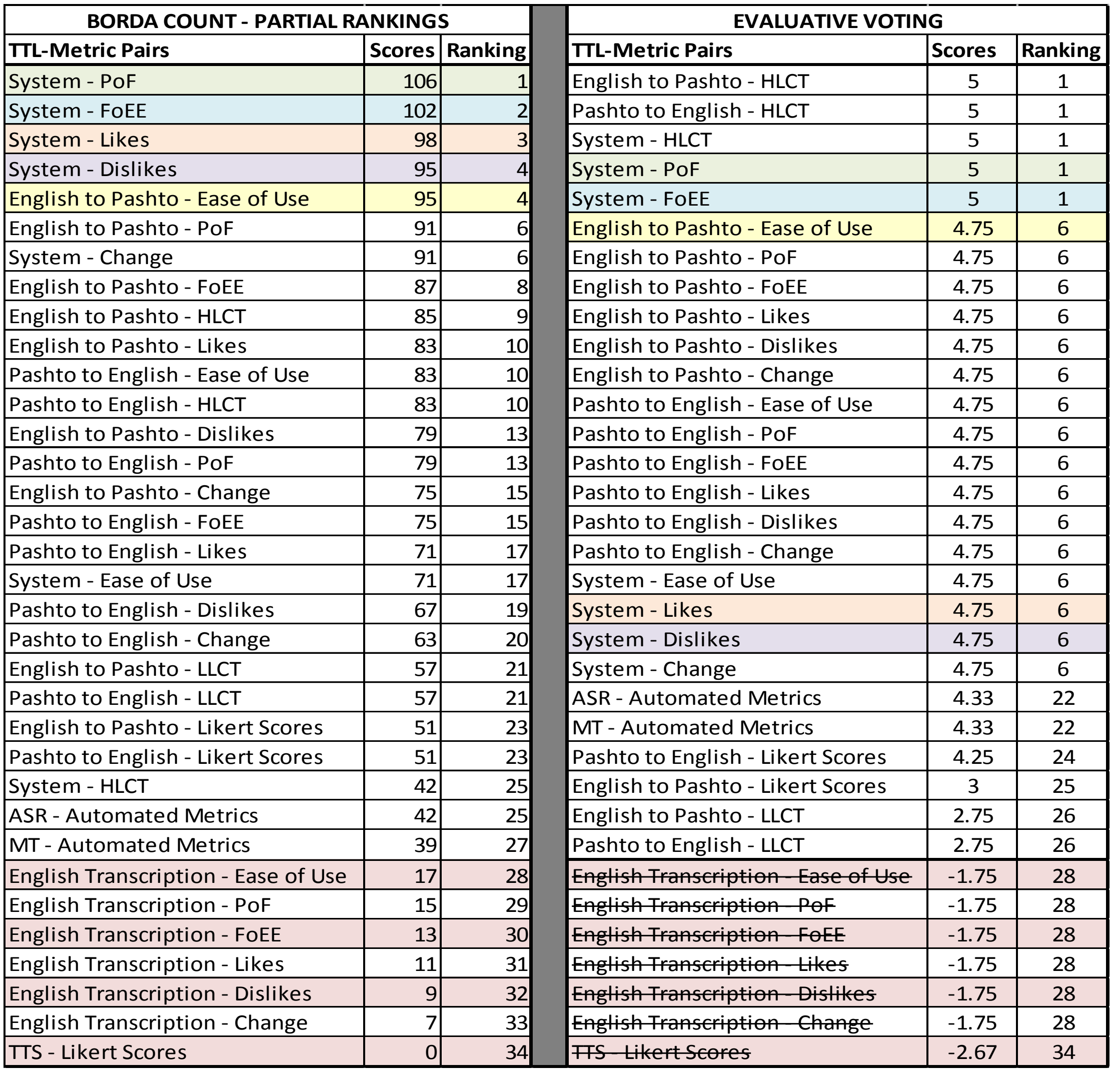
BORDA COUNT	RATINGS (1 being top preference)			
TTL-Metric Pairs	Eval Designer	Sponsor	Tech Dev	User
ASR - Automated Metrics	26	26	8	
MT - Automated Metrics	27	27	9	
TTS - Likert Scores				
English Transcription - Ease of Use			29	22
English Transcription - PoF			30	23
English Transcription - FoEE			31	24
English Transcription - Likes			32	25
English Transcription - Dislikes			33	26
English Transcription - Change			34	27
English to Pashto - HLCT	20	9	2	20
English to Pashto - LLCT	24	23	4	28
English to Pashto - Likert Scores	25	25	6	29
English to Pashto - Ease of Use	8	16	10	7
English to Pashto - PoF	9	17	11	8
English to Pashto - FoEE	10	18	12	9
English to Pashto - Likes	11	19	13	10
English to Pashto - Dislikes	12	20	14	11
English to Pashto - Change	13	21	15	12
Pashto to English - HLCT	21	8	3	21
Pashto to English - LLCT	22	22	5	30
Pashto to English - Likert Scores	23	24	7	31
Pashto to English - Ease of Use	14	10	16	13
Pashto to English - PoF	15	11	17	14
Pashto to English - FoEE	16	12	18	15
Pashto to English - Likes	17	13	19	16
Pashto to English - Dislikes	18	14	20	17
Pashto to English - Change	19	15	21	18
System - HLCT	1	7	1	19
System - Ease of Use	2	1	27	2
System - PoF	3	2	22	3
System - FoEE	4	3	23	4
System - Likes	5	4	24	5
System - Dislikes	6	5	25	5
System - Change	7	6	26	6

Table 62 - Borda Scores for TTL-Metric pairs for S2S test planning of Partial Ratings

