P1522: A FORMAL STANDARD FOR TESTABILITY AND DIAGNOSABILITY MEASURES

Mark Kaufman
NWAS
PO Box 5000
Corona, CA 91718
909-273-5725
kaufman.mark@corona.navy.mil

John Sheppard
ARINC
2551 Riva Road
Annapolis, MD 21401
410-266-2099
jsheppar@arinc.com

Abstract - Members of the Maintenance and Diagnostic Control subcommittee of IEEE's Standards Coordinating Committee 20 (SCC20) are developing a standard for testability and diagnosability characteristics and metrics. The objective of this standard, P1522 is to provide notionally correct, useful, and mathematically precise definitions of testability measures that may be used to either measure or predict the testability of a system. Notionally correct means that the measures are not in conflict with intuitive and historical representations. The end purpose is to provide an unambiguous source for definitions of testability and diagnosability metrics. In this paper, we present a summary of the work completed so far on P1522 and a roadmap for its completion. We cover the organization of the standard, the sources of the measures, how these measures relate to the other AI-ESTATE standards, and information modeling.

INTRODUCTION

As systems became more complex, costly, and difficult to diagnose and repair, 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 although it was not imposed as a requirement.

MIL-STD-2165 and most other MIL-STDs were cancelled by the Perry Memo in 1994 [2]. At that time, MIL-STD-2165 was transitioned into a handbook and became MIL-HDBK-2165. With the DoD's current emphasis on the use of industry standards, the continuing need to control the achievable testability of delivered systems in the DoD and commercial sectors, and the lack of a replacement for MIL-STD-2165 (commercial or DoD), there is a need for a new industry standard that addresses system testability issues and that can be used by both commercial and government sectors. To be useful, this commercial standard must provide specific, unambiguous definitions of criteria for assessing system testability.

Recent initiatives by the Institute of Electrical and Electronics Engineers (IEEE) on standardizing test architectures have provided an opportunity to standardize testability metrics. The IEEE 1232 Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE) series of standards provide the foundation for precise and unambiguous testability and diagnosability metrics.

The purpose of the AI-ESTATE series of standards is to standardize on the interfaces for diagnostic elements of an intelligent test environment and on

0-7803-5432-X/99/$10.00 © 1999 IEEE

representations of knowledge and data used in intelligent diagnostics. Generally, AI-ESTATE will support the development of applications using artificial intelligence (AI) techniques in the field of system test and diagnosis, and will facilitate intercommunications, interoperability, and reuse of both knowledge and reasoners in a variety of test and diagnostic applications.

This paper describes how testability metrics are being developed based on the information models in the AI-ESTATE 1232 standards. The standard under development is IEEE P1522 Trial Use Standard for Testability and Diagnosability Characteristics and Metrics.

BACKGROUND

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 device to improve production yields. This has been slowly expanded to include the aspects of field maintainability such as false alarms, isolation percentages, and other factors associated with the burden of maintaining a system.

In the industry, many terms such as test coverage and Fraction of Fault Detection (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 the establishment of standard testability metrics. An immediate benefit will come with a consistent, precise, 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 we strive to establish concurrent engineering practices, the interchange between the testability function and other functions becomes even more important. To create integrated diagnostic environments, where the elements of automatic testing, manual testing, training, maintenance aids, and technical information work in concert with the testability element, 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 an accident 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 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 is meant to encompass 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

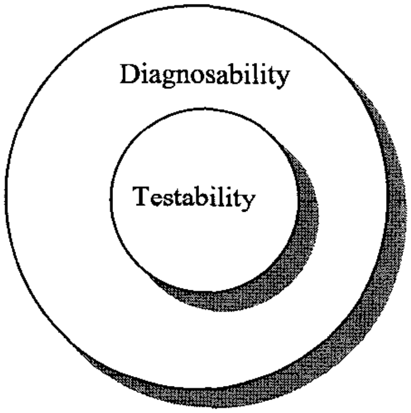

Figure 1. Relationship Between Diagnosability and Testability

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.

OBJECTIVES

It is the objective of the P1522 standard to provide notionally correct, inherently useful, and mathematically precise definitions of testability metrics and characteristics. It is expected that the metrics may be used to either measure the testability 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. Beyond that, the measures must be either measurable or predictable. The former may be used in the specification and enforcement of acquisition clauses concerning factory and field-testing, and maintainability of complex systems. The latter may be used in an iterative fashion to improve the factory and field-testing and maintainability of complex systems. The most useful of all are measures that can be used for both. Because of the last point, the emphasis will be on measurable quantities (metrics).

