FORMAL SPECIFICATION OF TESTABILITY METRICS IN IEEE P1522

John W. Sheppard, ARINC, (410) 266-2099, jsheppar@arinc.com
Mark Kaufman, NSWC Corona Div., (909) 273-5725, kaufmanma@corona.navy.mil

ABSTRACT

The objective of the IEEE P1522 Testability and Diagnosability Metrics standard is to provide notionally correct and mathematically precise definitions of testability measures that may be used to either measure the testability characteristics of a system, or predict the testability of a system. Notionally correct means that the measures are not in conflict with intuitive and historical representations. Predictive testability analysis may be used in an iterative fashion to improve the factory, field-testing, and maintainability of complex systems. The end purpose is to provide an unambiguous source for definitions of common and uncommon testability and diagnosability terms such that each individual encountering it can know precisely what that term means.

Keywords: testability, diagnosability, information models, metrics, P1232, P1522, standard

1 INTRODUCTION

Test costs impact overall costs for design, production and maintenance. Testability impacts test costs. Although system complexity does impact testability, system design has a much larger impact. Testability provides a means of making design decisions based on the impact on test costs.

As weapons systems became more complex, costly, and difficult to diagnose and repair, DoD initiatives were started to address these problems. The objective of one of these initiatives, testability, was to make systems easier to test. Early on, this focused on having enough test points in the right places. As systems evolved, it was recognized that the system design had to include characteristics to make the system easier to test. This was the start of considering testability as a design characteristic. As defined in MIL-STD-2165, testability is "a *design characteristic* which allows the status (operable, inoperable, or degraded) of an item to be determined and the isolation of faults within the item to be performed in a timely manner." [1]. The purpose of MIL-STD-2165 was to provide uniform procedures and methods to control planning, implementation, and verification of testability during the system acquisition process by the Department of Defense (DoD). It was to be applied during all phases of system development—from concept to production to fielding. This standard, though deficient in some areas, provided useful guidance to government suppliers. Further, lacking any equivalent industry standard, many commercial system developers have used it to guide their activities

0-7803-7094-5/01/$10.00 ©2001 IEEE

although it was not imposed as a requirement. Acquisition reform ended any further development of a testability standard by the DoD.

Lord Kalvin over a century ago commented: "When you can measure what you are speaking about and express it in numbers, you know something about it; and when you cannot measure it, you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind. It may be the beginning of knowledge, but you are scarcely in your thought advanced to the stage of a science." To add to Lord Kalvin's insight, when different tools measure something differently, and the basis for the measurement is not known or understood; it is not knowledge; it is not science; it is marketing.

There are a number of testability metrics in use that appear to be the same thing, but are expressed differently There are multiple definitions for some metrics. It is not possible to compare the metrics generated by one tool with metrics generated by another without a great deal of work. Some means of standardizing the basics of testability measurements is needed. The intent is not to restrict the number or type of metrics, but to provide a sound, understandable, and repeatable basis for those measurements.

2 P1522 OBJECTIVES

The IEEE P1522 draft Standard for Testability and Diagnosability Characteristics and Metrics [5] is being developed by the Diagnostic and Maintenance Control (D&MC) Subcommittee of the IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems. The purpose of the standard is to provide formal, unambiguous definitions of testability and diagnosability metrics and characteristics. P1522 builds on the fundamental definitions in standard information models related to test and diagnosis, drawing primarily from IEEE Std 1232—Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE) [4].

The goals of the P1522 standard are:

- Provide definitions of testability and diagnosability characteristics and metrics that are independent of specific test and diagnosis technologies.
- Provide definitions of testability and diagnosability characteristics and metrics that are independent of specific system under test technologies.
- Provide unambiguous definitions of testability and diagnosability metrics to support procurement and support organizations.
- Provide selected, qualitative definitions of testability and diagnosability characteristics to assist procurement and support organizations in evaluating system testability and diagnosability.

P1522 is based on the mathematical definitions of testability and diagnosability metrics in existing standard information models. Where entities are required that have not been defined in any existing standard information model, P1522 will provide its own information model to satisfy the deficiency. It is not the intent of the standard to impose any implementation-specific requirements in terms of actually computing the metrics; however, metrics not computed using the identified standard information models must be demonstrated to be mathematically equivalent to the definitions provided.