BORDA COUNT	SCORES (Higher being top preference)				POINTS
TTL-Metric Pairs	Eval Designer	Sponsor	Tech Dev	User	
ASR - Automated Metrics	8	8	26	0	42
MT - Automated Metrics	7	7	25	0	39
TTS - Likert Scores	0	0	0	0	0
English Transcription - Ease of Use	0	0	5	12	17
English Transcription - PoF	0	0	4	11	15
English Transcription - FoEE	0	0	3	10	13
English Transcription - Likes	0	0	2	9	11
English Transcription - Dislikes	0	0	1	8	9
English Transcription - Change	0	0	0	7	7
English to Pashto - HLCT	14	25	32	14	85
English to Pashto - LLCT	10	11	30	6	57
English to Pashto - Likert Scores	9	9	28	5	51
English to Pashto - Ease of Use	26	18	24	27	95
English to Pashto - PoF	25	17	23	26	91
English to Pashto - FoEE	24	16	22	25	87
English to Pashto - Likes	23	15	21	24	83
English to Pashto - Dislikes	22	14	20	23	79
English to Pashto - Change	21	13	19	22	75
Pashto to English - HLCT	13	26	31	13	83
Pashto to English - LLCT	12	12	29	4	57
Pashto to English - Likert Scores	11	10	27	3	51
Pashto to English - Ease of Use	20	24	18	21	83
Pashto to English - PoF	19	23	17	20	79
Pashto to English - FoEE	18	22	16	19	75
Pashto to English - Likes	17	21	15	18	71
Pashto to English - Dislikes	16	20	14	17	67
Pashto to English - Change	15	19	13	16	63
System - HLCT	0	27	0	15	42
System - Ease of Use	32	0	7	32	71
System - PoF	31	32	12	31	106
System - FoEE	30	31	11	30	102
System - Likes	29	30	10	29	98
System - Dislikes	28	29	9	29	95
System - Change	27	28	8	28	91

Table 63 compares the partial rankings of the Borda Count to the Evaluative Voting ratings for the S2S *TTL-Metric* pairs. Enabling *Stakeholders* to abstain from ranking an alternative for a comparable negative or "NV" rating in Evaluative Voting does not appear to change the ordering of alternatives. However, Table 63further highlights the necessity for defining a threshold as to if a *TTL-Metric* pair should remain a blueprint candidate or not. The specific issue is that the Borda Count does not enable a *Stakeholder* to differentiate between a *TTL-Metric* pair to whose evaluation they are indifferent vs. one they do not want to evaluate. For example, Table 63 shows that the *User's* scores for ASR – Automated Metrics, MT – Automated Metrics, and TTS – Likert Scores are "0" which corresponds to their lack of vote in the raw data. Referring back to the Evaluative Voting ratings captured in Table 32, the *Users* indicated a "NV" for these three *TTL-Metric* pairs. Also referring to Table 63, the *Evaluation Designer's* scores for TTS – Likert scores and the six English Transcription *TTL-Metric* pairs are given as "0" which correspond to their lack of vote in the raw data. However, Table 32 indicates that the *Evaluation Designer* rated these seven *TTL-Metric* pairs with a value of "-3" indicating a preference NOT to evaluate these pairs. This difference of opinion is not reflected in the Borda Count; the Borda Count makes no distinction between alternatives that a *Stakeholder* is indifferent to as compared to ones they are against.

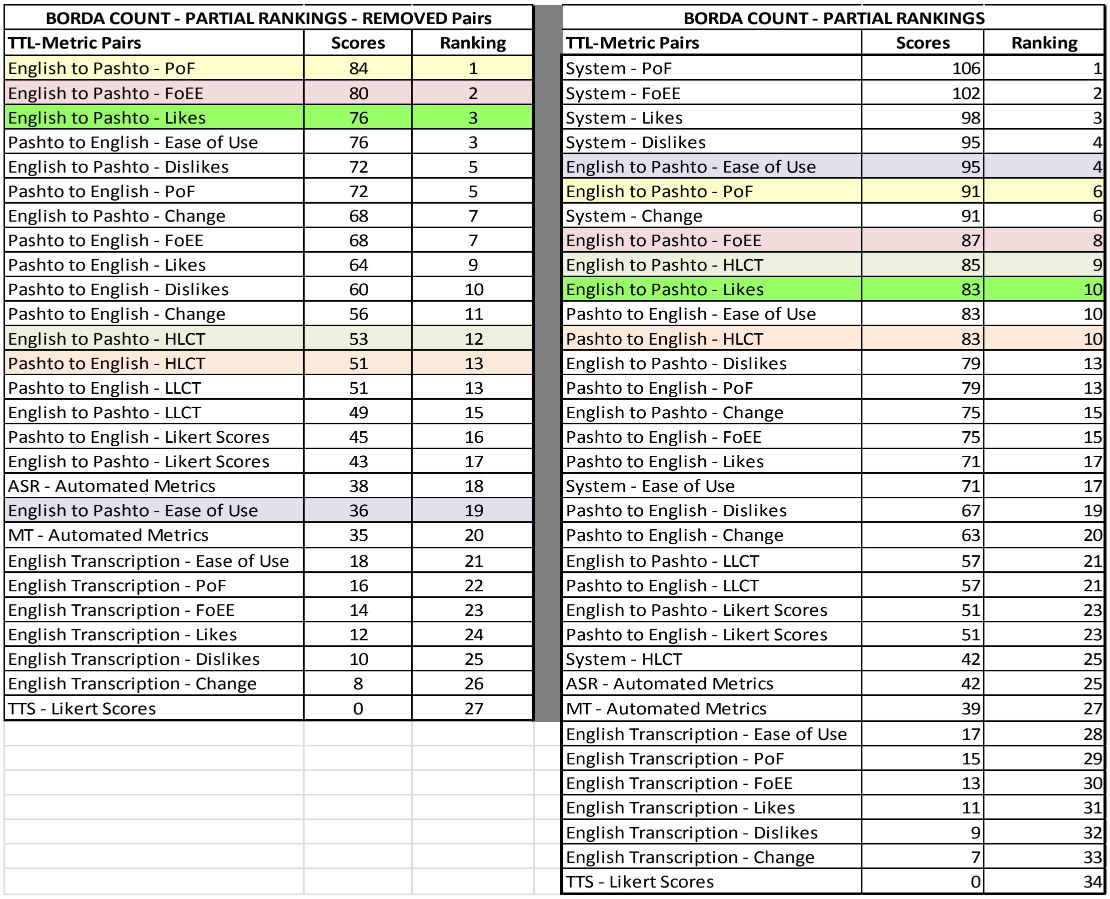
Table 63 - Comparison of Partial Borda Count Rankings to Evaluative Voting Ratings for S2S *TTL-Metric* Pairs