Things that can be enumerated by observation and folded into the defined figures-of-merit will be developed into metrics. However, a few measures are inherently useful on the design side even if they are not measurable in the field and they are defined in a separate section in P1522. The end purpose is to provide an unambiguous source for definitions of common and uncommon testability and diagnosability terms such that each individual encountering the metric can know precisely what that metric measures.

ASSUMPTIONS

The development of a diagnostic capability includes system level analysis. As such, it is assumed that a system level approach is undertaken, and those diagnostic strategies and testability criteria have been explicitly developed or at least structured. These may be variables in the formulation, but cannot be completely undefined. The primary assumptions are twofold and deal with inherent usefulness from prior experience and the ability to precisely define the term from first principles. In some cases, we will assume the existence of a diagnostic model such as one based on the IEEE 1232 series of standards. Metrics will be derived from the entities and attributes based on these information models. In other cases, we will rely on a demonstrated ability to measure items related to the testing at the design, factory, and field levels concerning the maintainability of complex systems. In the latter case, information models will be developed as necessary to define all relevant entities and attributes.

Each term carries with it a number of additional assumptions (such as single or multiple failure) and is explicitly dealt with on a term by term basis in the section on metrics and characteristics.

ISSUES

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 all *detected* faults detected by BIT/ETE. [1] False alarms were excluded from the definition. From these two variations grew many others. As noted in "Organizational-Level Testability" [3]FFD can be defined as:

- Fraction of all faults detected or detectable by BIT/ETE
- Fraction of all detectable faults detected or detectable with BIT/ETE
- Fraction of all faults detected through the use of defined means. Defined means implies all means of detection that have been identified.
- Percentage of all faults automatically detected by BIT/ETE
- Percentage of all faults detectable by BIT/ETE
- Percentage of all faults detectable on-line by BIT/ETE
- Percentage of all faults and out-of-tolerance conditions detectable by BIT/ETE
- Percentage of all faults detectable by any means

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 built from the ground up.

This overloading is so severe, for example, that there was no definition of FFD is *System Test and Diagnosis*, [4] 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 require 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.

Another aspect of the challenge is that many metrics sound different but are not. Below are some examples.

- *Ambiguity Group Isolation Probabilities* is the cumulative probability that any detected fault can be isolated by BIT or ETE to an ambiguity group of size L or less.
- *Fault Isolation Resolution* is the cumulative probability that any detected fault can be isolated to an ambiguity group of a targeted size or less.
- *Isolation Level* is the ratio of the number of ambiguity groups to the total number of isolatable components.
- *System Operational Isolation Level* is the percentage of observed faults that result in isolation to *n* or fewer replaceable units.

All of these terms were and are valuable. The value of these terms will be increased with precise meanings for each one.

APPROACH

The AI-ESTATE standards have a number of goals.

(1) Provide a standard interface between diagnostic reasoners.

(2) Provide formal data specifications to support the exchange of information relevant to the techniques commonly used in system test and diagnosis.

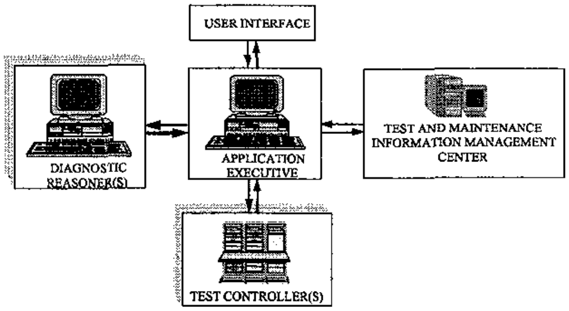

Figure 2. AI-ESTATE Architecture

(3) Maximize compatibility of diagnostic reasoning system implementations.

(4) Accommodate embedded, coupled, and stand-alone diagnostic systems.

(5) Facilitate portability, reuse, and sharing of diagnostic knowledge.

According to IEEE Std 1232-1995, the AI-ESTATE architecture is "a conceptual model" in which "AI-ESTATE applications may use any combination of components and intercomponent communication"[5] . We note that the intent of AI-ESTATE is to provide a formal, standard framework for the exchange of diagnostic information (both static and dynamic) in a test environment. This exchange occurs at two levels. At the first level, data and knowledge is exchanged through a neutral exchange format, as specified by IEEE Std 1232.1-1997 [6]. At the second level, information is exchanged as needed between software applications within the test environment. This information includes entities as read in from a model or information on the current state of the diagnostic process [7].

