Recent Advances in IEEE Standards for Diagnosis and Diagnostic Maturation

John W. Sheppard
Department of Computer Science
The Johns Hopkins University
3400 North Charles Street
Baltimore, MD 21218
jsheppa2@jhu.edu

Timothy J. Wilmering
Phantom Works
The Boeing Company
PO Box 516, M/C S034-1240
St. Louis, MO 63166
timothy.j.wilmering@boeing.com

Abstract—Efforts by the Department of Defense to increase use of commercial or dual-use technologies have resulted in the levying of new requirements on automatic test systems. One of the areas where requirements are being levied is in the exchange of diagnostic and maintenance information. These requirements have led to the creation or revision of several IEEE standards intended to support such information exchange. In this paper, we explain the nature of the revisions being made to IEEE STD 1232 (AI-ESTATE). We also explain the nature of the information being modeled for IEEE P1636 (SIMICA) and its relationship to AI-ESTATE. Finally, we provide a discussion of a new XML-based exchange format being incorporated into AI-ESTATE, SIMICA, and related standards to satisfy exchange requirements under the DoD and industry-led Automatic Test Markup Language (ATML) initiative and explain the role of AI-ESTATE and SIMICA within the larger scope of ATML.

TABLE OF CONTENTS

1. INTRODUCTION

Recent work within the Department of Defense on standardizing open architectures for weapons systems and support systems has led to requirements being levied on the exchange of diagnostic and maintenance information. The IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems has been developing standards for diagnostic knowledge exchange and diagnostic services with their IEEE STD 1232-2002 Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE). Because of the new DoD requirements, it was determined that a significant revision to the AI-ESTATE standard was required.

In addition to the new exchange requirements, members of the diagnostic community have indicated an interest in defining a standard for Bayesian diagnostics. Bayesian diagnostic models involve specifying random variables corresponding to tests and diagnoses utilizing a network structure to relate the random variables to one another. With each node in the Bayesian network is a specification prescribing the conditional probabilities of each of the values of that node given the "parent" (or dependent) nodes in the network.

Current and emerging requirements addressing concerns in diagnostic accuracy and diagnostic system maturation have also led to a new IEEE standards project—IEEE P1636 Standard Software Interface for Maintenance Information Collection and Analysis (SIMICA). Currently, SCC20 is examining maintenance and logistics data from both commercial and military sources in an effort to construct a formal information model for SIMICA to support diagnostic maturation. Two significant areas of standardization currently underway include defining captured test result data from automatic test equipment and defining information captured by a diagnostic reasoner during actual test sessions.

In this paper, we explain the nature of the revisions being made to the AI-ESTATE standard and describe how the Bayesian diagnostic model is being standardized within SCC20. We also explain the nature of the information being modeled for SIMICA and its relationship to AI-ESTATE. We also provide a discussion of a new XML-based exchange format being incorporated into AI-ESTATE and SIMICA to satisfy exchange requirements under the DoD and industry-led Automatic Test Markup Language (ATML) initiative and explain the role of AI-ESTATE and SIMICA within the larger scope of ATML.