BORDA COUNT - PARTIAL RANKINGS				EVALUATIVE VOTING		
TTL-Metric Pairs	**Scores**	**Ranking**		**TTL-Metric Pairs**	**Scores**	**Ranking**
System - PoF	106	1		English to Pashto - HLCT	5	1
System - FoEE	102	2		Pashto to English - HLCT	5	1
System - Likes	98	3		System - HLCT	5	1
System - Dislikes	95	4		System - PoF	5	1
English to Pashto - Ease of Use	95	4		System - FoEE	5	1
English to Pashto - PoF	91	6		English to Pashto - Ease of Use	4.75	6
System - Change	91	6		English to Pashto - PoF	4.75	6
English to Pashto - FoEE	87	8		English to Pashto - FoEE	4.75	6
English to Pashto - HLCT	85	9		English to Pashto - Likes	4.75	6
English to Pashto - Likes	83	10		English to Pashto - Dislikes	4.75	6
Pashto to English - Ease of Use	83	10		English to Pashto - Change	4.75	6
Pashto to English - HLCT	83	10		Pashto to English - Ease of Use	4.75	6
English to Pashto - Dislikes	79	13		Pashto to English - PoF	4.75	6
Pashto to English - PoF	79	13		Pashto to English - FoEE	4.75	6
English to Pashto - Change	75	15		Pashto to English - Likes	4.75	6
Pashto to English - FoEE	75	15		Pashto to English - Dislikes	4.75	6
Pashto to English - Likes	71	17		Pashto to English - Change	4.75	6
System - Ease of Use	71	17		System - Ease of Use	4.75	6
Pashto to English - Dislikes	67	19		System - Likes	4.75	6
Pashto to English - Change	63	20		System - Dislikes	4.75	6
English to Pashto - LLCT	57	21		System - Change	4.75	6
Pashto to English - LLCT	57	21		ASR - Automated Metrics	4.33	22
English to Pashto - Likert Scores	51	23		MT - Automated Metrics	4.33	22
Pashto to English - Likert Scores	51	23		Pashto to English - Likert Scores	4.25	24
System - HLCT	42	25		English to Pashto - Likert Scores	3	25
ASR - Automated Metrics	42	25		English to Pashto - LLCT	2.75	26
MT - Automated Metrics	39	27		Pashto to English - LLCT	2.75	26
English Transcription - Ease of Use	17	28		~~English Transcription - Ease of Use~~	-1.75	28
English Transcription - PoF	15	29		~~English Transcription - PoF~~	-1.75	28
English Transcription - FoEE	13	30		~~English Transcription - FoEE~~	-1.75	28
English Transcription - Likes	11	31		~~English Transcription - Likes~~	-1.75	28
English Transcription - Dislikes	9	32		~~English Transcription - Dislikes~~	-1.75	28
English Transcription - Change	7	33		~~English Transcription - Change~~	-1.75	28
TTS - Likert Scores	0	34		~~TTS - Likert Scores~~	-2.67	34

Table 64 examines the impact of removing alternatives that have already been ranked. For this example, assume that the *System* is no longer available, yet the *Stakeholder Preferences* were captured prior to this knowledge. Table 64 shows the Borda Scores before and after the seven *System TTL-Metric* pairs are removed. Several pairs, common to both conditions, are highlighted to show how their ranking position may or may not have changed once the *System* pairs were removed. One significant jump is that of the English to Pashto – Ease of Use pair. It is originally tied for 4th with the System – Dislikes pair and then moves to 19th once the *System* pairs are removed.

Table 64 - Comparison of Partial Borda Count Rankings with Removed Pairs to Partial Borda Count Rankings with all Pairs



BORDA COUNT - PARTIAL RANKINGS - REMOVED Pairs		
TTL-Metric Pairs	**Scores**	**Ranking**
English to Pashto - PoF	84	1
English to Pashto - FoEE	80	2
English to Pashto - Likes	76	3
Pashto to English - Ease of Use	76	3
English to Pashto - Dislikes	72	5
Pashto to English - PoF	72	5
English to Pashto - Change	68	7
Pashto to English - FoEE	68	7
Pashto to English - Likes	64	9
Pashto to English - Dislikes	60	10
Pashto to English - Change	56	11
English to Pashto - HLCT	53	12
Pashto to English - HLCT	51	13
Pashto to English - LLCT	51	13
English to Pashto - LLCT	49	15
Pashto to English - Likert Scores	45	16
English to Pashto - Likert Scores	43	17
ASR - Automated Metrics	38	18
English to Pashto - Ease of Use	36	19
MT - Automated Metrics	35	20
English Transcription - Ease of Use	18	21
English Transcription - PoF	16	22
English Transcription - FoEE	14	23
English Transcription - Likes	12	24
English Transcription - Dislikes	10	25
English Transcription - Change	8	26
TTS - Likert Scores	0	27

BORDA COUNT - PARTIAL RANKINGS		
TTL-Metric Pairs	**Scores**	**Ranking**
System - PoF	106	1
System - FoEE	102	2
System - Likes	98	3
System - Dislikes	95	4
English to Pashto - Ease of Use	95	4
English to Pashto - PoF	91	6
System - Change	91	6
English to Pashto - FoEE	87	8
English to Pashto - HLCT	85	9
English to Pashto - Likes	83	10
Pashto to English - Ease of Use	83	10
Pashto to English - HLCT	83	10
English to Pashto - Dislikes	79	13
Pashto to English - PoF	79	13
English to Pashto - Change	75	15
Pashto to English - FoEE	75	15
Pashto to English - Likes	71	17
System - Ease of Use	71	17
Pashto to English - Dislikes	67	19
Pashto to English - Change	63	20
English to Pashto - LLCT	57	21
Pashto to English - LLCT	57	21
English to Pashto - Likert Scores	51	23
Pashto to English - Likert Scores	51	23
System - HLCT	42	25
ASR - Automated Metrics	42	25
MT - Automated Metrics	39	27
English Transcription - Ease of Use	17	28
English Transcription - PoF	15	29
English Transcription - FoEE	13	30
English Transcription - Likes	11	31
English Transcription - Dislikes	9	32
English Transcription - Change	7	33
TTS - Likert Scores	0	34

Table 64 demonstrates that the Borda Count is prone to agenda manipulation where scores can change if *TTL-Metric* pairs are added or subtracted. When generating test plans for a developing technology, it's possible for updated information to add or subtract alternative blueprint elements (not just *TTL-Metric* pairs). If *Stakeholder Preferences* have already been gathered for a given element (e.g. *TTL-Metric* pairs), then the *MRED Operator* would have to re-capture the Borda ratings from all of the *Stakeholders* for all of the *TTL-Metric* pairs. This is not the case with the 11-point linguistic ratings; a *Stakeholder* would be presented with the new *TTL-Metric* pair and rate that it from “Absolutely Reject” to “Absolutely Prefer.” Likewise, if a *TTL-Metric* pair were removed as a viable option, then it would simply be removed from MRED without the need to re-captured *Stakeholder Preferences*.