AI-ESTATE assumes the presence of an "application executive." We emphasize that this application executive need not be a physically separate software process but can be identified as a "view" of the software process when it involves the communication activity. This view of the architecture is shown in Figure 2.

The two component standards of AI-ESTATE focus on two distinct aspects of the stated objectives. The first aspect concerns the need to exchange data and knowledge between conformant diagnostic systems. By providing a standard representation of test and diagnostic data and knowledge and standard interfaces between reasoners and other elements of a test environment, test, production, operation, and support costs will be reduced.

Two approaches can be taken to address this need: 1) providing interchangeable files (1232.1); and 2) providing services for retrieving the required data or knowledge through a set of standard accessor services (1232.2). AI-ESTATE is structured such that either approach can be used.

The 1232.1 standard defines several formal models, including a common element model, a fault tree model, and an enhanced diagnostic inference model. The common element model defines information entities, such as a test, a diagnosis, an anomaly, and a resource, which are expected to be needed by any diagnostic system. The common element model also includes a formal specification of costs to be considered in the test process. The proposed standard for Diagnosability and Testability Metrics (P1522) uses all the models developed in 1232.1 and 1232.2 as a basis to precisely define metrics. How the information models are used will be discussed in the Information Model section.

The second aspect concerns the need for functional elements of an AI-ESTATE conformant system to interact and interoperate. The AI-ESTATE architectural concept provides for the functional elements to communicate with one another via a *communications pathway*. Essentially, this pathway is an abstraction of the services provided by the functional elements to one another. Thus, implementing services of a reasoner for a test system to use results in a communication pathway being established between the reasoner and the test system.

AI-ESTATE services (1232.2) are provided by reasoners to the other functional elements fitting within the AI-ESTATE architecture. These reasoners may include diagnostic systems, test sequencers, maintenance data feedback analyzers, intelligent user interfaces, and intelligent test programs. The current focus of the standards is on diagnostic reasoners.

INFORMATION MODELING

ISO 10303–11 (EXPRESS) and ISO 10303–12 (EXPRESS-I) are used to define information models and exchange formats for diagnostic knowledge [8], [9]. 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 [10]. The intent is to ensure that definitions of these entities are unambiguous.

For example, central to the test and diagnosis problem is the definition of a "test." If we ask a digital test engineer what a test is, it is possible that the answer will be something like "a set of vectors used to determine whether or not a digital circuit is working properly." On the other hand, if we ask a diagnostic modeler what a test is, the answer is likely to be "any combination of stimulus, response, and a basis for comparison that can be used to detect a fault."

On the surface, these two definitions appear very similar; however, there is a fundamental difference. For the digital test engineer, there is an implicit assumption that a "test" corresponds to the entire suite of vectors. For the diagnostic modeler, individual vectors are tests as well.

As a similar example, the test engineer and diagnostic modeler are likely to have different definitions for "diagnosis." The act of doing diagnosis, for most test engineers, corresponds to running tests after dropping off of the "go-path." For the diagnostic modeler, since "no fault" is a diagnosis, the entire test process (including the go-path) is part of doing diagnosis.

It may appear that we are "splitting hairs," but formal definition of terms and information entities is an exercise in splitting hairs. Further, such hair-splitting is essential to ensure that communication is unambiguous—especially when we are concerned with communication between software processes. No assumption can go unstated; otherwise, the risk exists that something will be misunderstood. Information models formally state all of the assumptions.

METRICS SOURCES

Currently, the AI-ESTATE subcommittee is gathering information from the DoD and industry about model representations, their associated metrics, and the processes put in place to utilize them. The results of this review will form the basis for defining the metrics to be included in the standard and the procedural guidance to be included in a proposed "Recommended Practice."