0-7803-9546-8/06/$20.00 © 2006 IEEE
Paper 151, Revision 4

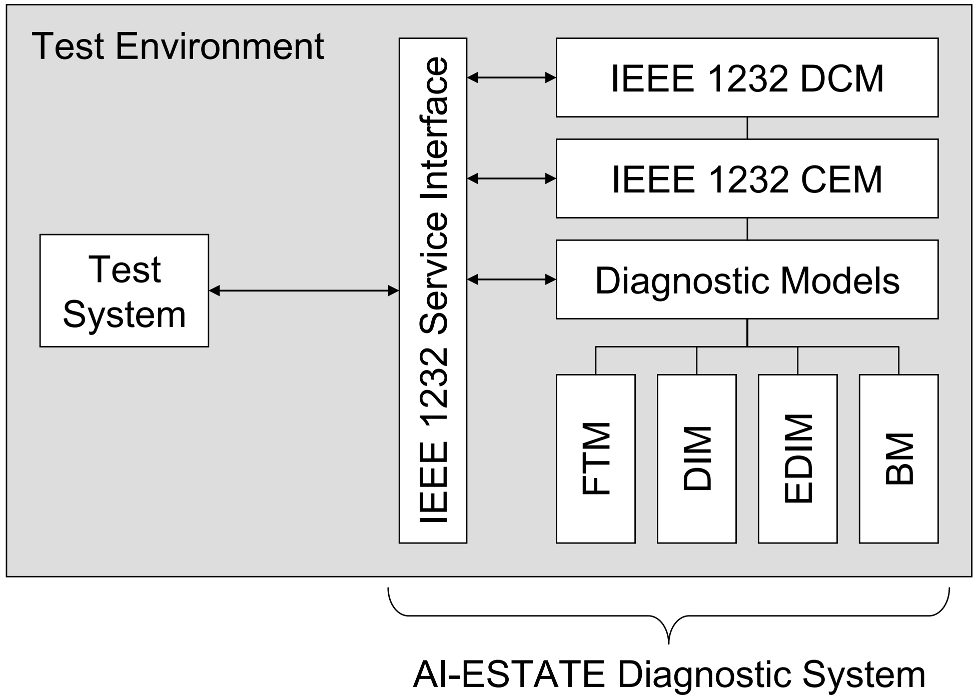


Figure 1–AI-ESTATE architecture. The AI-ESTATE-compliant diagnostic system (or reasoner) communicates with the remainder of the test system via a set of standard services. These services provide access to several "models," including the Common Element Model (CEM), various technique-specific diagnostic models (i.e., fault tree (FTM), diagnostic inference (DIM), enhanced diagnostic inference (EDIM), and Bayesian BM)), and a dynamic context model (DCM) to handled internal state of the reasoner.

2. Diagnostic Standards

The SCC20 Diagnostic and Maintenance Control (DMC) subcommittee is developing a family of standards ([1], [2]) that are product information exchange standards for test, diagnosis, and maintenance. The original standards developed by the DMC, the 1232 series, provided a means of exchanging information between diagnostic reasoners. The complete 1232 standard, which was published in November 2002 as a full-use 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. 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 as well as maintenance history information.

The basic architecture of an AI-ESTATE conformant diagnostic system is shown in Figure 1. AI-ESTATE is defined by a set of standard information models and software services, facilitating unambiguous communication of diagnostic information between the reasoner and the rest of the test system. The Common Element Model (CEM) provides a set of generic information entities, expected to be applicable to all types of diagnostic systems. AI-ESTATE also includes several "technique-specific" models supporting diagnostic reasoning based on fault trees (FTM), diagnostic inference models (DIM), enhanced diagnostic inference models (EDIM), and Bayesian models (BM—one of the subjects of this paper). AI-ESTATE also provides a unique information model, called the dynamic context model (DCM), that supports managing or abstracting diagnostic state for access and control by the rest of the test system.

In 1997, the DMC began to work on a new standard—IEEE STD 1522—focusing on expanding the work of MIL-STD 2165 that had been converted into a handbook. The approach taken to develop this replacement 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 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. This standard was recently published by the IEEE Standards Association as a "trial use" standard [2].

The Management of Test and Maintenance Information Standard (formerly IEEE P1389) is being re-worked and expanded as IEEE P1636 Software Interface for Maintenance Information Collection and Analysis (SIMICA). As a member of the SIMICA family, IEEE P1636.1 Standard for Test Session and Results Information defines an exchange mechanism for test results using XML. This standard is intended to serve as a replacement for the recently withdrawn IEEE STD 1545-1999, Standard for Parametric Data Logging [3].

3. Bayesian Information Model

The intent of the AI-ESTATE standard [1] is to provide a formal information model for the diagnostic domain to support unambiguous exchange of diagnostic information and a consistent software interface for diagnostic systems. The basis of AI-ESTATE is a set of formal information models that are used to represent the information required to support diagnostic reasoning in several forms. The Common Element Model specifies elements that are generally applicable to all reasoning approaches, while the Fault Tree Model, Diagnostic Inference Model, and Enhanced Diagnostic Inference Model provide support for specific approaches to diagnosis.

Currently SCC20 is revising AI-ESTATE to include a model to cover Bayesian diagnosis. In Figure 2, we present a new information model that extends the AI-ESTATE standard such that Bayesian networks can be represented. This figure depicts the model using a graphical modeling language called EXPRESS-G [5], which corresponds to a subset of EXPRESS.

The Bayesian network information model captures information necessary for creating diagnostic Bayesian networks. Assumptions made with this model include that random variables corresponding to tests can only depend on diagnosis variables and other test variables. Diagnoses have no dependencies. In addition, the probability tables are to be fully explicated (including closure, i.e., summing to one across dependent joint distributions), and array position in the probability array corresponds to array position in the dependence array.

