Expert Systems with Applications xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Multi-relationship evaluation design: Formalization of an automatic test plan generator

Brian A. Weiss [a,1], Linda C. Schmidt [b,*]

[a] National Institute of Standards and Technology, 100 Bureau Drive, MS 8230 Gaithersburg, MD 20899, USA
[b] University of Maryland, Glenn L. Martin Hall, College Park, MD 20740, USA

ARTICLE INFO

Keywords:
Testing
Evaluation
Performance assessment
Test plans
Stakeholder preferences

ABSTRACT

The number of intelligent and advanced technologies in the manufacturing, military and homeland security industries is increasing. Evaluating these technologies is a critical step in their development cycle. Test designers have put forth considerable effort in creating methods to accelerate the test-plan development process. The multi-relationship evaluation design (MRED) methodology is an automatic test plan generator. MRED collects multiple inputs, processes them interactively with a test designer and outputs evaluation blueprints that specify key test-plan characteristics. This paper describes MRED's process and presents the mathematical representations used by MRED and the stakeholder preference handling strategy. A robot arm is the example used to demonstrate MRED.

1. Introduction

Intelligent and advanced technologies are currently deployed in manufacturing, military, homeland security and automotive sectors of industry. Evaluation events are critical steps 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 technology's functionality. Test exercises are an opportunity to realize the technology's current abilities and limitations and inform future test efforts. Here, "the term test refers to a planned evaluation event or exercise focused on capturing data to generate performance metrics of a specific technology under scrutiny" (Weiss, Schmidt, Scott, & Schlenoff, 2010). Evaluation designers expend considerable effort in creating methods to accelerate the test-plan development process (Sukhatme & Bekey, 1995). These efforts are most visible when designers must produce intricate test plans to evaluate intelligent and advanced technologies. Numerous researchers have documented the importance of evaluations and how they guide artificial intelligence (AI) system research and development (Cohen & Howe, 2008; Gao & Tsoukalas, 2002).

The multi-relationship evaluation design (MRED) methodology is an automatic test plan generator that will allow evaluation designers to accelerate the test-plan development process. MRED collects multiple inputs from various source categories and automatically outputs evaluation blueprints that specify key test-plan characteristics. MRED input comes from stakeholders, who provide not only their preferences, but also technology state assessments and the resources available for testing. This information along with the relationships among these inputs is combined as input into the MRED algorithm.

Stakeholder preference capture and processing is a critical function of MRED and a significant point of emphasis in this paper. These subjective preferences are supported by each Stakeholder's knowledge of the facts. Providing preferences to ultimately select evaluation blueprints is different than what is encountered in product development. Each class of Stakeholders could potentially select entirely unique test-plan blueprints with very different test elements. This is not the case in product development where preferences provided on constituent attributes (product size, weight, etc.) all contribute to the same overriding goal of profit for the business. In product development, the decision-makers are usually all employees of the same entity. In the typical development of advanced technology evaluation, input from different Stakeholders (often with competing interests) is collected and processed for decision-making. This effort leverages evaluative voting discussed in Section 4.5 (Hillinger, 2004).

This paper is outlined as follows: Section 2 presents the overall MRED methodology; Section 3 describes the example to which MRED will be applied; Section 4 mathematically formalizes the MRED process; and Section 5 concludes the paper.

2. Multi-relationship evaluation design

MRED is an interactive algorithm that processes input categories and outputs one or more constituent test plan elements in

* Corresponding author. Tel.: +1 301 405 0417.
E-mail addresses: brian.weiss@nist.gov (B.A. Weiss), lschmidt@umd.edu (L.C. Schmidt).
[1] Tel.: +1 (301) 975 4373.

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.083

one or more evaluation blueprints (Fig. 1) (Weiss & Schmidt, 2012). MRED leverages the relationships across the inputs and the influences the inputs have on the outputs. The overall methodology was proposed in Weiss and Schmidt (2011a) while the output blueprint evaluation elements were defined in Weiss et al. (2010), Weiss and Schmidt (2010). The relationships between specific inputs and outputs were presented in Weiss and Schmidt (2011b, 2011c). This section presents the MRED model inputs, and output blueprint elements as shown in Fig. 1.

2.1. Input categories

MRED relies upon information, data, and preferences from five categories (shown in Fig. 1).

2.1.1. Technology test levels (TTLs)

TTLs are defined as the technology's constituent *Components* and *Capabilities* along with the *System*, in its entirety (Weiss et al., 2010; Weiss, 2012). They are defined as:

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

2.1.2. Metrics

Pertinent *Metrics* are also input according to the input *TTLs*. *Metrics* fall into one of two groups:

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

2.1.3. Technology state – maturity

MRED defines *Technology state* as a technology's fitness for testing. *Technology state* is described by the element of *Maturity* which is identified for each individual *TTL* (Weiss & Schmidt, 2011c). *Maturity* is defined with respect to MRED as: the state of development 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. *Maturity* must be input into MRED for a *TTL* to be considered for evaluation. The *Maturity* level is defined 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 be defined as one of the following:

- *Immature* – The *technology test level* being tested has yet to be developed or is still in the process of being developed.
- *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 associated behaviors available.

Additional details on the *technology state* elements including *maturity* can be found in Weiss (2012), Weiss and Schmidt (2011c).

2.1.4. Test resources

This category of inputs signifies the availability of the viable *environments, tools*, and *personnel*. They are defined as:

- *Environment* – The physical venue, supporting infrastructure, artifacts, and props that will support the test(s). The *environment* can influence the behavior of the personnel and can restrict which *TTLs* can be tested. MRED defines three different *environments*: *lab, simulated*, and *actual*.
- *Tools* – The tools, equipment, and/or technology that will collect quantitative and/or qualitative data during the test. *tools* also include the means to produce the necessary metrics from the captured data. *Tools* are defined based upon the nature of the *metrics* they are used to capture. This means that an evaluation may call for *technical performance tools* and/or *utility assessment tools*.
- *Personnel* – Individuals that will use the technology and indirectly interact with the technology. These include:
  - *Tech users* – *Personnel* that use the technology during the evaluation. These individuals are either identified as *end-users, trained users*, or *technology developers*.
  - *Team members* – Individuals that interact with *tech users* during the evaluation to realistically support the scenario that the technology is immersed.
  - *Participants* – Individuals that indirectly interact with the technology during an evaluation.