3 TESTABILITY AND DIAGNOSABILITY METRICS

Testability has been broadly recognized as the "-ility" that deals with those aspects of a system that allow the status (operable, inoperable, or degraded) or health state to be determined. Early work in the field primarily dealt with the design aspects such as controllability and observability. Almost from the start this was applied to the manufacturing of systems where test was seen as a means to improve production yields. The scope of testability has been expanded to include the aspects of field maintainability such as false alarms, isolation percentages, and other factors associated with maintaining a system.

In the industry, many terms such as test coverage and Fraction of Faults Detected (FFD) are not precisely defined or have multiple definitions. Further, each diagnostic tool calculates these terms differently and therefore the results are not directly comparable. Some measures, such as false alarm rate, are not measurable in field applications. Other measures such as Incremental Fault Resolution, Operational Isolation, and Fault Isolation Resolution appear different, but mean nearly the same thing.

Lacking well-defined testability measures, the tasks of establishing testability requirements, and predicting and evaluating the testability of the design are extremely difficult. This in turn makes effective participation in the design-for-testability process difficult. These difficulties will be greatly diminished by standard testability metrics. An immediate benefit will come with a consistent, precise, and measurable set of testability attributes that can be compared across systems, tools, and within iterations of a system's design.

MIL-STD-2165 did not have precise and unambiguous definitions of measurable testability figures-of-merit and relied mostly on a weighting scheme for testability assessment. (It should be noted, however, that the standard did permit the use of analytical tools for testability assessment such as SCOAP, STAMP, and WSTA).

As concurrent engineering practices are established, the interchange between the testability function and other functions becomes even more important. Integrated diagnostic environments are where the elements of automatic testing, manual testing, training, maintenance aids, and technical information work in concert with the testability element. To create these environments we must maximize the reuse of data, information, knowledge, and software. Complete diagnostic systems include Built-In-Test (BIT), Automatic Test Equipment (ATE), and manual troubleshooting. It would be desirable to be able to predict and evaluate the testability of systems at these levels.

It is not a coincidence that the P1522 standard contains both the word testability and the word diagnosability. The distinction is not always easy to maintain, especially in light of the expansion of the use of the testability term. Figure 1 shows the basic relationship, with diagnosability being

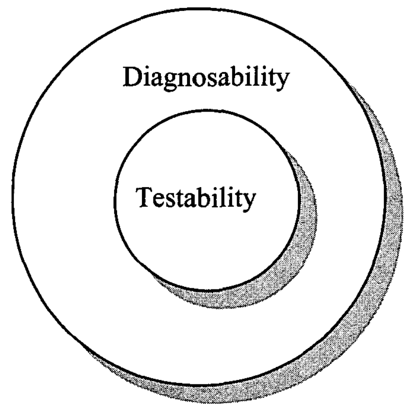

Figure 1. Relationship Between Diagnosability and Testability

the larger term and encompassing all aspects of detection, fault localization, and fault identification. The boundary is fuzzy and often it is not clear when one term applies and the other does not. The P1522 standard encompasses both aspects of the test problem. Because of the long history of the use of the testability term, we will seldom draw a distinction. However, the use of both terms is significant in that testability is not independent of the diagnostic process. The writing of test procedures cannot and should not be done separately from testability analyses. To do so, would be meeting the letter of the requirements without considering the intent.

3.1 Metric Issues

Metrics are a measure of some identifiable quantity. The metrics of P1522 are derived from data obtained from the information models in IEEE Std 1232. At the foundation of all the metrics is a set of counts of elements that support test and diagnosis. These counts are derived from constructs in the information models. For example, the number of faults, components, and functions are obtainable from the 1232 models. A testability analysis tool could ask for the total number of faults and then ask for the number of faults detected. The tool could then calculate the fraction of faults detected. This example is extremely simplified.

MIL-STD-2165 defined Fraction of Faults Detected (FFD) two ways. The first is the fraction of *all* faults detected by BIT/External Test Equipment (ETE). The second is the fraction of *all detectable* faults detected by BIT/ETE [1]. False alarms were excluded from the definition. From these two variations grew many others.

One problem with traditional metrics is that they are "overloaded." Overloaded in this case means that due to "common understanding" of the terms, fine variations are not specified. Consequently, users of the term do not necessarily know the implications of a precise definition. Discussions of overloaded terms go on at length, in part because everyone in the discussion has brought along a lot of mental baggage. Often, progress is only made when a neutral term is chosen and the meaning is built from the ground up. This overloading is so severe, for example, that there was no definition of FFD in *System Test and Diagnosis* [9], the authors preferring to use Non-Detection Percentage (NDP). FFD is the negative of NDP and is equal to 1–NDP.