Tests and diagnoses are incorporated from the AI-ESTATE Common Element Model with two types of attributes added to these entities. First, probabilities are associated with test outcomes (e.g., PASS and FAIL) and diagnosis outcomes (e.g., GOOD, CANDIDATE, and SUSPECT). These probabilities, defined as a list, provide the conditional probability tables for the respective random variables. These tables go with the second attribute—the "dependsOnElement" attribute—that identifies the dependence relationship between random variables. Note that the original confidence attribute on these entities corresponds to pass/fail outcome probabilities ($\Pr(o(P$ and $\Pr(o(F))$) and diagnosis probabilities ($\Pr(D_i)$) respectively, all of which are specified in the full lexical EXPRESS model.

AI-ESTATE also defines several "standard services" for a diagnostic reasoner to use within a larger test environment. The reason for defining such services is to facilitate "plug-and-play" compatibility across reasoners. These standard services work directly with the new Bayesian model [1]. First, all "accessor" services are defined relative to any entity or attribute within the AI-ESTATE information model (including extended models). Second, the higher-order services, related to reasoner control and diagnostic inference, do not depend on the specifics of the underlying model. In other words, the services do not specify whether the inference process is using a fault tree, a diagnostic inference model, or a Bayesian network; therefore, the same services will work directly with the new model.

Note that the model shown in Figure 2 provides a level of generality beyond most modern diagnostic Bayesian models. Specifically, most models assume there are no dependence relationships between tests where the model in Figure 2 allows for such dependencies (by including the attribute "dependsOnElement L[0:?]," which is defined as a "select type" that can be instantiated as a bayesTest). One can argue that such dependence relationships are not required. In fact, including them could add unnecessary computational burden to any inference algorithm that processes the network; however, SCC20 decided to include the relationships to provide a more general structure in the event some tests within the system are not conditionally independent. In addition, methods of reducing computational complexity exist, such as treating diagnostic Bayesian networks as naïve Bayes "multi-nets [6]." Such networks simplify computation by assuming conditional independence between all of the tests even when such dependence relationships are known to exist. By permitting tests to depend on other tests, these independence assumptions can be relaxed by permitting some of the dependencies to be added back [7].

4. Modeling Failure Distributions

During the balloting process of the AI-ESTATE standard, a significant, unintended "error" was introduced. Specifically, model modifications made to address ballot comments concentrating on modeling cost and, in particular, failure probability resulted in any connections between faults, failures, and failure rate being deleted inadvertently from the model. As part of the revision process, these connections are being restored and a more robust model of failure probability is being introduced.

The model for failure distribution is shown in Figure 3 and will be included in the AI-ESTATE CEM. Specifically, to address this error, every diagnostic conclusion (i.e.,