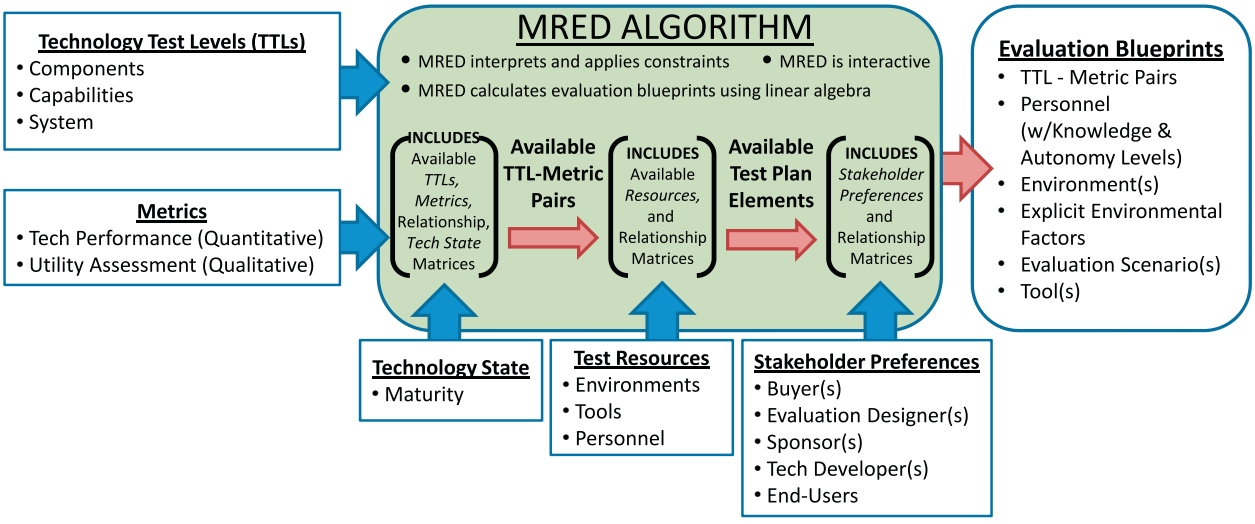

Fig. 1. MRED model with input and output (Weiss, 2012).

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Test resources have been detailed in Weiss (2012), Weiss et al. (2010), Weiss and Schmidt (2010).

2.1.5. Stakeholder preferences

This last category includes preferences from five specific individuals (or groups) and was originally detailed in Weiss and Schmidt (2011b) 0. The five individuals are:

- *Buyer(s)* – *Stakeholder(s)* purchasing the technology
- *Evaluation designer(s)* – *Stakeholder(s)* creating the test plans by determining and identifying the MRED inputs
- *Sponsor(s)* – *Stakeholder(s)* paying for the technology development and/or evaluation
- *Technology developer(s)* – *Stakeholder(s)* designing and constructing the technology
- *User(s)* – *Stakeholder(s)* that will be or is already using the technology

Stakeholders provide their preferences with respect to the *TTL-Metric* pairs,[2] *environments, tools, personnel, explicit environmental factors*, and *evaluation scenarios* (Weiss & Schmidt, 2010, 2011a).

2.2. Output evaluation blueprints

Each set of blueprints will include one (or more) *TTL-Metric* pairs, an *Environment* for testing, *Tools* to support the collection and analysis of data to generate the corresponding *Metric(s)* (Weiss & Schmidt, 2010), *personnel* including those who will test the technology and those who will execute the evaluations (Weiss et al., 2010; Weiss & Schmidt, 2011b), *knowledge* and *autonomy levels* for those *Personnel* who will directly and indirectly interact with the technology during the test (Weiss & Schmidt, 2011b) *evaluation scenarios* describing the type of exercises in which the technology will be immersed and *explicit environmental factors* which indicate the levels of *feature complexity* and *feature density* within the *environment* (Weiss & Schmidt, 2010).

2.3. Relationships

Relationships are a core element to MRED and are defined among the various inputs and between inputs and outputs. Relationships defined between the inputs and outputs have been discussed extensively in previous work. Several of the relationships among the inputs include:

- *Components* and *capabilities* – This relationship is the influence each *component* has on performing or realizing the *capabilities* within the *system*. It is defined in a single binary matrix.
- *Metrics* and *TTLs* – This relationship is defined in two binary matrices and indicates which *metrics* are applicable to each *TTL*. The first matrix (U_1) represents which *technical performance metrics* can be produced when testing the *TTLs*. The second matrix (U_2) represents which *quantitative assessment metrics* can be produced when testing the *capabilities* and the *system*.
- *TTLs* and *environments* – This relationship indicates which of the available *environments* each of the *TTLs* can be evaluated within. It is defined in three binary matrices. The first matrix (X_1) represents which *components* and *capabilities* can be evaluated within the *lab environments*; the second matrix (X_2) represents which *TTLs* (among all three types) can be evaluated within the *simulated environments;* and the third matrix (X_3) indicates which *capabilities* and the *system* can be evaluated within the *actual environment(s)*.

[2] *TTL-metric*pairs are specific *Technology test levels*and *metrics* that are coupled together. Multiple *TTLs*can be coupled with the same *metrics* and vice versa.