Even the number of faults counted in the field requires a more precise definition. The "overloaded" version is simply a count of all the things that failed. The quantity of all faults, as usually defined in the industry, is different. The quantity of faults detected by BIT/ETE, and the quantity of faults detected exclude the occurrence of false alarms. Intermittent faults are classified as a single fault. Temporary faults, those caused by external transients of noise, are not classified as faults.

3.2 Classes of Metrics

In developing P1522, the DMC drew upon several documents and standards related to testability and evaluated their appropriateness from a standards perspective. Specifically, some of the sources considered included MIL STD 2165A [1] and INT DEF STAN 00-13/3 [4]. In addition, sources such as the Navy's Testability Handbook [12], documents from Rome Laboratories [9], and textbooks such as *System Test and Diagnosis* [11] were examined.

After reviewing these sources, several classes of metrics were identified for specification in the standard. The principal approach chosen involves defining several "fundamental" measures based on entities within standard diagnostics domain information models (see below). The fundamental measures include counts of entities within the model, costs of items within the test and diagnosis context, and basic detection and isolation metrics. From these measures and metrics, several "higher order" metrics are being defined. These metrics have been categorized as fault detection metrics, fault isolation metrics, and fault resolution metrics.

4 INFORMATION MODELS

ISO 10303–11 (EXPRESS) is used to define information models and exchange formats for diagnostic knowledge [7]. These international standards are being maintained by the STEP (Standard for the Exchange of Product model data) community. The current approach to static information exchange within AI-ESTATE is to derive the exchange format from the formal information models as specified in the ISO standards.

The purpose of information modeling is to provide a formal specification of the *semantics* of information that is being used in an "information system." Specifically, information models identify the key entities of information to be used, their relationships to one another, and the "behavior" of these entities in terms of constraints on valid values [7]. The intent is to ensure that definitions of these entities are unambiguous.

The AI-ESTATE standards [1][2][2][4] are information exchange standards for test and diagnosis. The original standards, the 1232 series, developed a means of exchanging information between diagnostic reasoners. As the information models for the 1232 standards were developed, it became apparent that these models could be used for standardizing testability and diagnosability metrics.

In 1998, the third of a series of three standards was published by the IEEE addressing issues in system-level diagnostics. IEEE Std 1232-1995 [1] defines the architecture of an AI-ESTATE-conformant system and has been published as a "full-use" standard; however, this standard was published before the vision of AI-ESTATE was fully developed. IEEE Std 1232.1-1997 [2] defines a knowledge and data exchange standard and is now a "full-use" standard. In 1998, IEEE Std 1232.2-1998 [3] was approved. It is now published as a "trial-use" standard. Trial-use indicates that it is preliminary in nature, and the standards committee is seeking comments from organizations attempting to implement or use the standard.

Recently, the 1232 standard documents were merged into a single standard that encompasses the 1232-1995, 1232.1-1997 and the 1232.2-1998 documents. The complete 1232 standard, contains the diagnostic information models and formally defines a set of standard software services to be provided by a diagnostic reasoner in an open-architecture test environment. The 1232 standards were developed using information modeling, resulting in the definition of five models addressing static and dynamic aspects of the diagnostic domain. The information models are; the AI-ESTATE Common Element Model (CEM), the AI-ESTATE Fault Tree Model (FTM), the AI-ESTATE Diagnostic Inference Model (DIM), the AI-ESTATE Enhanced Diagnostic Inference Model (EDIM), and the AI-ESTATE Dynamic Context Model (DCM).

In 1997, the AI-ESTATE committee began to work on a new standard focusing on expanding the work of the cancelled MIL-STD 2165. The military standard focused on specifying the essential elements of a testability program and explained the elements needed to define a testability program plan. In addition, MIL-STD 2165 included the "definition" of several testability metrics. MIL –STD 2165 included a testability checklist to be used to determine overall system testability. With the cancellation of military standards and specifications by the Perry Memo in 1994 [9], and with the lack of specificity and clarity in MIL-STD 2165, it became evident that a replacement was necessary. The approach taken to develop this standard involved defining testability and diagnosability metrics based on standard information models. Specifically, it was found that the AI-ESTATE models provided an excellent foundation for defining these metrics. AI-ESTATE provides formal definitions of the information required for test and diagnosis—the same information required for determining the testability and diagnosability of a system. With these formal definitions, the constraint language of EXPRESS can be applied directly to define metrics and characteristics of testability and diagnosability.

