A Standards-Based Approach to Gray-Scale Health Assessment Using Fuzzy Fault Trees

Patrick J. Donnelly, Liessman E. Sturlaugson, John W. Sheppard
Department of Computer Science
Montana State University
Bozeman, MT 59717
{patrick.donnelly2, liessman.sturlaugson, john.sheppard}@cs.montana.edu

***Abstract*—As part of a project to examine how current standards focused on test and diagnosis might be extended to address requirements for prognostics and health management, we have been exploring alternatives for incorporating facilities to represent gray-scale health information in the IEEE Std 1232 Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE). In this work, we extend the AI-ESTATE Common Element Model to provide "soft outcomes" on tests and diagnoses. We then demonstrate how to use these soft outcomes with the AI-ESTATE Fault Tree Model to implement a "fuzzy" fault tree. The resulting model then enables isolating faults within a system such that levels of degradation can also be tracked. In this paper, we describe the proposed extensions to AI-ESTATE as well as how those extensions work to implement a fuzzy fault tree using the demonstration circuit from previous Automatic Test Markup Language (ATML) demonstrations.**

I. Introduction

Current standards for prognostics and health management (PHM) are mostly based upon providing a basic framework for managing applications to be used in a diagnostic environment and to support off-line analysis of performance and historical data in a prognostic context. Since its publication in 2010, we have been examining methods for extending the IEEE 1232-2010 Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE) standard [1] to incorporate elements supporting gray-scale health information with the intent of increasing the ability to provide a standards-based approach to performing on-line prognostic analysis. This paper evaluates the ability of these proposed changes to support gray-scale health by exploring a concrete model containing uncertainty using a fuzzy fault tree. Prior to publishing AI-ESTATE, we demonstrated the ability of that standard to support interoperable exchange of diagnostic data and seamless integration of diagnostic applications in a larger test environment. As part of this work, we encoded a Fault Tree Model of the AI-ESTATE Demonstration Phase II from AUTOTESTCON 2009 [2].

In a traditional fault tree, as specified by AI-ESTATE, each test returns a single outcome (usually either pass or fail) and that outcome determines a single branch to follow in the tree. The result of applying this outcome might be to perform another test or, at the leaves of the tree, to identify a diagnosis. Given the specific results of the tests, only a single path of the fault tree would be analyzed. *Fuzzy fault trees*, also known as *soft decision trees*, provide a method to encode uncertainty in the diagnostic process. This often requires analyzing multiple branches of the fault tree but yields a set of possible diagnoses, each with an individual degree of membership in the associated candidate set. Each test in the fault tree is assigned a membership function. These membership functions map a given continuous value to a level of membership. The value of the resulting test is assigned to one or more linguistic variables, corresponding a state of the test with the associated membership value. In practice, the diagnostic reasoner could ignore or prune certain branches of the tree if the membership values do not reach a predefined threshold. The defuzzified value for each candidate diagnosis can be utilized as a degree of failure (i.e., gray-scale health) for each fault in the model.

In this paper, we demonstrate our proposed extensions to AI-ESTATE by adapting the fault tree of the ATML demonstration circuit as a fuzzy fault tree. This adaptation incorporates fuzzy sets in the model and applies the proposed extensions to AI-ESTATE necessary to represent degree of failure. Specifically, this paper extends the ATML model to represent gray-scale health, evaluates the model with a fuzzy fault tree, demonstrates a common defuzzification technique, and finally, evaluates the ability of our proposed extensions to AI-ESTATE to support PHM.

II. Background

A. Fault Trees

The diagnostic process can be defined as a fault isolation procedure that uses information from system observations and tests [3]. More specifically, traditional fault diagnosis is often regarded as a process whereby an ambiguity group is refined successively based on performing tests. This view can be described in the context of set theory whereby an ambiguity group corresponds to the set of candidate diagnoses consistent with some set of test results. Thus a final diagnosis consists of the intersection of the sets consistent with each individual test outcome.

Fault tree based tools have existed since the 1970s [2], and fault trees are one of the most common ways to formalize a fault isolation procedure, as described above [3]. Formally, we define a fault tree as a graph $FT = (V, E)$ as follows.

- Let $(w, v) \in E$ be some edge in FT.

978-1-4673-0700-0/12/$31.00 ©2012 IEEE

- Given edge $(w, v) \in E$, let $\text{Pa}(v) = w$ denote the unique parent of v in FT. In other words, any $v \in V$ can have at most one parent.
- Let $r \in V$ denote the unique vertex in V such that $\text{Pa}(r) = \text{NIL}$. We call this the root of FT.
- For any $v \in V$ and fault universe F, Let $\tau_v(u)$ be a function
$$\tau : V \to A$$
where $A \subseteq F$, corresponds to the result of performing a test on some object u in our universe of discourse U. In other words, u corresponds to our unit under test (UUT), and $\tau_v(u)$ returns the ambiguity group $A \subseteq F$ corresponding to the result of performing test τ_v on our UUT.
- Let $T_v^{FT}(u) = \tau_v(u) \cap T_{\text{Pa}(v)}^{FT}(u)$, where $T_{\text{Pa}(r)}^{FT}(u) = F$.