Traceability of *Stakeholder Preferences is* another concern with the Borda Count. For example, suppose a test plan required rating three *TTL-Metric* pairs (A, B, and C) where *Stakeholder Preferences* are captured according to the Borda Count. One year later, an updated test plan is being developed that also involves three *TTL-Metric* pairs, it’s a different set from the first evaluation (A, D, and E). Although *Stakeholder Preferences* (in the form of Borda ratings) were captured for A during the first test planning exercise, they are now useless considering A was originally compared to B and C, yet now is only being compared to D and E. Likewise, capturing *Stakeholder Preferences* of D and E using Borda Scores inhibits the *MRED Operator* from comparing these preferences to those of B and C from the initial test. An argument could be made that the *MRED Operator* institute “Rules of Use” for the Borda Count, yet there do not appear to be rules that address the above concerns. For example, a *Stakeholder* could be instructed to rank all of their preferred (for evaluation) *TTL-Metric* pairs in order of preference followed by those that they are indifferent. Any pairs they do not want to see evaluated should not be ranked. This is not a reasonable solution; four *TTL-Metric* pairs could be presented where one *Stakeholder* is “for evaluating” two of them “indifferent” on a third and “against” the fourth. The Borda scores here would be the same as another *Stakeholder* who is “for evaluating” all four; “for evaluating” three and “indifferent” on one; “for evaluating” one and “indifferent” on three, etc. In all of these situations, the Borda Scores would be identical even though the *Stakeholders* have different opinions.

Glossary

Actual Environment – Domain of operations that the technology is intended for usage. The evaluation team is typically extremely limited as to the data they can collect since they cannot control any environmental variables.

Autonomy Level – Refers to a specific level of authority, from “None” to “High,” that *Personnel* are afforded during an evaluation. A test plan assigns *Personnel* a specific level of authority with respect to the technology and within the environment.

Blueprint – Specifications created as the result of test planning that specify the key characteristics of a test event.

Buyer – A type of *Stakeholder* who is interested in, planning to, or already has purchase the technology.

Capability - A specific ability of a technology. A *Capability* is enabled by either a single *Component* or multiple *Components* working together.

Component – Essential part or feature of a *System* that contribute to the *System's* ability to accomplish a goal(s).

Decision-Making (DM) Autonomy – Environmental – Refers to the level of authority that the *Personnel* have in interacting with each other and the environment during an evaluation.

Decision-Making (DM) Autonomy – Technical – Refers to the level of authority that a *Tech User* has in operating the technology.

End-User – A specific type of *Tech User.* An *End-User* is the intended user of the technology. An *End-User* may also be a *User.*

Environment – Physical venue, supporting infrastructure, artifacts, and props that support a test.

Environment-based – Type of *Evaluation Scenario* that enable the *Tech User* to perform relevant activities within the *Environment* based upon an advanced *Operational Knowledge* and provided with a high-level objective.

Evaluation Designer – A type of *Stakeholder* who is responsible for creating test plans and executing the test event. In the case of MRED, the *Evaluation Designer* is usually the *MRED Operator*.

Evaluation Scenario – Govern exactly what the technology will encounter and the challenges it will face within the testing *Environment.*

Explicit Environmental Factors – Characteristics within the environment that impact the technology, influence the actions of the *Personnel* and therefore, affect the outcome of the evaluation.

Feature Complexity – Intricacy of the various features within the *Environment*.

Feature Density - Number of features that impact the technology and influence the decision-making of the *Personnel* within the test *Environment*.

Functional – A state of *Maturity* of a *Technology Test Level* where the specific *Technology Test Level* is operable, yet still under development.

Fully-Developed – A state of *Maturity* of a *Technology Test Level* where the specific *Technology Test Level* is operable and complete in its development.

Knowledge Level – Refers to a specific level, from "None" to "High," that *Personnel* possess regarding their knowledge of either the technology being considered for testing or the intended use-case environment for the technology.

Lab – Controlled *Environment* enabling evaluation designers to isolate and manipulate variables to determine how they impact performance of specific *Technology Test Levels*.

Maturity – The fitness for operation of individual *Components*, *Capabilities*, and the *System*.

Measures - A performance indicator that can be observed, examined, detected and/or perceived either manually or automatically.

Metrics – The interpretation of one or more contributing elements, e.g. measures that correspond to the degree to which a set of attribute elements affects its quality.

MRED – An interactive automatic test plan generator that takes test plan input and outputs one or more evaluation blueprints.

MRED Operator - The individual that inputs data and information into MRED.

Non-Functional – A state of *Maturity* of a *Technology Test Level* where the specific *Technology Test Level* is inoperable.

Operational Knowledge – The level of practical information and experience an individual has about the *Actual* environment, the intended use-case situations for the technology and other pre-existing technologies that the technology under test leverages and/or supports.

Participant – An individual that indirectly interacts with the technology during the test event.

Participant – An individual that indirectly interacts with the technology during an evaluation.

Personnel – Individuals that will directly or indirectly interact with the technology during a test event. Includes *Tech Users, Team Members,* and *Participants.*

Resources – Category of inputs that signify the availability of viable *Environments, Tools,* and *Personnel*.

Simulated Environment – *Environment* that is less controlled than the *Lab* limited what test variables can be controlled and manipulated. This *Environment* is more operationally-relevant than the *Lab* yet not as authentic as the *Actual Environment.*

Sponsor – A type of *Stakeholder* that funds the technology development and/or the test event (both planning and execution).

Stakeholders - Someone who has a vested interest in the technology, and therefore the evaluation.

Stakeholder Preferences - Represent the desires of an individual or group of *Stakeholders* and are provided by the evaluation *Stakeholders*.

System – A group of cooperative or interdependent *Components* forming an integrated whole to accomplish a specific goal(s).

Task/Activity-based – Type of *Evaluation Scenario* that specifies the *Tech User* complete a specific task within the *Environment* where they may use the technology as they see fit.

Team Member – Individuals that work with *Tech Users* during the evaluation as they would to realistically support the use-case scenario that the technology is immersed.

Technical Knowledge – The level of information and experience an individual has about the technology and how it should be employed to maximize success.

Technical Performance – *Metrics* related to quantitative factors (e.g. accuracy, distance, time, etc.). These metrics may be required by the program sponsor, to meet user expectations, inform the technology developers on their design, etc.

Technology-based – Type of *Evaluation Scenario* that provides specific instructions to the *Tech User* as to how they should use the technology within the *Environment.*

Technology Developer (Tech Dev) – A type of *Tech User* and a type of *Stakeholder*. They are a member of the organization that developed the technology. Those specific *Technology Developer's* that are involved in test design have at least some responsibility for designing and building the technology.