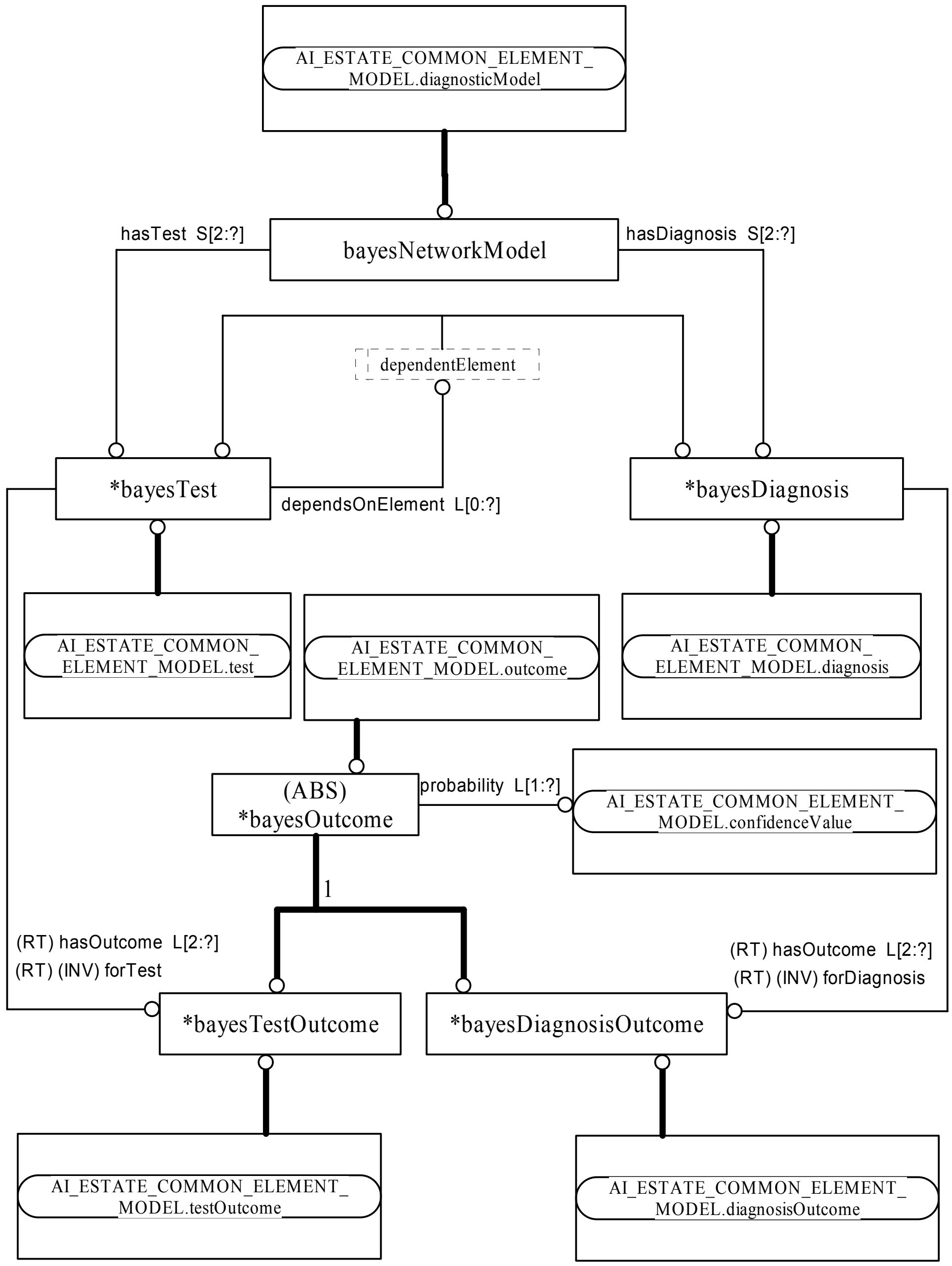


Figure 2–New AI-ESTATE information model to provide for Bayesian diagnosis. The model imports entity definitions from the AI-ESTATE Common Element Model. Formal constraints are defined in the lexical version of the model.

diagnosis) in the AI-ESTATE model includes an optional attribute labeled "hasDistribution," that relates the diagnosis entity to the failureDistribution entity shown in the figure. The entity "diagnosis" is defined in 1232 to be a supertype of both "fault" and "failure," thus permitting the diagnostic process to focus on either physical faults or functional failures (or both).

Of particular interest is the generality of this model compared to traditional "exponential" distributions used in traditional reliability analysis. Specifically, an abstract failure distribution is defined based on the generalized gamma distribution:

$$f(t) = \frac{\beta}{\Gamma(k)\theta}\left(\frac{t}{\theta}\right)^{k\beta-1} e^{-(t/\theta)^{\beta}}$$

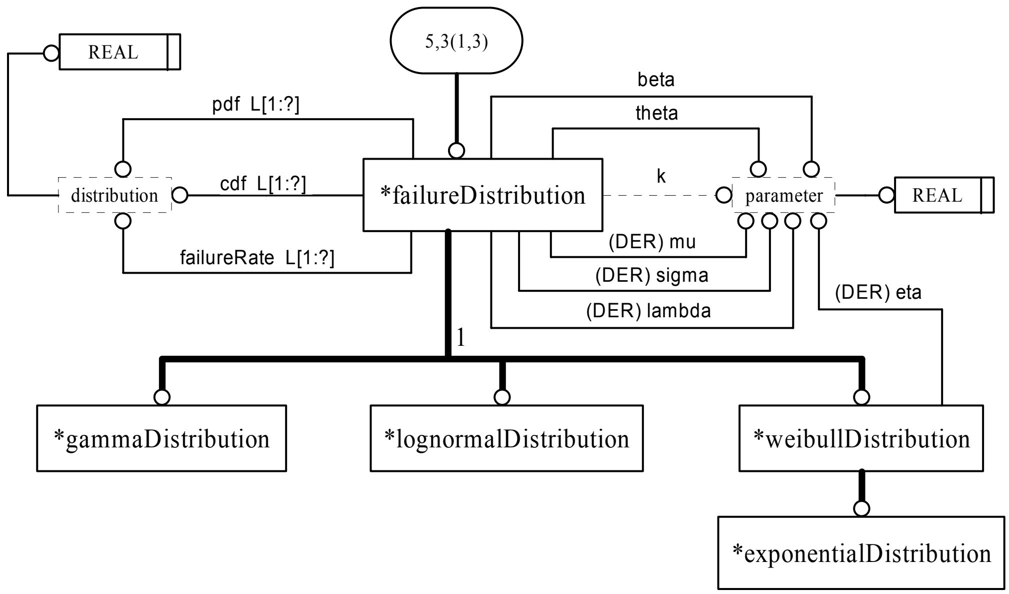


Figure 3–AI-ESTATE failure distribution information model. Failure distributions are defined as special cases of the "generalized gamma distribution."