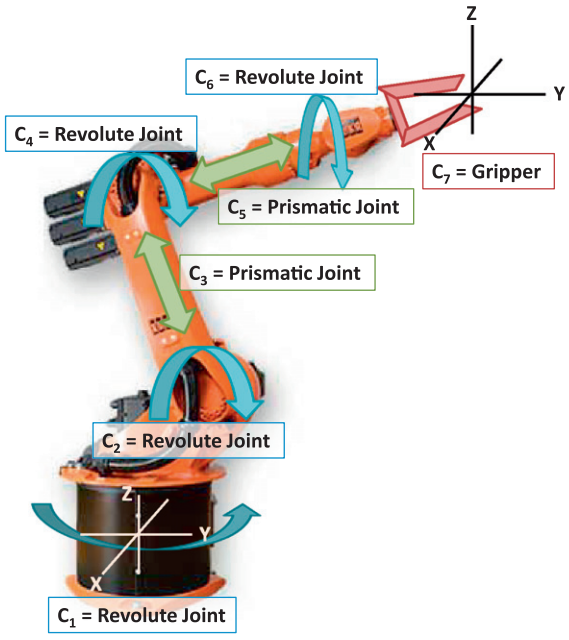

Fig. 2. Robotic arm[14].

- *Metrics* and *tools* – This relationship denotes which *tools* are necessary to collect data in support of the candidate *metrics* and is defined in two binary matrices. Y_1, the first matrix, indicates which *tools* are necessary to collect data in support of the candidate *technical performance metrics*; Y_2 denotes which *tools* are required to collect data in support of the candidate *utility assessment metrics*.

Before presenting the formalized MRED model an illustrative example is introduced. This example is refined throughout the formalization to highlight each step in the process.

3. Example – robotic arm

An example robotic arm is used to present the MRED process. The example robotic arm, shown in Fig. 2, is a *System* with 7 *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 7 *Components* function to provide 7 *Capabilities* (P_1, P_2, and P_3 are translation in *X*, *Y*, and *Z* of the end-effector; P_4, P_5, and P_6 are roll, pitch, and yaw of the end-effector; and $P_7$is grasping). Note that the reference frame of these *Capabilities* is the coordinate frame at the tool point with respect to the base shown in Fig. 2. The *TTLs, Metrics*, and *Technology State* elements are discussed in Section 4 as MRED is applied to this example.

This robotic arm example is an abstract example and is not indicative of any specific system currently on the market. The original image used in Fig. 2 is modified to illustrate the MRED. The example, with stated *Components and Capabilities*, is designed to illustrate MRED. Likewise, this simple example does not include the many other factors that would likely be assessed in developmental testing (e.g., user interface, controller).

4. Process formalization

MRED's overall process is formally presented in this section. Each of the subsections provides detail on one or more of the specific steps within MRED. The robotic arm example is used to highlight the process.

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Table 1
TTLs and *metrics* defined for robotic arm.[15]

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)	
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		Satisfaction			
β = 3								

4.1. TTLs, metrics, and relationships

MRED begins with the *MRED operator*[3] inputting the available *TTLs* and corresponding *metrics* (both *technical performance* and *utility assessment*). The sets of τ *components* ($\mathbf{c}$), φ *capabilities* ($\mathbf{p}$) and the *system* ($\mathbf{s}$) are defined as:

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

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

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

The sets of α *technical performance metrics* and β *utility assessment metrics* are expressed as:

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

$$\mathbf{a} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots \mathbf{a}_\beta\} \tag{5}$$

Table 1 applies these definitions to the robotic arm example. Note that this is a simplified example and presents a subset of the *Metrics* that could be potentially captured in evaluations of this technology. The first row of the table, "c=" defines the seven *Components* while the third row of the table denotes the seven *Capabilities* of the robotic arm. The fifth row, "t=" specifies eight example *Technical Performance Metrics* while the seventh row, "a=" states three example *Utility Assessment Metrics*.

This small quantity of *Metrics* is defined to maintain the simplicity of the robotic arm example. As similarly stated in Section 3 with *Components and Capabilities*, a more realistic example would contain a greater amount of quantitative and qualitative *Metrics*.

Next, the *MRED operator* defines two sets of relationships; the *components* and *capabilities* relationship matrix and the *metrics* and *TTLs* relationship matrices. The *components* and *capabilities* relationship matrix, *O*, is defined:

$$O = \begin{array}{c} \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_\tau \end{array} \begin{array}{c} \begin{array}{cccc} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_\varphi \end{array} \\ \begin{bmatrix} \mathbf{o}_{11} & \mathbf{o}_{12} & \cdots & \mathbf{o}_{1\varphi} \\ \mathbf{o}_{21} & \mathbf{o}_{22} & \cdots & \mathbf{o}_{2\varphi} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{o}_{\tau 1} & \mathbf{o}_{\tau 2} & \cdots & \mathbf{o}_{\tau\varphi} \end{bmatrix} \end{array} \tag{6}$$

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 2 presents the corresponding *O* matrix for the robotic arm example.

Two *metrics* and *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 3 presents the U_1 relationship matrix while U_2 would appear similarly.

[3] The term *MRED operator* is defined as the individual that inputs data and information into MRED. This is usually the *evaluation designer* facilitator who is guiding the blueprint generation process.

Table 2
O Relationship matrix for robotic arm.

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

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

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

Fig. 3 presents a screenshot of the Matlab interface that was developed to realize the interactive element of this effort. This screen capture presents the input of the *O*, U_1 and U_2 matrices.

4.2. Technology state

The *MRED Operator* now inputs the *technology state* information (*maturity*) for the *components* according to Fig. 4. This step also includes inputting the *technology state* information for the *capabilities* and the *system*, if explicitly known. *maturity* ($\mathbf{m}$) 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*) (Weiss & Schmidt, 2011a). Table 4 presents the *Maturity* vectors for the robot arm where the first row indicates the values $\mathbf{m}_1$ vector, the second row indicates the values of the $\mathbf{m}_2$ vector, and the third row indicates $\mathbf{m}_3$.