As stated earlier, the testability and diagnosability metrics are defined and computed using entities from the 1232 information models. As stated elsewhere, the definition and method of computation of these metrics will also remain valid using entities from a user defined model as long as the entities are mathematically equivalent to those of the 1232 information models. The 1522 standard defined two distinct sets of measures–predictive and historical.

The predictive measures define the testability and diagnosability metrics derived from the knowledge specification information models in 1232. These measures can be generated after a 1232 compliant diagnostic model has been developed which requires the diagnostic model to be defined in terms of the CEM and one of the following: FTM, DIM or EDIM. The predictive measures thus provide a quantitative evaluation of how well the system can be tested and diagnosed using the model developed. Depending upon whether the model developed is in the form of a fault tree (FTM) or a test-diagnosis inference relationship (DIM or EDIM), the 1522 predictive metrics will be computed differently and can result in differing quantitative values.

The 1522 standard also anticipates that apart from the predictive measures, an additional set of historical measures that can be computed using relevant data gathered from the field. This data would consist of maintenance history data and other test data from the field to be utilized to validate the predictive measures generated from entities defined in the diagnostic model. Maintenance history and other field data would be a direct outcome of the final diagnostic state of the system as defined in the DCM and hence these measures will be based on entities defined in the DCM and can be used to validate the predictive measures for that system. These measures generated with enough data from the field will eventually allow the testing community to evaluate the effectiveness of the diagnostic model developed.

5 P1522 METRICS

The approach being taken to define formal, unambiguous testability and diagnosability metrics is based on the assumption that the 1232 information models define formal semantics for information of interest to the testability and diagnosability domain. In this section, several specific, fundamental measures will be defined using EXPRESS notation [7] and are based on indicated information models.

In defining the fundamental measures, the following assumptions were made.

- The primary information models used to define these measures are drawn from IEEE Std P1232—Standard for AI-ESTATE. The specific models used include the Common Element Model, the Fault Tree Model, the Diagnostic Inference Model, the Enhanced Diagnostic Inference Model, and the Dynamic Context Model.
- All cost-related metrics shall be computed with respect to a single test or repair within the Common Element Model as qualified by specifically identified constraints and types.
- To manage the complexity of computing isolation-related metrics and to remove dependence on diagnostic and maintenance strategies, all isolation metrics shall be computed under the assumption that only a single, independent fault exists at any given time.
- When defining the functions for detection and isolation, it is clear that the type of underlying diagnostic model (i.e., fault tree model, diagnostic inference model, or enhanced diagnostic inference model) will establish detectability and isolateability differently. Consequently, these functions will be defined with an argument specifying which of the model types provided in IEEE Std 1232 is being used.

Within IEEE Std 1232 (AI-ESTATE), cost is defined within the information model to be categorized by the type of cost to which they relate. One dimension of the cost set identifies whether the cost is a measure of time or if it is a calculated cost. A time-related cost is a measure of the time it takes to perform a task. A non-time-related cost is an expense that is computed, perhaps in financial terms or by an objective function. The second dimension to the cost group is based on the task to which the cost pertains: performance, setup, access, and reentry.

In defining the fundamental measures, the following assumptions are made:

- Cost is computed over a set of tests or repairs. In the simplest case, the set may contain only one test or repair.
- All cost calculations are based on action or resource cost with common units. No unit conversion is included in the definition. No user-defined units are permitted.
- Costs associated with actions or resources are assumed to reflect the cost incurred for that action or resource.
- The cost functions return the cost only for the test or repair at that element's point in the hierarchy—they do *not* roll up cost from the children.
- When using a cost function, a cost type is specified and does not include "USER_DEFINED."
- Four versions of the measures are specified—one associated with a particular level and a particular required context, one associated with a particular level but all contexts (i.e., no required context specified), one associated with a particular required context but all levels (i.e., no level specified), and one with no level and no required context specified.

By way of example, we will consider the definition of "Expected Percentage of Faults Detected." This is considered one of the higher order metrics and will illustrate how its definition is built up from the fundamental metrics. Expected Percentage of Faults Detected (EPFD) is the failure probability-weighted percentage of possible faults at a particular level within a given diagnostic model that can be detected by the set of tests that have been defined within that model. This metric is described by the following equation:

$$EPFD(mdl,lvl) = \frac{100 * \sum_{i=1}^{F_D} \lambda_{Di}}{\sum_{i=1}^{F_T} \lambda_i}$$

where F_D = the number of faults in the set returned by the fundamental measure `detectable_faults_set(mdl, lvl)`,

F_T = the number of faults in the set returned by the fundamental measure `faults_set(mdl, lvl)`,

λ_{Di} = the failure rate of the ith fault in the set returned by the fundamental measure `detectable_faults_set(mdl, lvl)`,

λ_i = the failure rate of the ith fault in the set returned by the fundamental measure `faults_set(mdl, lvl)`,

`mdl` = the diagnostic model for which this metric is to be calculated, and

`lvl` = the level of the diagnostic model at which faults are to be counted.