Technology State – A technology's capacity for testing.

Technology Test Levels (TTLs) – Technology's constituent *Components* and *Capabilities* along with the *System,* as a whole.

TTL-Metric Pair – A specific *TTL* that is coupled with a *Metric* that can be generated from testing this specific *TTL.*

Technology User (Tech User) – An individual that directly interacts with the technology during a test event.

Team Member – Individuals that work with *Tech Users* during the test event as they would to realistically support the use-case scenario that the technology is immersed.

Tools – The equipment and/or technology that will collect quantitative and/or qualitative data during a test event to support the generation of desired *Metrics.* Depending upon the nature of the technology and the specific *Metrics*, *Tools* may also account for the equipment required to analyze and post-process the data after the test event to produce the necessary *Metrics.*

Trained User (Trn User) - A specific type of *Tech User.* A *Trn User* is an individual selected to interact with the technology during a test event, yet is neither a *Technology Developer* nor an *End-User.*

Utility Assessments – Metrics related to qualitative factors that express the condition or status of being useful and usable to the target user population.

User – A type of *Stakeholder.* A *User* represents the population who will be or are already using the technology. A *User* may also be an *End-User.*

Validation – Shows proof of compliance of requirements. Specifically, verification indicates that the technology can meet each requirement as proven through performance of a test, analysis, inspection or demonstration.

Verification – Demonstrates that the technology accomplishes the intended purpose in the intended environment. Sucessful validation demonstrates that the technology meets the expectations of the stakeholders as shown through performance of a test, analysis, inspection or demonstration.

Bibliography

[1] Air Force Space Command, 2008, "Space and Missile Systems Center Standard – Systems Engineering Requirements and Products," SMC Standard SMC-S-001, http://www.everyspec.com.

[2] Albus, J., Barbera, A., Scott, H., & Balakirsky, S., 2006, "Collaborative Tactical Behaviors for Autonomous Ground and Air Vehicles," in *Proceedings of the Unmanned Ground Vehicle Technology VII – SPIE Conference*, **5804**, pp. 244-254.

[3] Albus, J., 2002, "Metrics and Performance Measures for Intelligent Unmanned Ground Vehicles," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[4] Alwin D. & Krosnick, J., 1985, "The Measurement of Values in Surveys: A Comparison of Ratings and Rankings," *The Public Opinion Quarterly*, **49**(4), pp. 535-552.

[5] Aoyama, H., Ishikawa, K., Seki, J., Okamura, M., Ishimura, S., & Satsumi, Y., 2007, *International Journal of Advanced Robotic Systems*, **4**(2).

[6] Arrow, K., 1978, "Extended Sympathy and the Possibility of Social Choice," *Philosophia*, **7**(2), pp. 223-237.

[7] Arrow, K., Sen, A., & Suzumura, K. (Eds.), 2002, *Handbook of Social Choice and Welfare: Volume 1*, North Holland.

[8] Ayyub, B., 2001, *Elicitation of Expert Opinions for Uncertainty and Risks*, New York, NY: CRC Press.

[9] Balaguer, B., Balakirsky, S., Carpin, S., & Visser, A., 2009, "Evaluating Maps Produced by Urban Search and Rescue Robots: Lessons Learned from RoboCup," *Journal of Autonomous Robots: Characterizing Mobile Robot Localizaiton and Mapping*, R. Madhavan et al., eds., **27**(4), pp. 449-464.

[10] Balakirsky, S., Scrapper, C., Carpin, S., & Lewis, M., 2006, "USARSim: Providing a Framework for Multi-Robot Performance Evaluation," in *Proceedings of the 2006 Performance Metrics for Intelligent Systems Workshop*.

[11] Balinski, M. & Laraki, R., 2007a, "A Theory of Measuring, Electing, and Ranking," *Proceedings of the National Academy of Sciences of the United States of America,* **104**(21), pp. 8720-8725.

[12] Balinski, M. & Laraki, R., 2007b, "Election by Majority Judgment: Experimental evidence, (mimeograph) Paris: Ecole Polytechnique, Centre National De La Recherche Scientifique, Laboratoire D-Econommetrie, Cahier, No. 2007-28, http://www.economie.polytechnique.edu/accueil/recherche/.

[13] Bialczak, R., Nida, J., Pettitt, B., & Kalpha, M., 2002, "Comparison Methodology for Robotic Operator Control Units," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[14] Billman, L. & Steinberg, M., 2007, "Human System Performance Metrics for Evaluation of Mixed-Initiative Heterogeneous Autonomous Systems," in *Proceedings of the 2007 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[15] Bostelman, R., Hong, T., Madhavan, R., Chang, T., & Scott, H., 2006, "Performance Analysis of Unmanned Vehicle Positioning and Obstacle Mapping," in *Proceedings of the Unmanned Systems Technology VIII - SPIE Conference*, **6230**(2).

[16] Brams, S. and Fishburn, P., 1978, "Approval Voting," *The American Political Science Review*, **72**(3), pp. 831-847.

[17] Calisi, D., Iocchi, L., & Nardi, D., 2008, "A Unified Benchmark Framework for Autonomous Mobile Robots and Vehicles Motion Algorithms (MoVeMA benchmarks)," *In Workshop on Experimental Methodology and Benchmarking in Robotics Research (RSS 2008).*

[18] Calisi, D., & Nardi, D., 2009, "A Performance Evaluation of Pure-Motion Tasks for Mobile Robots with Respect to World Models," *Journal of Autonomous Robots: Characterizing Mobile Robot Localization and Mapping*, R. Madhavan et al., eds., **27**(4), pp. 465-481.

[19] Cecconi, P., Franceschini, F., & Galetto, M., 2007, "The Conceptual Link between Measurements, Evaluations, Preferences and Indicators, According to Representational Theory," *European Journal of Operational Research*, **179**(1), pp. 174-185.

[20] Cohen, P. & Howe, A., 2008, "How Evaluation Guides AI Research," *AI Magazine*, **9**(4).

[21] Conley, S.A., 2009, "Test and Evaluation Strategies for Network-Enabled Systems," *International Test and Evaluation Association (ITEA) Journal*, **30**, pp. 111-116.

[22] Cook, W., 2006, "Distanced-based and ad hoc consensus models in ordinal preference ranking," *European Journal of Operational Research*, **172**, pp. 369-385.

[23] Dautenhahn, K., 2007, "Methodology & Themes of Human-Robot Interaction: A Growing Research Field," *International Journal of Advanced Robotic Systems*, **4**(1).