When *maturity* is unknown for the *capabilities* and *system* ($\mathbf{m}_2, \mathbf{m}_3$), MRED calculates these vectors. The *maturity* for the *capabilities* is presented in the normalized Eq. (8).

$$\mathbf{m}_{2_j} = \mathbf{m}_1 \mathrm{Col}_i(O) / \sum_{i=1}^{\tau} o_{ij} \tag{9}$$

Like $\mathbf{m}_1$, values of $\mathbf{m}_2$ will range from 0 to 1. *maturities* less than 1 indicate an *immature capability*, which may or may not be available for testing given its specific state. A value of 1 indicates a *fully devel-*

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Table 3
U_1 relationship matrix for robotic arm.

Metrics – technical performance	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)
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

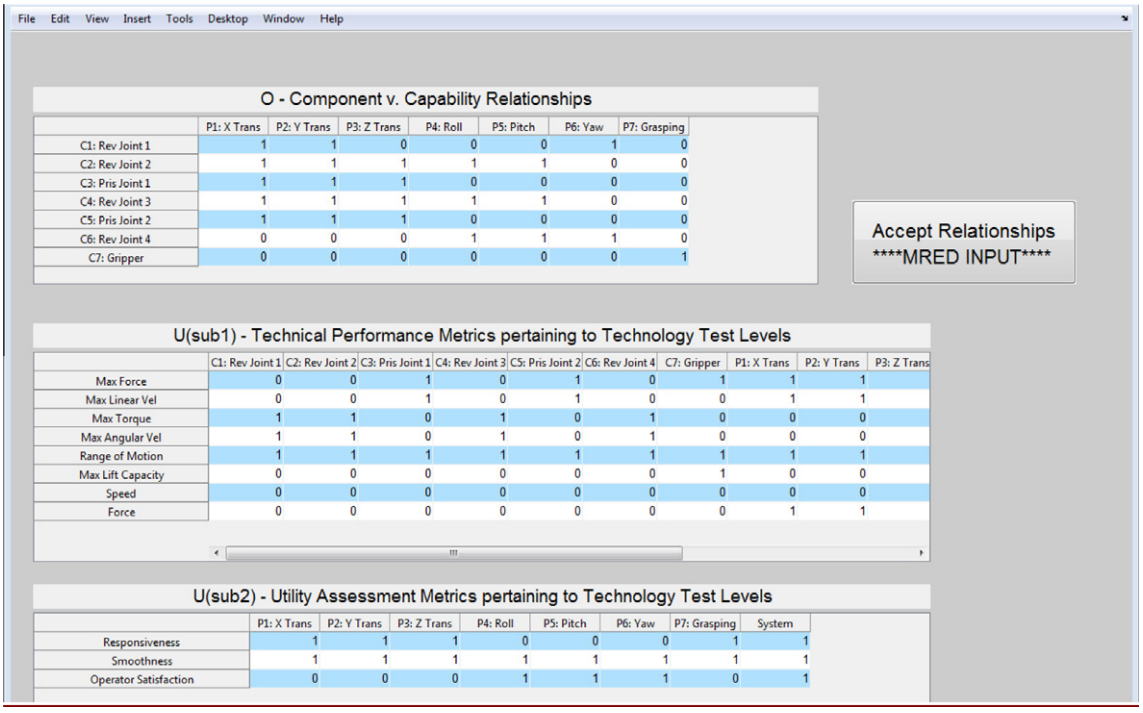

Fig. 3. Matlab interface showing O, U_1, and U_2 Inputs.

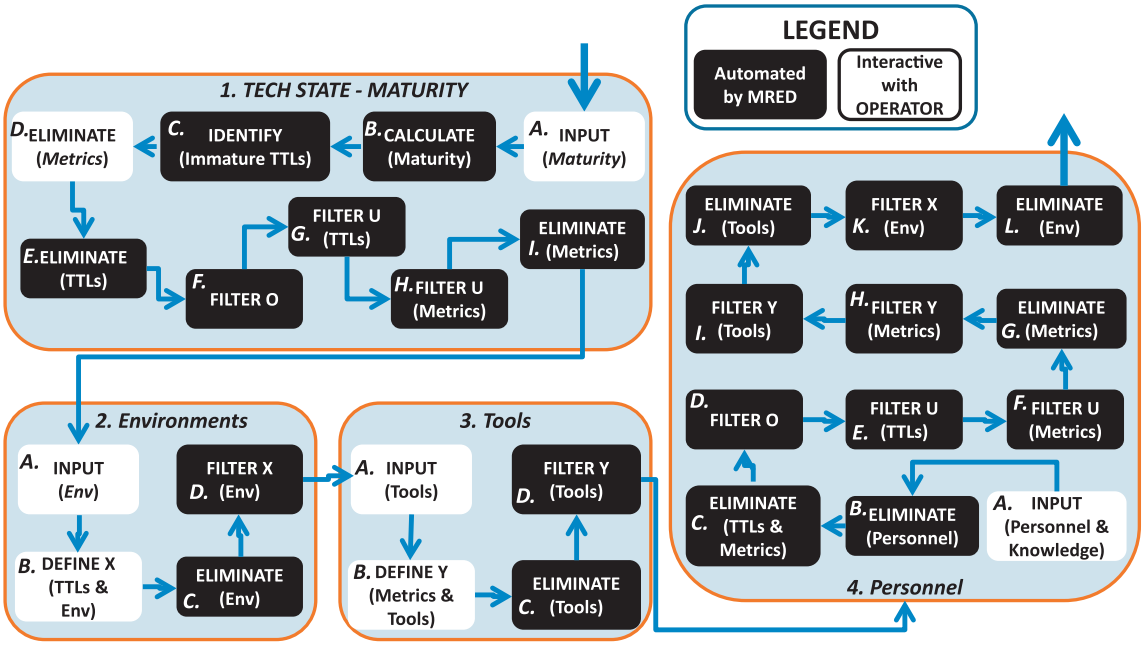

Fig. 4. MRED constraint handling and element filtration process.