Within the definition of *EPFD* is the determination of the fault set and the detectable fault set. The fault set is the set of faults defined within a model (`mdl`) at a particular level of test (`lvl`). This set is determined from the following EXPRESS function:

```
USE FROM AI_ESTATE_COMMON_ELEMENT_MODEL
      (diagnostic_model, level, fault);

FUNCTION faults_set(mdl : diagnostic_model; lvl : level) :
            SET [0:?] OF fault;
      LOCAL
            fault_set : SET [0:?] OF fault := [];
      END_LOCAL;

      fault_set := QUERY(tmp <* mdl.model_element |
            ('AI_ESTATE_COMMON_ELEMENT_MODEL.fault' IN TYPEOF(tmp)));
      fault_set := QUERY(tmp <* fault_set |
            (lvl IN tmp.at_indenture_level));
      return(fault_set);
END_FUNCTION;
```

Similarly, the detectable fault set is the set of faults (which are a subset of the total set of faults) that can be detected by a given set of tests. This set of tests is given by `tests_set(mdl, lbl)`,. The function for determining the detectable fault set is defined as follows:

```
USE FROM AI_ESTATE_COMMON_ELEMENT_MODEL
      (diagnostic_model, test, diagnosis, level);
USE FROM AI_ESTATE_FAULT_TREE_MODEL;
USE FROM AI_ESTATE_DIAGNOSTIC_INFERENCE_MODEL;
USE FROM AI_ESTATE_ENHANCED_DIAGNOSTIC_INFERENCE_MODEL;

FUNCTION detectable_faults_set(mdl : diagnostic_model; lvl : level) :
                  SET [0:?] OF diagnosis;
      LOCAL
            results : SET [0:?] OF test_result;
            detectable_set : SET [0:?] OF diagnosis := [];
            diag_inf : SET [0:?] OF inference := [];
      END_LOCAL;

      IF ('AI_ESTATE_FAULT_TREE_MODEL.
                  fault_tree_model' IN TYPEOF(mdl)) THEN
            REPEAT I := LOINDEX(mdl.fault_tree.result) TO
                        HIINDEX(mdl.fault_tree.result);
                  detectable_set := detectable_set +
                        get_leaves(mdl.fault_tree.result[I]);
            END_REPEAT;
      END_IF;
      IF ('AI_ESTATE_DIAGNOSTIC_INFERENCE_MODEL.
                  diagnostic_inference_model' IN TYPEOF(mdl)) THEN
            REPEAT I := LOINDEX(mdl.inference) TO
                        HIINDEX(mdl.inference);
                  diag_inf := QUERY(tmp <* mdl.inference[I].conjuncts |
                        'AI_ESTATE_DIAGNOSTIC_INFERENCE_MODEL.
                        diagnostic_inference' IN TYPEOF(tmp));
                  diag_inf := diag_inf +
                        QUERY(tmp <* mdl.inference[I].disjuncts |
                        'AI_ESTATE_DIAGNOSTIC_INFERENCE_MODEL.
                        diagnostic_inference' IN TYPEOF(tmp));
```

```
                diag_inf := QUERY(tmp <* diag_inf |
                      tmp.diagnostic_assertion.standard_diagnosis_value
                      = CANDIDATE);
          END_REPEAT;
    END_IF;
    IF ('AI_ESTATE_ENHANCED_DIAGNOSTIC_INFERENCE_MODEL.
                enhanced_diagnostic_inference_model' IN TYPEOF(mdl)) THEN
          REPEAT I := LOINDEX(mdl.inference) TO
                      HIINDEX(mdl.inference);
                diag_inf := QUERY(tmp <* mdl.inference[I].conjuncts |
                      'AI_ESTATE_ENHANCED_DIAGNOSTIC_INFERENCE_MODEL.
                      diagnostic_inference' IN TYPEOF(tmp));
                diag_inf := diag_inf +
                      QUERY(tmp <* mdl.inference[I].disjuncts |
                      'AI_ESTATE_ENHANCED_DIAGNOSTIC_INFERENCE_MODEL.
                      diagnostic_inference' IN TYPEOF(tmp));
                diag_inf := QUERY(tmp <* diag_inf |
                      ((tmp.pos_neg = NEGATIVE) AND
                      (tmp.diagnostic_assertion.
                      standard_diagnosis_value = GOOD)) OR
                      ((tmp.pos_neg = POSITIVE) AND
                      (tmp.diagnostic_assertion.
                      standard_diagnosis_value = CANDIDATE)));
          END_REPEAT;
    END_IF;
    detectable_set := QUERY(tmp <* detectable_set |
                            (lvl IN tmp.at_indenture_level) AND
                            ('AI_ESTATE_COMMON_ELEMENT_MODEL.fault'
                            IN TYPEOF(tmp)));
    return(detectable_set);
END_FUNCTION;
```