[24] Department of the Navy, 2004, "Naval Systems Engineering Guide," http://www.everyspec.com.

[25] Dieter, G. & Schmidt, L., 2009, Engineering Design, McGraw-Hill Higher Education, New York, NY.

[26] Djang, P. & Lopez, F., 2009, "Unmanned and Autonomous Systems Mission Based Test and Evaluation," in *Proceedings of the 2009 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

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

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

[29] Erlandson, R., 1978, "System Evaluation Methodologies: Combined Multidimensional Scaling and Ordering Techniques," *IEEE Transactions on Systems, Man, and Cybernetics*, **8**(6).

[30] Fan, Z. & Ma, J., 1999, "An Approach to Multiple Attribute Decision Making Based on Incomplete Information on Alternatives," *Proceedings of the 32nd Hawaii International Conference on System Sciences.*

[31] Felsenthal, D. & Machover, M., 2008, "The Majority Judgment Voting Procedure: a Critical Evaluation," *Homo Oeconomicus*, **25**(3/4), pp. 319-334.

[32] Fleming, M., 1952 "A Cardinal Concept of Welfare," *The Quarterly Journal of Economics*, **66**(3), pp. 366-384.

[33] Forrest, M. & Andersen, B., 1986, "Ordinal Scale and Statistics in Medical Research," *British Medical Journal - Statistics in Medicine*, **292**, pp. 537-538

[34] Forsberg, K., Mooz, H., & Cotterman, H., 2005, *Visualizing Project Management*, Third Edition, New York, NY: J. Wiley & Sons, Inc.

[35] Freedy, A., Freedy, E., DeVisser, J., Weltman, G., Kalphat, M., Palmer, D. & Coyeman, N., 2006, "A Complete Simulation Environment for Measuring and Assessing Human-Robot Team Performance," in *Proceedings of the 2006 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[36] Frost, T., Norman, C., Pratt, S., Yamauchi, B., McBride, Bill, & Peri, G., 2002, "Derived Performance Metrics and Measurements Compared to Field

Experience for the Packbot," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[37] Fundamentals of Statistics, 2010, "Scales," Retrieved from http://www.statistics4u.com/fundstat_eng/cc_scales.html#.

[38] Gao, R. & Tsoukalas, L., 2002, "Performance Metrics for intelligent Systems: An Engineering Perspective," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

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

[40] Green, S., Billingshurst, M., Chen, X., & Chase, J.G., 2008, "Human-Robot Collaboration: A Literature Review and Augmented Reality Approach in Design," *International Journal of Advanced Robotic Systems*, **5**(1), pp.1-18.

[41] Harsanyi, J., 1955, "Cardinal Welfare, Individualistic Ethics, and Interpersonal Comparisons of Utility," *The Journal of Political Economy,* **63**(4), pp. 309-321.

[42] Harvey, C. & Osterdal, L., 2010, "Cardinal Scales for Health Evaluation," *Decision Analysis*, **7**(3), pp. 256-281.

[43] Hazelrigg, 2012, Fundamentals of Decision Making – For Engineering Design and Systems Engineering, Pearson Education, Inc.

[44] Hillinger, C., 2004, "Voting and the Cardinal Aggregation of Judgments,' *Munich Discussion Paper* 2004-9, http://epub.ub.uni-muenchen.de/353/.

[45] Hubey, H., 2001, "General Scientific Premises of Measuring Complex Phenomena," *NIST Special Publication*, ISSU 970, pp. 524-532.

[46] Jacoff, A., Downs, A., Virts, A., & Messina, E., 2008, "Stepfield Pallets: Repeatble Terrain for Evaluating Robot Mobility," in *Proceedings of the 2008 Performance Metrics for Intelligent Systems (PerMIS) Workshop*, pp. 29-34.

[47] Jacoff, A. & Messina, E., 2007a, "DHS/NIST Response Robot Evaluation Exercises," *IEEE International Workshop on Safety, Security, and Rescue Robotics*.

[48] Jacoff, A. & Messina, E., 2007b, "Urban Search and Rescue Robot Performance Standards: Progress Update," in *Proceedings of the SPIE Defense and Security Conference*.

[49] Jacoff, A. & Messina, E., 2007c, "Urban Search and Rescue Robot Performance Standards: Progress Updated," *Proceedings of the Unmanned*

Systems Technology IX – SPIE Conference, G.R. Gerhart et al., eds., **6561**, pp. 65611L.

[50] Jacoff, A., Messina, E., & Evans, J., 2000, "A Standard Test Course for Urban Search and Rescue Robots," *in Proceedings of the 2000 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[51] Jacoff, A., Messina, E., & Weiss, B., 2003, "Intelligent Systems for Urban Search and Rescue: Challenges and Lessons Learned," *Proceedings of SPIE – the International Society for Optical Engineering,* **5071**.

[52] Jacoff, A., Messina, E., Weiss, B., Tadokoro, S., & Nakagawa, Y., 2003, "Test Arenas and Performance Metrics for Urban Search and Rescue Robots," *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*.

[53] Jacoff, A. & Tadokoro, S., 2005, "RoboCup 2004: Rescue Robot League," *AI Magazine*.

[54] Jacoff, A., Weiss, B., & Messina, E., 2003, "Evolution of a Performance Metric for Urban Search and Rescue Robots," in *Proceedings of the 2003 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[55] Jian-Jun, Z., Ru-Qing, R., Wei-Jun, Z., Xin-Hua, W., & Jun, Q., 2007, "Research on Semi-automatic Bomb Fetching for an EOD Robot," *International Journal of Advanced Robotic Systems*, **4**, pp. 247-252.

[56] Lacaze, A., Murphy, K., & DelGiorno, M., 2002, "Autonomous Mobility for the Demo III Experimental Unmanned Vehicles," in *Proceedings of the AUVSI 2002 Conference*.

[57] Leedom, D., 2003, "Advancing the State-of-the-Art in Intelligent Systems: Scientific Rigor in Our Methods of Experimentation," *Plenary Presentation at the 2003 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[58] Ma, J., Lu, J., & Zhang, G., 2009, "Decider: A fuzzy multi-criteria group decision support system," *Knowledge-Based Systems*, **23**(1), pp. 23-31.

[59] Mankins, J., 1995, "Technology Readiness Levels," *Advanced Concepts Office, Office of Space Access and Technology, NASA*, White Paper.

[60] Mankins, J., 2009, "Technology Readiness Assessments: A retrospective," *Acta Astronautica*, **65**(9-10), pp. 1216-1223.