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Table 4
Maturity for robotic arm.

m_1=	1	1	1	1	1	0	0
m_2=	1.00	1.00	1.00	0.67	0.67	0.67	0.00
m_3=	0.71						

oped capability which denotes that a *capability* is ready for all potential tests.

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 eq. (9).

$$\mathbf{m}_3 = \left(\sum_{i=1}^{\varphi} \mathbf{m}_{2_i}\right)/\varphi \tag{10}$$

The *system maturity* ranges from 0 to 1, similar to the *capability maturities*, where its values are interpreted in the same manner. Table 4 presents the **m**vectors for the robotic arm example. Note that the *capability* and *system maturities* were calculated using Eqs. (8) and (9).

Next, MRED alerts the *MRED operator* which *TTLs* are *immature* (Step C of Box 1 in Fig. 4). The *MRED operator* removes the relationships between those *metrics* and *TTLs* in U_1 and U_2if a *TTL's* immaturity does not allow the corresponding *metric* to be captured (Step D of Box 1 in Fig. 4).

4.3. Constraint-handling process

Rejecting candidate *TTLs*, or any other blueprint element, is a non-trivial process that requires several steps. One way to consider this process is 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 throughout Section 4. 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.

Table 5
X_1 relationship matrix for robotic arm.

X_1	Lab environments		
Components and capabilities	Controls lab	Robotics lab	Force/torque lab
C_1	1	0	1
C_2	1	0	1
C_3	1	0	1
C_4	1	0	1
C_5	1	0	1
P_1	0	1	0
P_2	0	1	0
P_3	0	1	0
P_4	0	1	0
P_6	0	1	0

- 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.

Fig. 4 presents MRED's constraint handling and element filtration process as the *Technology state* and available *resources* (*environments, tools*, and *personnel*) are input. since *maturity* has been defined for all *TTLs* at this point, the steps (A through E) in box 1 (Fig. 4) are executed.

4.4. Resources

4.4.1. Environments

The process outlined in Fig. 4 continues into box 2. The *MRED operator* now 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 four specific steps (A. INPUT, B. DEFINE X, C. ELIMINATE, and D. FILTER X) in box 2 are engaged. Eq. (10) presents the X_1 matrix, which is defined as the binary relationship matrix between *Components* and *Capabilities* to the available *Lab Environments*. Similar binary relationship matrices, X_2 and X_3, defined for the *Simulated and Actual Environments*, are not shown for brevity.

$$X_1 = \begin{array}{c} \\ c_1 \\ \vdots \\ c_\tau \\ p_1 \\ \vdots \\ p_\varphi \end{array} \begin{array}{c} \begin{array}{cccc} e_{1_1} & e_{1_2} & \cdots & e_{1_\gamma} \end{array} \\ \begin{bmatrix} x_{1_{1,1}} & x_{1_{1,2}} & \cdots & x_{1_{1,\gamma}} \\ \vdots & \vdots & \ldots & \vdots \\ x_{1_{\tau',1}} & x_{1_{\tau',1}} & \cdots & x_{1_{\tau',\gamma}} \\ x_{1_{\tau'+1,1}} & x_{1_{\tau'+1,2}} & \cdots & x_{1_{\tau'+1,\gamma}} \\ \vdots & \vdots & \ldots & \vdots \\ x_{1_{\tau+\varphi,1}} & x_{1_{\tau+\varphi,2}} & \cdots & x_{1_{\tau+\varphi,\gamma}} \end{bmatrix} \end{array} \tag{11}$$

Table 5 presents X_1 corresponding to the robotic arm example. Note that two *Components* (C_6 and C_7) have been eliminated for their immaturity. Likewise, two *Capabilities* (P_5, and P_7) have been eliminated based upon the *MRED Operator's assessment of the Metrics*

Table 6
Y_1 Relationship matrix for robotic arm.

	Tools		
Technical performance metrics	Tension sensor	Dynamometer	LADAR
Max force	1	1	0
Max linear velocity	0	0	1
Max torque	0	1	0
Range of motion	0	0	1
Max lift capacity	1	0	0
Speed	0	0	1
Force	1	1	0

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

that could not be captured (with respect to these *Capabilities*) given the immaturity of these two *Capabilities*.

Once the remaining steps are completed in box 2, it is time to input and refine the available *Tools*.

4.4.2. Tools

A process now occurs for the *Tools* (shown in box 3 in Fig. 4) similar to what was just performed 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 Eq. (11).

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

Table 6 presents the Y_1 for the robotic arm example. Once the steps are complete in box 3 of Fig. 4, it is time to input the available *Personnel*. This leads to further eliminating and filtering of the remaining candidate blueprint elements.

4.4.3. 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 Fig. 4. Input *personnel* are captured in the matrix N defined in Eq. (12).

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

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*
- $\text{Row}_4(N)$ corresponds to *team members*
- $\text{Row}_5(N)$ corresponds to *participants*
- $\text{Col}_1(N)$ corresponds to presence of personnel (0 – Unavailable, 1 – Available)
- $\text{Col}_2(N)$ corresponds to *technical knowledge* (0 – None, 1 – Low, 2 – Medium, 3 – High)
- $\text{Col}_3(N)$ corresponds to *operational knowledge* (0 – None, 1 – Low, 2 – Medium, 3 – High)

Table 7
Personnel Restrictions by *TTLs* and *Metrics*.

	Applicable goal types for participation				
	Technical performance			Utility assessment	
	Component	Capability	System	Capability	System
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

MRED Operator inputs the above information via user-interface such as the Matlab example shown in Fig. 5.

MRED eliminates the *Personnel, TTLs*, and *metrics* (4.B. in Fig. 4). Table 7 presents the constraints specifying which *tech users* can evaluate the various *TTLs* in support of the two types of *metrics* (Weiss & Schmidt, 2011b, 2012).

Elimination of *TTLs* and *Metrics* at the next step not only satisfies the *personnel* constraints, 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 Fig. 4.