Given the set of faults or detectable faults, the failure rate for each of the faults can be taken directly from the "`has_rate`" attribute of the fault in the information model.

6 STATUS OF STANDARD

The development of P1522 depends on the completion of IEEE P1232 (AI-ESTATE). P1232 has completed the first circulation for ballot and passed with an 87% approval; however, approximately 400 comments (~350 editorial) have been levied against the document. The DMC is currently in the process of resolving the comments and modifying the standard to correct the deficiencies identified during the ballot.

IEEE P1522 depends on the information models within P1232. As the P1232 models are modified, corrected, and updated, the measures and metrics defined in P1522 will be updated to reflect these changes. Currently, all of the fundamental measures have been defined relative to draft 4.0 of P1232. Draft 5.0 is expected to be the final draft for P1232 and should be completed by the fall of 2001. The fault detection and fault isolation metrics have also been defined. It is expected that fault resolution metrics will be completed by fall 2001. It is the hope of the DMC that the P1522 standard will be balloted as a "trial use" standard by the end of 2001.

7 CONCLUSION

The primary difficulty with most current approaches to testability and diagnosability assessment is that they lack a formal foundation. The result of this is that associated testability and diagnosability metrics lack formal, unambiguous understanding within the test community. The purpose of IEEE P1522 is to satisfy this deficiency and provide a common basis for discussing and comparing testability/diagnosability characteristics of a given system.

While the DMC expects to ballot P1522 by the end of 2001, much work remains to be done. First, this version of the standard will be "trial use," meaning that it will only be valid for two years and will be published primarily for the purpose of obtaining comments and feedback from industry. The DMC will then spend the next several years responding to comments, enhancing the models and definitions, and bringing the specification to "full use" status. If you have an interest in the definition of this important industry standard, you are invited to contact either of the subcommittee co-chairs (i.e., the authors). For further information, you may also visit the DMC web site at http://grouper.ieee.org/groups/1232.

8 REFERENCES

[1] IEEE Std 1232-1995. *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Overview and Architecture*, Piscataway, NJ: IEEE Standards Press.

[2] IEEE Std 1232.1-1997. *IEEE Trial-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Data and Knowledge Specification*, Piscataway, NJ: IEEE Standards Press.

[3] IEEE Std 1232.2-1998. *IEEE Trial-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Service Specification*, Piscataway, NJ: IEEE Standards Press.

[4] IEEE P1232/D4.0. 2000. *Draft Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)*, Piscataway, NJ: IEEE Standards Press.

[5] IEEE P1522/D1.0. 2001 *Draft Standard Testability and Diagnosability Characteristics and Metrics*, Piscataway, NJ: IEEE Standards Press.

[6] INT DEF STAN 00-13/3, *Requirements for the Achievement of Testability in Electronic and Allied Equipment*

[7] ISO 10303-11:1994. *Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 11: Description Methods: The EXPRESS Language Reference Manual*, Geneva, Switzerland: International Organization for Standardization.

[8] MIL STD 2165. 1985. *Testability Program for Electronic Systems and Equipment*, Washington, DC: Naval Electronic Systems Command (ELEX-8111)

[9] Perry, William. 1994. "Specifications and Standards—A New Way of Doing Business," US Department of Defense Policy Memorandum.

[10] Simpson, W., Bailey, J., Barto, K. and Esker, E., 1985 "Organization-Level Testability Prediction", ARINC Research Corporation Report 1511-01-3623 Prepared for the Rome Air Development Center.

[11] Simpson, W. and Sheppard, J. 1994. *System Test and Diagnosis*, Boston, MA: Kluwer Academic Publishers.

[12] *Testability Analysis Handbook*. 1992. Naval Undersea Warfare Center.