for parameters β, k, and θ. Note that these parameters are not the typical parameters used to define failure distributions but are required to enable a general definition over all distributions. These parameters enable us to define the actual failure distributions as specializations of this abstract distribution, where otherwise this modeling would be more cumbersome. Mathematically, we can define the typical parameters μ, σ, and λ as follows:

$$\mu = \ln(\theta) + \frac{1}{\beta}\ln\left(\frac{1}{\lambda^2}\right)$$

$$\sigma = \frac{1}{\beta\sqrt{k}}$$

$$\lambda = \frac{1}{\sqrt{k}}$$

From here, each of the more common distributions can be defined. These distributions include the exponential distribution, the Weibull distribution, the lognormal distribution, and the gamma distribution.

Given these parameter calculations, we can develop an information model that includes definitions of subtype distributions as follows. First, the gamma distribution is a special case of the generalized gamma distribution where $\lambda = \beta\sqrt{k} = \sigma$. The lognormal distribution is also a special case corresponding to $\lambda = 0$. When $\lambda = 1$ and $\sigma = 1$, then the generalized gamma distribution is equivalent to the exponential distribution. Finally, the Weibull distribution arises when $\lambda = 1$ and $\beta = 1/\sigma$. Thus, any of the common failure distributions can be represented with the above model.

5. Standards for Diagnostic Maturation

5.1 Motivation

The former IEEE P1389 Management of Test and Maintenance Information Standard was initiated due to growing industry recognition of the need for a specification for access and exchange of diagnostic and maintenance product information. It is generally recognized that initial test and maintenance solutions that are fielded with new systems are generally less than perfect and are initially liable to contribute substantially to system ownership costs where those solutions are deficient. The organizations that deliver complex systems are rapidly becoming cognizant of the need to monitor the effectiveness of their product health management solutions in their customers' application domains.

The effort to "mature" a supportable design begins at the conceptual design stage and continues throughout the system life cycle. The data used for analysis in the early stages of diagnostic analysis and design is more readily available to the analyst than that which must be obtained from disparate sources after the subject product is delivered to the customer. Once a system is fielded and begins to be used in an operational environment, unexpected and unplanned system level design interactions, operational and

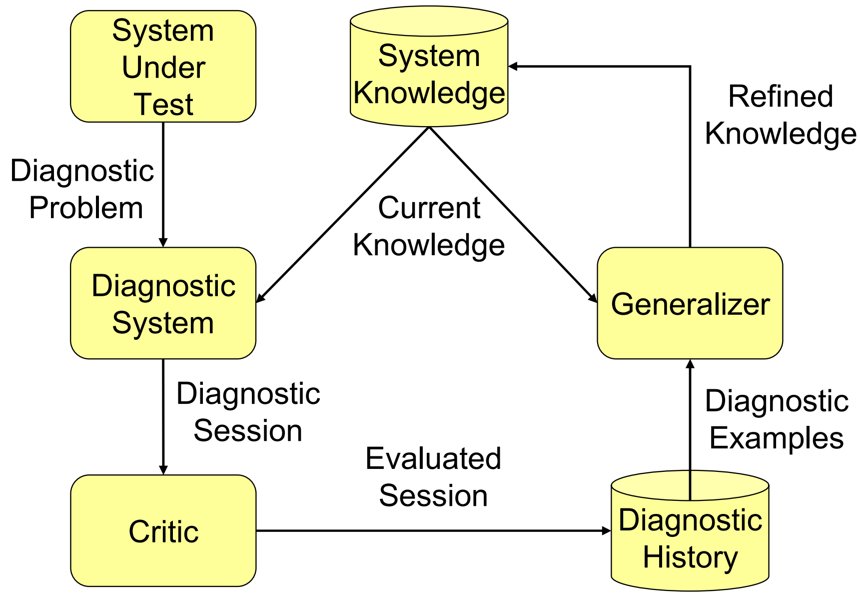


Figure 4–A diagnostic maturation process utilizing techniques from data mining and machine learning. As a "closed loop" process, diagnostic maturation feeds the results of diagnosis to a critic that evaluates the results of diagnosis and stores the results in a composite history for the system. A "learning" agent analyzes the diagnostic history to generalize and adapt the knowledge with the goal of refining and improving the diagnostic process.

environmental stresses, and other influences can degrade the performance of the diagnostic design from what was predicted.