The approach being taken to develop this standard involves defining testability and diagnosability metrics based on standard information models. Specifically, it was found that the AI-ESTATE models provided an

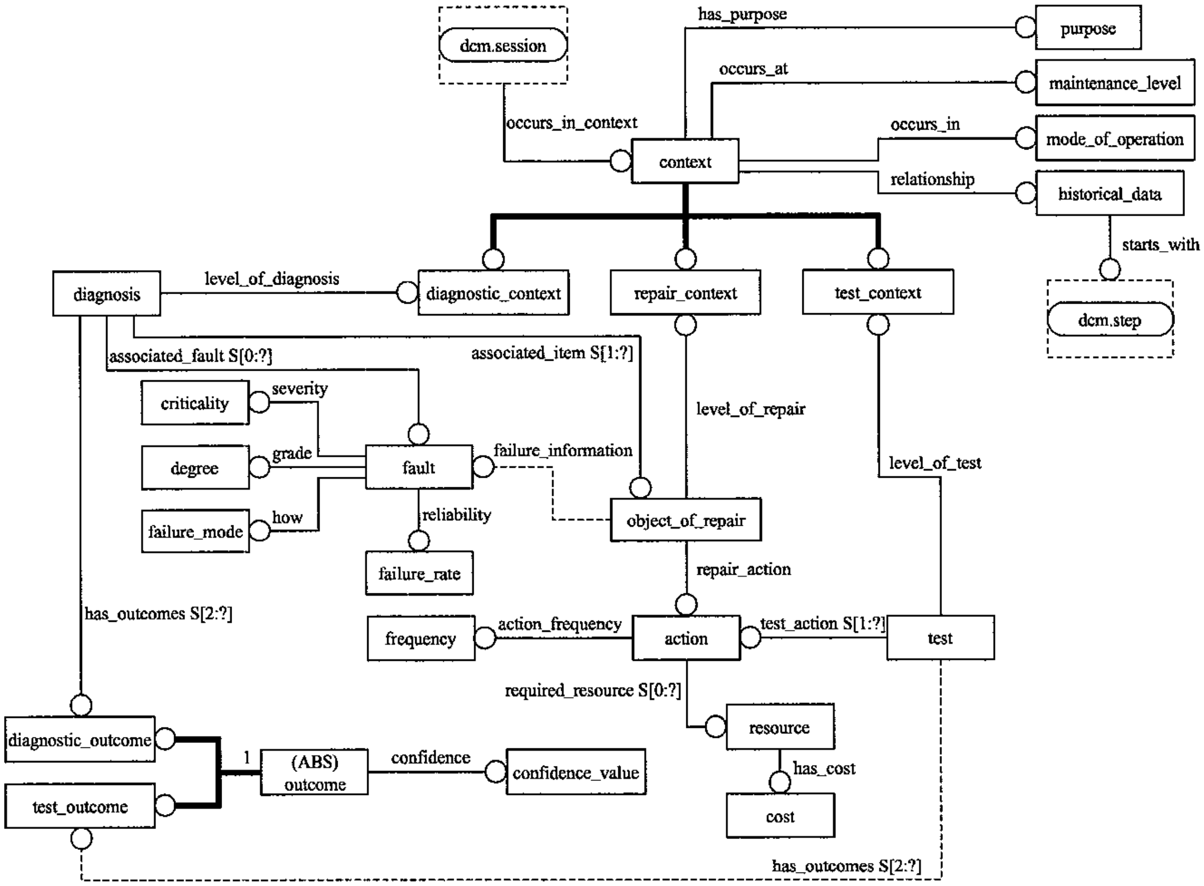

Figure 3. Revised Common Element Model

excellent foundation for defining these metrics. As an example, one metric defined using the model is Fractions of Faults Detected (FFD).

The FFD metric assumes the existence of a diagnostic model that ties tests (especially test outcomes) to potential faults in the system analyzed. Within AI-ESTATE, tests, diagnoses, and faults are modeled explicitly in the common element model. In addition, AI-ESTATE includes specifications for two additional diagnostic models—the fault tree model and the Enhanced Diagnostic Inference Model (EDIM). Due to its generality, the EDIM was used to define FFD.

The assumptions used to define FFD are as follows:

- We are interested in the various metrics at a particular level;
- A hierarchical element exists at a particular level;
- No descendant of a hierarchical element is at the same level as that hierarchical element; and
- At this point, we do not care about the ordering of the levels.