[61] Messina, E., 2009, "Robots to the Rescue," *Crisis Response Journal*, **5**(3), pp. 42-43.

[62] Messina, E. & Jacoff, A.S., 2007, "Measuring the Performance of Urban Search and Rescue Robots," *IEEE Conference on Homeland Security Technologies*.

[63] Model Based Systems Engineering (MBSE) Initiative, International Council on Systems Engineering (INCOSE), 2008, "Survey of Model-Based Systems Engineering (MBSE) Methodologies, INCOSE-TD-2007-003-01.

[64] Morrison, M., 2002, "Aggregation Biases in Stated Preference Studies," *Australian Economic Papers*, **39**(2), pp. 215-230.

[65] Morse, E., Steves, M., & Scholtz, J., 2005, "Metrics and Methodologies for Evaluating Technologies for Intelligence Analysts," *International Conference on Intelligence Analysis*.

[66] Mosteller, F. & Tukey, J., 1977, *Data Analysis and Regression*, Boston: Addison-Wesley.

[67] NASA Systems Engineering Handbook, 2007, NASA/SP-2007-6105, Rev 1, http://ntrs.nasa.gov/.

[68] National Opinion Research Center, 1982, "General Social Surveys, 1972-82: Cumulative Codebook," Chicago: National Opinion Research Center, Univeristy of Chicago.

[69] New Product Development, 2011, Retrieved on January 3, 2012, http://en.wikipedia.org/wiki/Product_development.

[70] Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., & Burion, S., 2005, "Human-Robot Teaming for Search and Rescue," *IEEE Pervasive Computing: Mobile and Ubiquitous Systems*, pp. 72-78.

[71] O'Brien, R., 1979, "The Use of Pearson's R with Ordinal Data," *American Sociological Review*, **44**(5), pp. 851-857.

[72] Office of the Deputy Assistant Secretary of Defense (ODASD) – Systems Engineering, Retrieved on April 29, 2012, http://www.acq.osd.mil/se/.

[73] Olcer, A. & 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**, pp. 93-114.

[74] Olsen, D. & Goodrich, M., 2003, "Metrics for Evaluating Human-Robot Interactions," in *Proceedings of the 2003 Performance Metrics for Intelligent Systems (PerMIS) Workshop.*

[75] Pei-you, C. & Yi-ling, L., 2009, "Negotiation Model Based on Uncertainty Multi-Attribute Decision Making," *Control and Decision Conference*, pp. 1553-1556.

[76] Ramanathan, R. & Ganesh, L., 1994, "Group Preference Aggregation Methods Employed in AHP: An Evaluation and an Intrinsic Process for Deriving Members' weightages," *European Journal of Operational Research*, **79**(2), pp. 249-265.

[77] Remley, K., Koepke, G., Messina, E., Jacoff, A., & Hough, G., 2007, "Standards Development for Wireless Communications for Urban Search and Rescue Robots," *International Symposium on Advanced Radio Technologies (ISART) 2007*, pp. 66-70.

[78] Saari, D., 1990, "The Borda Dictionary" *Social Choice and Welfare.* **7**, pp. 279-317.

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

[80] Sauser, B., Ramirez-Marquez, J., Verma, D., & Gove, R., 2006, "From TRL to SRL: The Concept of Systems Readiness Levels," *Conference on Systems Engineering Research*.

[81] Schenk, J., & Wade, R., 2008, "Robotic Systems Technical and Operational Metrics Correlation," in *Proceedings of the 2008 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[82] Schlenoff, C., Steves, M., Weiss, B., Shneier, M., & Virts, A., 2007, "Applying SCORE to Field-Based Performance Evaluations of Soldier-Worn Sensor Technologies," *Journal of Field Robotics – Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems*, 24, pp. 671-698.

[83] Schlenoff, C., Weiss, B., & Steves, M., 2010, "Lessons Learned in Evaluating DARPA Advanced Military Technologies," in *Proceedings of the 2010 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[84] Schlenoff, C., Weiss, B., Steves, M., Sanders, G., Proctor, F., & Virts, A., 2009, "Evaluating Speech Translation Systems: Applying SCORE to TRANSTAC Technologies," in *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[85] Schlenoff, C., Weiss, B., Steves, M., Virts, A., & Shneier, M., 2006, "Overview of the First Advanced Technology Evaluations for ASSIST," in *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[86] Scholtz, J., 2002, "Evaluation Methods for Human-System Performance of Intelligent Systems," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[87] Scholtz, J., 2005, "Implementation of a Situation Awareness Assessment Tool for Evaluation of Human-Robot Interfaces," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, **35**(4).

[88] Scholtz, J., Antonishek, B., & Young, J., 2004, "Evaluation of Human-Robot Interaction in the NIST Reference Search and Rescue Test Arenas," in *Proceedings of the 2004 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[89] Scholtz, J.C., Antonishek, B., & Young, J.D., 2005, "A Field Study of Two Techniques for Situation Awareness for Robot Navigation in Urban Search and Rescue," *Proceedings of the IEEE Ro-Man Conference*.

[90] Scholtz, J.C., Theofanos, M.F., & Antonishek, B., 2006, "Development of a Test Bed for Evaluating Human-Robot Performance for Explosive Ordnance Disposal Robots," in *Proceedings of the 1st Annual Conference on Human-Robot Interaction*.

[91] Scrapper, C., Mahavan, R., Jacoff, A., Lakaemper, R., Censi, A., Godil, A., & Wagan, A., 2008, "Quantitative Assessment of Robot-generated Maps," *Performance Evaluation and Benchmarking of Intelligent Systems*, R. Madhavan et al., eds., pp. 221-248.

[92] Scrapper, C., Mahavan, R., Jacoff, A., Lakaemper, R., Censi, A., Godil, A., & Wagan, A., 2009, "Quantitative Assessment of Robot-generated Maps," *Performance Evaluation and Benchmarking of Intelligent Systems*, R. Madhavan et al., eds., pp. 221-248.

[93] SE Handbook Working Group, International Council on Systems Engineering (INCOSE), 2010, "Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities," C. Haskins (Ed.). San Diego, CA.

[94] Sen, A., 1977, "Social Choice Theory: A Re-Examination," *Econometrica*, **45**(1), pp. 53-89.

[95] Sipser, M., 1997, *Introduction to the Theory of Computation*. Boston, MA: PWS Publishing Company.