4.5. Stakeholder preferences

The next phase of MRED is to capture *stakeholder preferences*. MRED captures preferences from the pertinent *stakeholders* on an

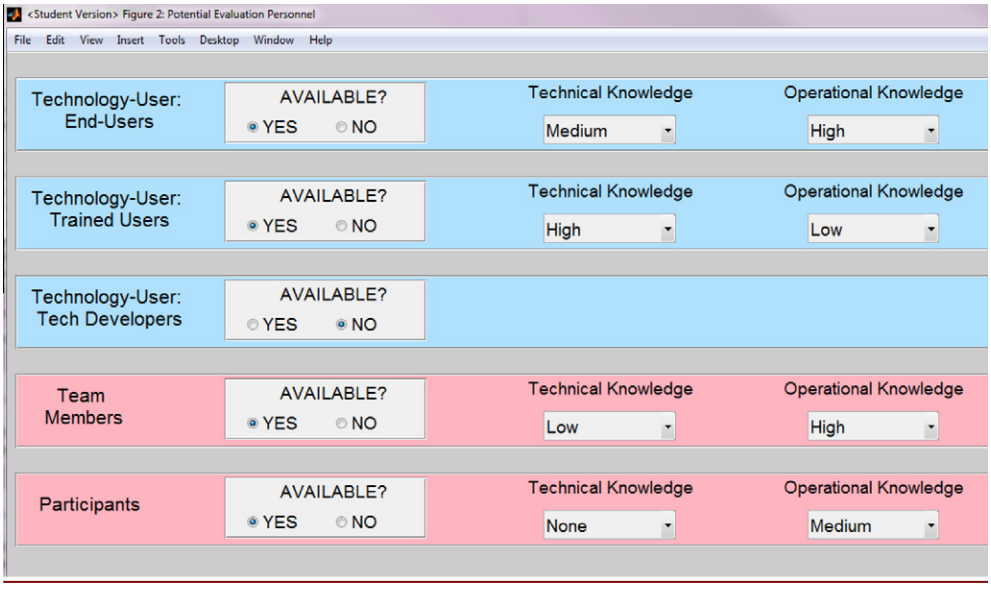

Fig. 5. Matlab user-interface depicting the available *personnel* and their greatest knowledge levels.

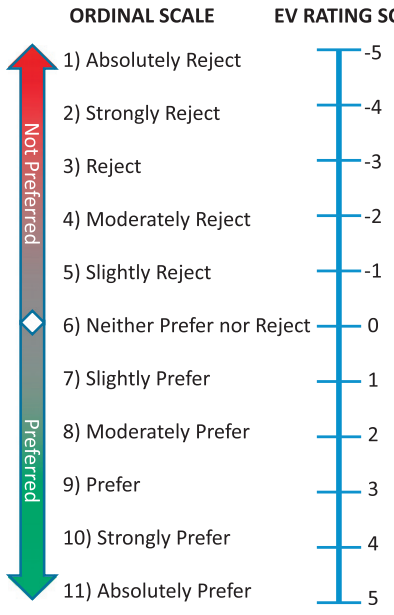

Fig. 6. Scales used for stakeholder preference capture and handling.

11-point, ordinal, linguistic scale to signify their preference for, neutrality toward, or preference against a specific evaluation blueprint element (when solicited for pairing with other blueprint elements) (Weiss and Schmidt, 2013). This scale ranges from absolutely prefer to absolutely reject and is presented in Fig. 6. These linguistic preferences are then mapped to a corresponding 11-point evaluative voting scale. MRED applies Hillinger's evaluative voting method to handle *stakeholder preferences* (Hillinger, 2004). Evaluative voting takes the numerical scores of each *stakeholder* across an alternative and produces an average score. The average for each alternative is calculated.

For example, applying the *EV-11* scale to MRED would be asking the *stakeholders* to score each of the available *TTL-metric pairs* with respect to their preference for evaluating a specific *TTL-metric* pair. If a *stakeholder* chooses not to vote on a specific element (due to a lack of information), the vote remains at the default of 'NV' to indicate they are recusing themselves from scoring that specific element. This is different from the Hillinger's EV method. MRED only averages in a score of '0' if a *stakeholder* actively scores a specific element as neutral. The rationale behind this decision is that neutral preferences have a mathematical impact on the overall scores, where their lack of inclusion can present misleading data. In addition, the standard deviation is calculated for all preferences across each alternative. The meaning of these values are that they present the level of *stakeholder* agreement for a given alternative; the smaller the standard deviation, the more the *stakeholders* agree upon the particular preference score for an alternative.

There are numerous benefits to integrating evaluating voting with MRED to capture *stakeholder preferences* (Dummett, 1998; Dym, Wood, & Scott, 2002; Hillinger, 2004; Sukhatme & Bekey, 1995). They are:

- Rating process (as opposed to ranking process)
- Enables *stakeholders* to abstain from voting
- Minimizes the burden placed on the *stakeholders* by minimizing the quantity of information to be collected
- Accounts for preferences of multiple *stakeholders*
- Captures ordinal preferences and produces interval measurements
- Capture preferences of alternatives such that comparisons can be made of preferences of the same alternative from one evaluation to the next
- Expresses strength of preference
- Standard deviation presents level of agreement among Stakeholders

MRED employs evaluative voting in a three-step, iterative process, with one exception (noted below). The three primary steps are QUERY, SCORE, ELIMINATE, with the fourth being GROUP. In terms of the Evaluative Voting approach, these steps are defined as:

- QUERY – MRED queries *Stakeholder Preferences* $(-5,-4,\ldots,-1, 0,1,\ldots,4,5)$ for each blueprint element (e.g. *TTL-metric* pair). These are captured in matrices for further use.
- SCORE – MRED applies evaluative voting strategy to score each blueprint element where scores range from -5 to 5.
- 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. >0).
- 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 (for all other blueprint elements) – MRED assigns the highest scoring blueprint element to the corresponding group of *TTL-metric* pairs and removes all other candidates from consideration for evaluation with this specific grouping.