When this degradation results in a system readiness issue or cost of ownership problem remedial actions must be taken. Deviations from the supportability design requirements and performance levels predicted in the earlier design phases must be analyzed for the operational process elements that are related to the performance issues. Significant deviations trigger an iterative closed loop process of root cause analysis - corrective action deployment and reevaluation—called a Maturation Cycle, or more formally in some circles, the FRACAS (Failure Reporting and Corrective Action System) process. In either case, the goal is to determine a corrective action that prevents or minimizes recurrence of the reported problem in subsequent use of the product. The process typically includes failure analysis, which in the context of FRACAS refers to the logical, systematic examination of a failed item to identify and analyze the mechanism and exact cause of failure. Corrective action is usually a drawing, model, process, software, or procedure change. Root causes of supportability problems can have many different sources, but when the cause is found to be a deficiency in the diagnostic test or test procedures then the issue must be addressed by the diagnostic maturation process. A Maturation Cycle or FRACAS activity should primarily be initiated as a function of system supportability performance monitoring; however, customer requests and other internal investigations can also trigger a cycle. In either case the result is a need for relevant, dependable data from valid design and maintenance information resources.

One approach to considering the diagnostic maturation process is shown in Figure 4. Here, the focus is on maturation after fielding; however, the process can be generalized to include the entire system life cycle. In this process, the "system under test" is tested and diagnosed. The results of test and diagnosis are captured for offline analysis, and the results of any analysis stored in the offline database as well. Various techniques such as data mining and machine learning can be applied to the historical data to refine the knowledge and processes used for perform test and diagnosis. The refinement closes the loop.

The diagnostic maturation process therefore requires ready access to design, maintenance, and other logistics support information sources. The heterogeneous nature of these sources possesses unique challenges to those who would extract meaningful knowledge from them. The state of current technology is such that the physical constraints having to do with access are easing, but consistent answers to the questions involving content understanding and integration remain considerable challenges [8].

What is required is an integrated information infrastructure for diagnostic maturation to simplify the management, access and delivery of product definition and supportability data used for complex products. Because the vast majority of this data is in existing systems, it is essential to provide support for tools and processes that can consolidate and access existing design baselines wherever they reside and however they may be represented (i.e., relational databases, flat file repositories, etc).

5.2 P1636 SIMICA

Accordingly, The IEEE SCC20 Diagnostic and Maintenance Control subcommittee has undertaken the task of developing a set of standards that fulfills this need. The potential scope of the maintenance information domain is sufficiently large that a set of component standards is being created, with P1636 SIMICA serving as the document which describes the relationships between the component standards and providing a top level schema that represents the relationships between the component information models. The goal of the family of standards is to provide standard, unambiguous definitions of maintenance information semantics, interrelationships, and access services. Together, these specifications will define a comprehensive formal information model for maintenance information related to the maturation of diagnostic systems and as such are directly related to IEEE STD 1232—Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE), but with equally close ties to emerging specifications in other related test information domains. Specifically, the goals of these specifications are to:

- Provide definitions of maintenance concepts and terminology relevant to the maturation of diagnostic systems.
- Provide an information model to serve as a basis for unambiguous interpretation and communication of data.
- Support the development of an efficient and usable means of moving such data between conforming applications.

The specifications will provide an implementation independent specification for a software interface to information systems containing data pertinent to the diagnosis and maintenance of complex systems consisting of hardware, software, or any combination thereof. These interfaces will support service definitions for creating application programming interfaces (API) for the access, exchange, and analysis of historical diagnostic and maintenance information. The use of formal information models will facilitate exchanging historical maintenance information between information systems and analysis tools, supporting the creation of open system software architectures for maturing system diagnostics [9].

The approach taken to developing SIMICA was to first create a process model that detailed all of the steps taken in the test, diagnosis, and repair of system components, and to then enumerate the data elements that were generated at each step of the process. As one can imagine, this yielded a very large set of potential data elements for consideration. The next step was to identify those elements that were most significant with regard to their impact on the maturation of test and diagnostic procedures. These elements are now being categorized into clusters of related information to provide the partitioning required to decompose the information domain into discrete schemata that will compose the SIMICA family of standards. The first of these schemas to emerge as a candidate specification describes the actionable information that is collected during a test session.