Referring to Figure 3, each diagnostic_outcome has a diagnosis associated with it. There may be one or more outcomes. Each diagnosis has a set of associated_fault and a set associated_item. This means that the fault does not have to manifest itself in the failed item. The associated_item is the physical

```
FUNCTION ffd(model:EDIM.edim; lvl:CEM.level) : REAL;
      LOCAL
            diag_count : INTEGER;
            diags : SET [0:?] OF EDIM.inference
            detect_set : SET [0:?] OF CEM.diagnosis := NULL;
      END_LOCAL;

      diag_count := SIZEOF(QUERY(tmp <* model.model_diagnosis |
      tmp.level_of_diagnosis = lvl);
      REPEAT I := LOINDEX(model.inference) TO HIINDEX(model.inference);
            diags := QUERY(tmp <* model.inference[I].conjuncts |
                         (TYPEOF(tmp) = 'EDIM.diagnostic_inference'));
            diags := diags + QUERY(tmp <* model.inference[i].disjuncts |
                         (TYPEOF(tmp) = 'EDIM.diagnostic_inference'));
            diags := QUERY(tmp <* diags |
                         tmp.pos_neg = negative OR
                         NOT(tmp.diagnostic_assertion = 'Good'));
            detect_set := detect_set + QUERY(tmp <* diags.for_diagnosis |
                         tmp.level_of_diagnosis = lvl);
      END_REPEAT;
      RETURN(SIZEOF(detect_set) / diag_count);

END_FUNCTION;
```

Figure 4. Sample Metric Definition in EXPRESS

location of the failure. A simplified version of FFD is the sum of all detectable diagnoses at a particular level of indenture over the sum of all diagnoses in the model at that same level. A detectable diagnosis is a diagnosis for which there exists an inference of an associated diagnostic outcome other than "good."

From these assumptions and the information models, we can define FFD using the procedural constructs of EXPRESS. Specifically, a function (FFD) can be specified as in Figure 4. In the process of defining this metric, several issues with the Common Element Model were discovered.

Further details of the FFD model can be found in AI-ESTATE – The Next Generation [11].

STATUS AND ROADMAP

P1522 is in preliminary draft form. Many of the changes to the AI-ESTATE standard have a direct relationship to P1522. In the coming months the P1522 standard will be modified and readied for ballot. During this same timeframe, the Revised Version of IEEE Std 1232 will be readied for ballot. During the development of P1522, it was discovered the information models of 1232 were not complete and sufficiently robust to fully support the needs of P1522. Further, the three AI-ESTATE standards are being merged into a single, cohesive document to support consistency and traceability.

CONCLUSION

The AI-ESTATE standards promise to facilitate production testing and long-term support of systems, as well as reducing overall product life-cycle cost. This will be accomplished by facilitating portability, knowledge reuse, and sharing of test and diagnostic information among embedded, automatic, and stand-alone test systems within the broader scope of product design, manufacture, and support.

With the maturing of the AI-ESTATE standard, the opportunity to use this standard to provide formal, unambiguous definitions of testability and diagnosability characteristics and metrics is before us. The diagnostic and maintenance control standards committee is focusing on capitalizing on the formal work for the AI-ESTATE standards to make the testability/diagnosability standard a reality.

AN INVITATION

AI-ESTATE is constantly seeking out people in government, industry, and academia to assist in developing the standards. Anyone interested in the work of AI-ESTATE is encouraged to get involved. Information on the progress of the standards and list of essential personnel is available at http://grouper.ieee.org/groups/1232

ACKNOWLEDGMENTS

In many ways, it is unfortunate that a paper such as this includes only the names of two authors. The work reported here is the result of efforts of a committee of devoted volunteers who have supplied their expertise in system test and diagnosis to develop strong, sound standards supporting the diagnostics community. We would like to thank Tony Bartonlini, Eric Gould, Les Orlidge, Randy Simpson, Tim Bearse, Tim Wilmering, Greg Bowman, Dave Kleinman, Lee Shombert, Sharon Goodall, Len Haynes, Jack Taylor, and Helmut Scheibenzuber for their efforts in developing the standards and promoting their use in industry.

REFERENCES

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

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

[3] 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.

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

[5] IEEE Std 1232-1995. 1995. Trial Use Standard for Artificial Intelligence and Expert System Tie to Automatic Test Equipment (AI-ESTATE): Overview and Architecture, Piscataway, New Jersey: IEEE Standards Press.

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

[7] 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.

[8] 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.

[9] ISO 10303-12:1997. Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 12: Description Methods: The EXPRESS-I Language Reference Manual, Geneva, Switzerland: International Organization for Standardization.

[10] Schenk, D. A. and P. R. Wilson. 1994. Information Modeling: The EXPRESS Way, New York: Oxford University Press.

[11] Sheppard, J. and Kaufman, M. 1999, "AI-ESTATE – The Next Generation", AUTOTESTCON '99 Conference Record, New York: IEEE Press.