MRED begins evaluative voting in the upper left box, I., by determining the preferred *TTL-metric* pairs. Table 8 presents the first step of querying the Stakeholders for their specific preferences according to the 11-point evaluative voting scale.

Table 8 shows the *stakeholder preferences* while Table 9 presents the scores of these preferences. In this specific case, the *MRED operator* defined the threshold for test consideration to be at 0. This means that any *TTL-metric* pairs scoring at or below 0 would be eliminated from further consideration. 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 by *TTL* (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,' and 'P_3 – Max linear velocity'), by *Metric* (e.g. all of the *TTLs* required to produce the 'range of motion' *metric* are grouped together) or a combination of the two at the *MRED operator's* discretion 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' is an important *metric* for both *components* and *capabilities*.

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. Based upon the data shown in Table 9, it is reasonable that the *MRED operator* could choose to test 'C_2 – range of motion' considering that it did not score much below 0 and range of motion *metrics* are already being captured for three other *TTLs*.

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 preserves 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 5).

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Table 8
stakeholder preferences of *TTL-Metric* Pairs.

TTL-Metric pairs	Stakeholder preferences				
	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 9
Evaluative voting scores of *stakeholder preferences* for *TTL-metric* pairs.

Evaluative voting		
TTL-metric pairs	Average	Std dev
C_4 – Range of motion	5.00	0.00
C_5 – Range of motion	5.00	0.00
P_3 – Range of motion	4.80	0.45
C_3 – Range of motion	4.67	0.58
P_3 – Max force	4.60	0.55
P_1 – Range of motion	4.40	0.89
P_2 – Range of motion	4.40	0.89
P_3 – Max linear velocity	4.00	0.71
C_3 – Max linear velocity	3.67	0.58
C_5 – Max force	3.67	1.53
C_5 – Max linear velocity	3.67	1.15
C_3 – Max force	3.33	1.15
C_4 – Max torque	3.33	1.53
C_4 – Max angular velocity	3.00	2.65
P_1 – Smoothness	2.80	2.95
P_2 – Responsiveness	2.80	2.95
P_2 – Smoothness	2.80	2.95
P_3 – Smoothness	2.80	2.95
P_1 – Max linear velocity	2.60	1.14
P_1 – Responsiveness	2.60	1.14
P_2 – Max linear velocity	2.60	1.14
P_3 – Responsivenss	2.60	1.14
P_1 – Max force	2.40	1.52
P_2 – Max force	2.20	1.30
P_1 – Force	1.80	2.39
P_2 – Force	1.80	2.39
P_3 – Force	1.80	2.39
C_1-Range of motion	0.67	1.53
C_2-Range of motion	−0.33	3.21
C_1-Max torque	−0.67	1.53
C_1-Max angular velocity	−1.00	2.65
C_2-Max torque	−1.00	1.73
C_2-Max angular velocity	−1.00	2.65

Table 10
Groupings of TTL-metric pairs.

METRIC GROUPINGS	GROUP	TTLs	Pair Average		GROUP	TTLs	Pair Averages
	Range of Motion	C_3	4.67		Max Force	P_3	3.80
		P_3	4.60			C_3	3.33
		P_2	4.00			P_2	2.20
		C_1	0.67		Max Linear Vel.	P_3	3.80
		C_2	-0.33			P_2	2.00
					Max Angular Vel.	C_3	3.67

Table 10 presents example groupings of *TTL-metric* pairs based upon the *stakeholder preferences* and scores generated from evaluative voting.

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 repeating a similar QUERY -> SCORE -> ELIMINATE process.

Obtaining *stakeholder preferences* during this test plan design process is just as crucial as getting feedback from *stakeholders* during product design. *Evaluation designers* are often not as informed as *end-users* on the practical applications, as knowledgeable of technological limits as *technology developers* so it's important to attain wide-ranging perspectives, or in tune with higher-level programmatic goals from the *sponsor's* perspective.

Now each *stakeholder* is asked to provide their *personnel* preferences for each grouping of *TTL-metric* pairs. Table 11 provides the *stakeholder preferences* and evaluative voting scores for *personnel* for the 'range of motion' grouping. Similarly, *stakeholder preferences* would be captured and scored for the other *metric* groupings. since this table relates to the presence of specific *personnel*, Ta-

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Table 11
Stakeholder preferences for *Personnel* for 'range of motion' *metric* grouping.

Stakeholder preferences	Capability – Range of motion, smoothness, responsiveness grouping					Evaluative voting	
Personnel	Buyer	Eval	Sponsor	Tech dev	User	**Average**	**STD DEV**
Tech user: End-user	Strongly Pref	Slightly Pref	Strongly Pref	Mod Pref	Abs prefer	3.20	1.64
Tech user: Trained User	Strongly Rej	Slightly Rej	Strongly Rej	Reject	Abs Reject	−3.60	1.67
Team member	Neither	Strongly Rej	Slightly Pref	Strongly Rej	Prefer	−0.80	3.11
Participant	Mod Pref	Mod Reject	Abs Rej	Strongly Rej	Mod Pref	−1.40	3.29

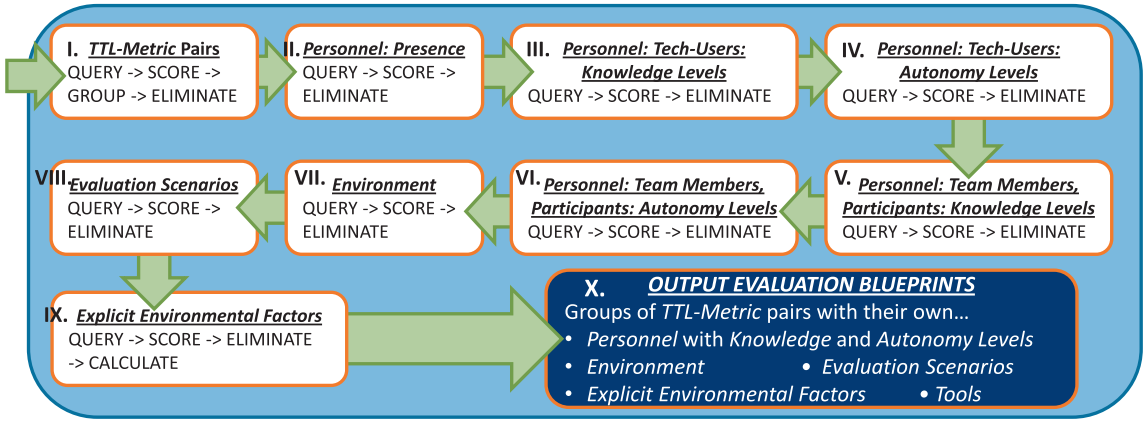

Fig. 7. Evaluative voting approach to capturing *stakeholder preferences* in MRED.