5.3 Test Session and Result Information

Arguably some of the most pertinent data of interest to the diagnostic maturation process is the results of the tests that were performed on a Unit Under Test (UUT). IEEE P1636.1 Standard for Test Session and Results Information will promote and facilitate interoperability between components of an automatic test system (e.g., between test executive and diagnostic reasoner) where test results need to be shared, facilitating both online and offline analysis. The schema defined in this specification provides a standard format for the transport or storage of both quantitative (measured values) and qualitative (pass/fail determination) test results. The schema design is such that ancillary information such as environmental conditions and system/operator messages may also be stored in an instance document. This information, while not specifically "results", is intended to permit use of an instance document for a variety of purposes, including statistical analysis and diagnostics [10].

5.4 Maintenance Action Information

Of equal importance to the results of the test session is a record of the maintenance actions that were performed as a consequence of the test session. It is becoming clear that one of the key issues in both the integration of diagnostic processes across system levels of indenture and analysis of the effectiveness of test and diagnostic processes is understanding the actions resulting from a diagnosis within the context of the system and repair environments. Of particular interest is the information that uniquely identifies affected system components and supports correlation of related information elements from multiple data sources. Accordingly, the next schema to be developed under the SIMICA umbrella will support the representation and interchange of information that is used to support typical system maintenance processes, i.e., those data elements that support actions associated with the removal, repair, and replacement of system components in a maintenance environment. At the time of this writing, the initial draft information model is being developed with the goal of initiating formal specification development at the upcoming DMC committee meeting.

6. ATML AND INFORMATION EXCHANGE

Recent work within the IEEE has embraced developing exchange formats based on the eXtensible Markup

Language (XML). In a similar vein, a recently-formed consortium of test and measurement industry and government leaders and participants have been participating in a cooperative effort to define a collection of XML schemas to represent test information used by multiple cooperative software entities involved in the test and diagnostic process. Collectively referred to as the Automatic Test Markup Language (ATML), these schemas are intended to promote data reuse and interoperability between test system components. Elements of ATML address the common data requirements used for Test Description, UUT Description, Test Station Requirements, Test Configuration, Test Adapters, Instrument Description, Test Results, and Diagnostics. The basic premise is that test information that conforms to the ATML Schemas can be accessed and manipulated by software tools that co-exist in an ATML test environment [11].

The mission of ATML is to "define a collection of XML schemas that allows ATE and test information to be exchanged in a common format adhering to the XML standard [4]." The XML schemata are being provided as part of the new IEEE P1232, the revision to IEEE 1232-2002. This revision will incorporate both the new Bayesian model and the specific XML schemata for the Bayesian model as well as all information models currently defined in IEEE STD 1232-2002 [1].

As stated in the above "mission statement," the principal goals of the ATML project focus on information exchange [4]. Specifically, the goals related to diagnostics that ATML seeks to achieve are:

1. Establish an industry standard for test information exchange.
2. Allow for managed extensibility of test information.
3. Ensure compatibility with other ATE information-based standards.
4. Allow for information exchange with legacy systems.
5. Create modular descriptions for test environments.
6. Leverage existing technologies in creating test environments.
7. Allow for the use of dynamic test sequences that can change with historical data.
8. Allow for the use of optimization techniques such as artificial intelligence.

The ATML consortium is working with IEEE SCC20 to develop these essential data exchange requirements into normative specifications. The XML Schemata defined by the ATML Consortium support information flow for the data elements to be exchanged between ATML-compliant test system components. The ATML Framework (P1671) is the document that describes the relationships of the component ATML specifications as they are formalized by IEEE SCC20. In some cases, the ATML requirements were already met by existing or in-work SCC20 specifications. For example, ATML diagnostic reasoner requirements were served by utilizing the existing IEEE 2002 AI-ESTATE specification. The focus of the P1232 revision is on goals 1–8 with particular emphasis placed on 7 and 8 (the scope of the Diagnostic and Maintenance Control subcommittee of SCC20). Similarly, the requirements for ATML Test Results and historical diagnostic data were congruous with the in-work P1636 SIMICA family of specifications with emphasis on closed-loop process improvement for diagnostics. In other cases, new specifications, such as the P1671.1 Test Description are being developed to formalize the information requirements.

As each of the projects and their associated information models are completed, XML schemata are being developed to address the "document-level" exchange of diagnostic and historical maintenance information. In addition, the subcommittee is developing interface specifications based on the services defined for the models to support exchange of portions of the models through web services such as Web Service Description Language (WSDL) and web transactions. In each case, XML "snippets" will be exchanged based on schemata developed for the standards.

7. CONCLUSION

Recent initiatives by the DoD have placed additional emphasis on providing mechanisms for exchanging information between support maintenance activities. These initiatives are bringing together government, academic, and industrial partners to work with the IEEE, through SCC20) to create a set of commercial standards meeting the maintenance industries information sharing requirements.