[96] Stanton, B., Antonishek, B., & Scholtz, J., 2006, "Development of an Evaluation method for Acceptable Usability," in *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[97] Stevens, S., 1946, "On the Theory of Scales of Measurement," *Science*, **103**(2684), pp. 677-680.

[98] Steves, M.P. & Scholtz, J., 2005, "A Framework for Evaluating Collaborative Systems in the Real World," *Proceedings of the 38th Hawaii International Conference on System Sciences*

[99] Sukhatme, G. & Bekey, G., 1995, "An Evaluation Methodology for Autonomous Mobile Robots for Planetary Exploration," *Proceedings of the First ECPD International Conference on Advanced Robotics and Intelligent Automation,* pp. 558-563.

[100] Test Plan, 2011, Retrieved January 3, 2012, from http://en.wikipedia.org/wiki/Test_plan

[101] Tetlay, A., and John, P., 2009, "Determing the Lines of System Maturity, System Readiness and Capability Readiness in the System Development Lifecycle," *7th Annual Conference on Systems Engineering Research 2009*.

[102] Thompson, M., 2008, "Testing the Intelligence of Unmanned Autonomous Systems," *International Test and Evaluation Association (ITEA) Journal*, 29(4), pp. 380-387.

[103] Thurston, D., 2001, "Real and Misconceived Limitations to Decision Based Design with Utility Analysis," *Journal of Mechanical Design*. **123**, pp. 176-182.

[104] US Department of Transportation, 2007, "Systems Engineering for Intelligent Transportation Systems – An Introduction for Transportation Professionals," http://ops.fhwa.dot.gov/publications/seitsguide/index.htm.

[105] Van Praag, B., 1991, "Ordinal and cardinal utility," *Journal of Econometrics*, **50,** pp. 69-89.

[106] Vasiljev, S., (unpublished manuscript) 2008, "Cardinal Voting: the Way to Escape the Social Choice Impossibility," SSRN 1116545, http://ssrn.com/abstract=1116545.

[107] Velleman, P. & Wilkinson, L., 1993, "Nominal, Ordinal, Interval, and Ratio Typologies are Misleading," *The American Statistician*, **47**(1), pp. 65-72.

[108] Weiss, A., Bernhaupt, R., Lankes, M., & Tscheligi, M., 2009, "The USUS Evaluation Framework for Human-Robot Interaction," *in AISB2009: Proceedings of the Symposium on New Frontiers in Human-Robot Interaction*.

[109] Weiss, B. & Menzel, M., 2010, "Development of Domain-Specific Scenarios for Training and Evaluation of Two-Way, Free Form, Spoken Language Translation Devices," *International Test and Evaluation Association (ITEA) Journal*, **30**(1), pp. 39-47.

[110] Weiss, B. & Schlenoff, C., 2008, "Evolution of the SCORE Framework to Enhance Field-Based Performance Evaluations of Emerging Technologies," *Proceedings of the 2008 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[111] Weiss, B. & Schlenoff, C., 2009, "The Impact of Scenario Development on the Performance of Speech Translation Systems Prescribed by the SCORE Framework," *Proceedings of the 2009 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[112] Weiss, B. & Schlenoff, C., 2010, "Performance Assessments of Two-Way, Free-Form, Speech-to-Speech Translation Systems for Tactical Use," in *Proceedings of the 2010 International Test and Evaluation Association (ITEA) Annual Symposium*.

[113] Weiss, B. & Schlenoff, C., 2011, "Performance Assessments of Two-Way, Free-Form, Speech-to-Speech Translation Systems for Tactical Use,' *International Test and Evaluation Association (ITEA) Journal,* **32**(1), pp. 69-75.

[114] Weiss, B., Schlenoff, C., Sanders, G., Steves, M., Condon, S., Phillips, J., & Parvaz, D., 2008, "Performance Evaluation of Speech Translation Systems," in *Proceedings of the 6th edition of the Language Resources and Evaluation Conference*.

[115] Weiss, B., Schlenoff, C., Shneier, M., & Virts, A., 2006, "Technology Evaluations and Performance Metrics for Soldier-Worn Sensors for ASSIST," in *Proceedings of the 2006 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[116] Weiss, B. & Schmidt, L., 2010, "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*.

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

[118] Weiss, B. & Schmidt, L., 2011b, "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)*.

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

[120] Weiss, B. & Schmidt, L., 2012, "Multi-Relationship Evaluation Design: Modeling an Automatic Test Plan Generator," To Appear *Proceedings of the 2012 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[121] Weiss, B., Schmidt, L., Scott, H., & Schlenoff, C., 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)*.

[122] Whitcomb, C., Palli, N. & Azarm, S., 1999, "A Prescriptive Production-Distribution Approach for Decision Making in New Product Design," *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews,* **29**(3), pp. 336-348.

[123] Wright, B. & Linacre, J., 1989, "Observations are Always Ordinal; Measurements, However, Must be Interval," *Archives of the Physical Medicine and Rehabilitation,* **70**(12), pp. 857-860.

[124] Wu, D., 2009, "Performance Evaluation: An Integrated Method Using Data Envelopment Analysis and Fuzzy Preference Relations," *European Journal of Operational Research*, **194**, pp. 227-235.

[125] 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), pp. 1500-1511.

[126] Yakowitz, D. & Lane, L., 1993, "Multi-attribute Decision Making: Dominance with Respect to an Importance Order of the Attributes," *Applied Mathematics and Computation,* **54**(2-3), pp. 167-181.

[127] Yanco, H., 2001, "Designing Metrics for Comparing the Performance of Robotic Systems in Robot Competitions," in *Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop*.

[128] Yue., R., Xiao, J., Li, K., Du, J., & Wang, S., 2010, "Design and Performance Analysis of Retractable-Claw Wheels for Field Robots," *International Journal of Robotics and Automation,* **25**(3), pp. 250-258.

[129] Zhang, H., Wang, W., Deng, Z., Zong, G., & Zhang, J., 2006, "A Novel Reconfigurable Robot for Urban Search and Rescue*," International Journal of Advanced Robotic Systems*, **3**(4), pp. 359-366.

[130] Zhang, Q., Cui, W., & Sha, Y., 2009, "Multiple Attribute Decision Making Based on Fuzzy Selected Subset and Linguistic Variables," *2009 International Conference on Research Challenges in Computer Science*.

[131] Zhang, Y., Eldershaw, C., Yim, M., Roufas, K., & Duff, D., 2002, "A Platform for Studying Locomotion Systems: Modular Reconfigurable Robots," in *Proceedings of the 2002 Performance Metrics for Intelligent Systems (PerMIS) Workshop.*