ble 11 states that the *stakeholders* prefer that the *end-users* be the *technology users* during tests to capture 'range of motion' data and that *trained users* and *technology developers* are less desirable (given both their lower score and their average scores being less than 0). Additionally, the *stakeholders* prefer that both *team members* and *participants* be involved in the 'range of motion' evaluations since their respective scores are above 0. In the case of the robotic arm example, the *MRED operator* may reasonably define *team members* to be other robot operators on the floor. Likewise, *participants* could be other employees that work near or around the robot yet would not have any direct interaction with it.

The QUERY -> SCORE -> ELIMINATE process continues with all of the remaining blueprint elements, as shown in Fig. 7, for each grouping. At the conclusion of this process, each grouping contains a complete set of evaluation plans that specify the *TTLs* to be evaluated, the *metrics* to be captured, the necessary evaluation *personnel* and their corresponding *knowledge* and *autonomy levels*, the *environment (s)* in which to evaluate the technology, the *evaluation scenarios* to drive the tests, the *explicit environmental factors*, and the required *tools*.

5. Conclusion and future work

The MRED process is formalized and demonstrates its potential as an automatic test plan generator. Among its contributions, this paper highlights the overall process including the objective removal of test plan elements given various constraints and relationships. The paper also presents an application of an iterative process of evaluative voting that is intertwined with the capture of *stakeholder preferences*.

An item of future work is to compare the impact of implementing preference capture and use strategies to confirm that evaluative voting is adequate for this application. Calculating cost of individual sets of evaluation blueprints is another area of exploration. Weighting of individual *stakeholder preferences* may be another valuable contribution to this effort since some *Stakeholders* may have greater importance than others in the program and/or others may have specific expertise regarding specific evaluation blueprint elements. Finally, Table 9 shows several situations that still require *MRED Operator* discretion, yet MRED provides clarity. Specifically, a *TTL-Metric* pair scored just below the '0' threshold for evaluation consideration while another pair scored just above this same threshold. MRED presents their standing within all of the scores, yet it's the *MRED Operator* who must ultimately decide if MRED holds firm to this threshold or not. Future efforts could expand MRED to automatically address this issue.

Acknowledgements

The authors would like to acknowledge NIST's Intelligent Systems Division for its continued support.

References

Cohen, P., & Howe, A. (2008). How evaluation guides AI research. *AI Magazine, 9*(4).
Dummett, M. (1998). The Borda count and agenda manipulation. *Social Choice and Welfare, 15*(2), 289–296.
Dym, C., Wood, W., & Scott, M. (2002). Rank ordering engineering designs: Pairwise comparison charts and Borda counts. *Research in Engineering Design, 13*, 236–242.
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.*
Hillinger, C. (2004). Voting and the cardinal aggregation of judgments. Munich Discussion Paper 2004-9, <http://epub.ub.uni-muenchen.de/353/> Accessed 11/24/2012.
Sukhatme, G. S. & Bekey, G. A. (1995). An evaluation methodology for autonomous mobile robots for planetary exploration. In *Proceedings of the first ecpd international conference on advanced robotics and intelligent automation* (pp. 558–563).
Weiss, B. A. (2012). Multi-relationship evaluation design (MRED): An interactive test plan designer for advanced and emerging technologies (Doctoral Dissertation). Available from digital repository at the University of Maryland (DRUM).
Weiss, B. A. & Schmidt, L. C. (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*. Baltimore, MD, USA, September 28–30.
Weiss, B. A. & Schmidt, L. C. (2011b). Multi-relationship evaluation design: formalizing test plan input and output elements for evaluating developing intelligent systems DETC2011-47971. In *Proceedings IDETC/CIE ASME 2011 international design engineering technical conferences & computers and information in engineering conference*. Washington, DC, USA. August 29–31.

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083

Weiss, B. A. & Schmidt, L. C. (2012). Multi-relationship evaluation design: modeling an automatic test plan generator. In *Proceedings of the 2012 performance metrics for intelligent systems (PerMIS) workshop*. College Park, MD. March 20–22.

Weiss, B. A., Schmidt, L. C., Scott, H., & Schlenoff, C. I. (2010). The Multi-Relationship evaluation design framework: Designing testing plans to comprehensively assess advanced and intelligent technologies DETC2010-28928. In *Proceedings IDETC/CIE 2010 ASME design engineering technical conferences and computers and information in engineering conference* Montreal, Quebec, Canada. August 15–18.

Weiss, B. A., & Schmidt, L. C. (2011a). The multi-relationship evaluation design framework: Producing evaluation blueprints to test emerging, advanced, and intelligent systems. *ITEA Journal, 32*(2), 191–200.

Weiss, B. A., & Schmidt, L. C. (2011c). Multi-relationship evaluation design: formalizing test plan input and output blueprint elements for testing developing intelligent systems. *ITEA Journal, 32*(4), 479–488.

Please cite this article in press as: Weiss, B. A., & Schmidt, L. C. Multi-relationship evaluation design: Formalization of an automatic test plan generator. *Expert Systems with Applications* (2013), http://dx.doi.org/10.1016/j.eswa.2012.12.083