The process by which information-based standards are developed must be a living process to keep pace with changing technologies. Within the test and maintenance community, advances in system design, system performance, miniaturization, sensor technology, and onboard computing (among many others) are changing the way systems are being supported. These advances are further increasing the dependence on exchanging accurate, timely information about the systems being supported.

This paper described two standards projects within SCC20 that are designed to address requirements for exchanging maintenance and diagnostic information. AI-ESTATE [1] has been under development since 1990 and currently exists as both an IEEE and IEC standard. SIMICA [9] is a new project focusing on improving diagnostics and feeding information back for improving the models and services provided by AI-ESTATE. Both projects are working closely with the ATML initiative to ensure they fit within the overall framework for exchanging test information.

Acknowledgments

We would like to acknowledge the work and contributions of the SCC20 Diagnostic and Maintenance Subcommittee in working to mature the P1232 standard and to develop the new P1636 standard. We especially thank Mark Kaufman, Darryl Busch, Michelle Harris, Tony Alwardt, Tim Davis, and John Ralph for contributions made to the development of the standards. We also thank the reviewers whose comments improved the overall presentation of this paper.

References

[1] IEEE 1232-2002 *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)*, Piscataway, NJ: IEEE Standards Press.

[2] IEEE STD 1522-2004. *IEEE Trial Use Standard Testability and Diagnosability Characteristics and Metrics*, Piscataway, NJ: IEEE Standards Press.

[3] IEEE STD 1545-1999. *IEEE Standard for Parametric Data Log Format*, Piscataway, NJ: IEEE Standards Press.

[4] Automatic Test Markup Language Consortium website, http://www.atml.org.

[5] 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.

[6] Friedman, Nir, Dan Geiger, and Moises Goldszmidt, "Bayesian Network Classifiers," *Machine Learning*, 29:131–161, 1997.

[7] Sheppard, John W., Stephyn Butcher, Kaufman, Mark, and Craig MacDougall, "Not-So-Naïve Bayesian Networks and Unique Identification in Developing Advanced Prognostics," *Proceedings of the 2006 IEEE Aerospace Conference*, IEEE Press, March 2006.

[8] Wilmering, Timothy J. "Semantic Requirements on Information Integration for Diagnostic Maturation," *AUTOTESTCON Conference Record*, Anaheim, CA, 2001.

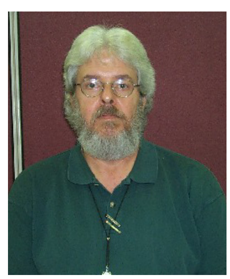
[9] IEEE P1636 *IEEE Standard Interface for Maintenance Information Collection and Analysis (SIMICA)* Draft 0.3 Piscataway, NJ: IEEE Standards Press, January 2006.

[10] IEEE P1636.1 *IEEE Standard for Test Session and Results Information* Draft 1.2 Piscataway, NJ: IEEE Standards Press, January 2006.

[11] IEEE P1671 *Draft Standard Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML*, May 2006.

Biographies

John W. Sheppard was recently appointed as an Assistant Research Professor in the Department of Computer Science, Johns Hopkins University. Dr. Sheppard started his career as a research computer scientist for ARINC and attained the rank of Fellow. His research interests include algorithms for diagnostic and prognostic reasoning, machine learning and data mining in temporal systems, and reinforcement learning. Dr. Sheppard holds a BS in computer science from Southern Methodist University and an MS and PhD in computer science from Johns Hopkins University. He has over 100 publications in artificial intelligence and diagnostics, including an authored book and an edited book. He is a senior member of the IEEE and currently serves as Vice Chair of the IEEE Standards Coordinating Committee 20 (SCC20) on Test and Diagnosis for Electronic Systems, Secretary and Past Chair of the Diagnostic and Maintenance Control subcommittee of SCC20, and Official Liaison of the Computer Society Standards Activities Board to SCC20.



Timothy J. Wilmering is an Associate Technical Fellow with the Boeing Company. His professional interests are centered on Integrated Vehicle Health Management (IVHM) architectures, software tool and information system modeling and integration, and the application of knowledge-based methodologies to these technologies. He has 20+ years of experience in the performance of test, diagnosis, and maintenance procedures both on and off aircraft, diagnostic architecture development and design guidance, development of analytical engineering tools, and definition of enterprise process and information integration standards and tools. His educational background includes a BS in computer science, a Graduate Certificate in artificial intelligence, and a Masters in Information Management. He is a member of the IEEE and co-chair of the Diagnostic and Maintenance Control subcommittee of IEEE Standards Coordinating Committee 20 on Test and Diagnosis for Electronic Systems.