Given this definition, we can think of a path through the fault tree as a sequence of intersection operations applied to the various ambiguity groups as a result of performing the corresponding tests along that sequence.

Thus a fault tree is a specific type of decision tree, wherein the interior nodes indicate specific tests to be run during the fault isolation procedure, and each possible outcome of a particular node's test becomes a branch of that node that points to further tests or to the final diagnoses at the leaves [4]. Each path from the root to the leaves represents a sequence of tests to reach a diagnosis, and thus a fault tree attempts to model the relevant test sequences, often optimized by some set of criteria [5]. The AI-ESTATE Fault Tree Model was specified to support this type of diagnostic decision tree but also includes the ability to associate intermediate diagnoses in interior nodes of the tree and to associate confidence information with tests and diagnoses [1].

Typically, when building a fault tree, the fault isolation procedure uses the information gain of successive tests to reduce the ambiguity of a diagnosis [5]. But what if there is uncertainty associated with test outcomes themselves? Furthermore, what if the actual failure state of a UUT corresponds to a level of degradation rather than a hard failure? With the traditional fault tree, the outcome of a particular test on an interior node determines the branch to follow, and the leaves of the fault tree identify hard faults. One way to relax these assumptions in the fault tree is to associate grades of fuzziness with the different outcomes of the tests. Fuzzy logic can then be used to evaluate the degree of failure associated with each possible diagnosis at the leaves of the tree. We refer to this degree of failure as a *gray-scale health assessment* of the associated diagnosis.

B. Fuzzy Sets

Fuzzy set theory enables a form of approximate reasoning [6], in which the membership functions associated with sets in a domain of discourse U are extended to permit a degree of membership. Specifically, in traditional crisp set theory, we say some object $u \in U$ either belongs to a set A or does not belong in set A. This can be represented as $\mu_A(u) = 1$ or $\mu_A(u) = 0$ respectively. Thus, in crisp set theory, a membership function is defined as
$$\mu_A : U \to \{0, 1\}.$$
Fuzzy sets, on the other hand, permit a degree of membership, given by
$$\mu_A : U \to [0, 1]$$
where $\mu_A(u) = 1$ and $\mu_A(u) = 0$ correspond to the crisp limits of the fuzzy sets, but $0 < \mu_A(u) < 1$ can also occur.

This idea can be extended to combining fuzzy sets by replacing the consistency constraint to utilize the fuzzy membership functions. To do this, we need to define the corresponding fuzzy set operators. Of particular interest to us here is fuzzy intersection; however, definitions for all fuzzy set operators exist. Specifically, one definition for fuzzy intersection (also known as the t-norm) corresponds to
$$\mu_{A \cap B}(u) = \min\{\mu_A(u), \mu_B(u)\}.$$
and another (which is used in fuzzy diagnostics) corresponds to
$$\mu_{A \cap B}(u) = \mu_A(u) \times \mu_B(u).$$
The introduction of fuzzy rules allows reasoning about these membership values while managing the uncertainty associated with real-valued set membership.

C. Fuzzy Fault Trees

Fuzzy set theory and fuzzy logic have previously been extended to decision trees to manage the uncertainty of following particular paths through the tree. The general fuzzy decision tree consists of interior nodes for branching on a set of possible outcomes, as before, but now each of the outcomes are possible with varying degrees of truth. The fuzziness associated with different outcomes are defined by the user, while the final set memberships are evaluated at the leaves, representing the various sets (i.e., diagnoses) [7].

More formally, given our definition of a fault tree above, we replace our test function $\tau_v(u)$ with membership function $\mu_v(u)$. Then the basic process is similar to that above whereby we maintain a running t-norm along the paths of the tree. Note that this process is complicated by the fact we may need to consider multiple paths. The resulting membership value in the candidate set for each fault must be determined by "defuzzifying" the resulting subsets. Several approaches exist for defuzzification; however, the one we used in our demonstration is the "center of area" method. This can be done as follows. Assume we are defuzzifying the candidate status for some fault $f \in F$ after applying our fuzzy fault tree FT. Then the defuzzified value for f, $z_{\text{coa}}(f)$, is computed as
$$z_{\text{COA}}(f) = \frac{\int_f \mu_{FT}(f) \cdot I(f)\, df}{\int_f \mu_{FT}(f)\, df}$$
where $\mu_{FT}(f)$ is the result of combining the running t-norms along the paths in the fault tree for fault f, and $I(f)$ acts in the numerator as an indicator function to limit the integral to the results consistent with fault f alone.

The fault tree and associated fuzzy membership functions can be hand-built by a domain expert based on the uncertainty associated with particular tests and on the accuracy and sensitivity of the instruments used for the test. Alternatively, the fuzzy membership functions can be derived from the data, as in [8], which calculates the membership values based on fuzzy entropy measures. Also, data-driven techniques for programmatically constructing fuzzy decision trees can be found in [9], [10], [11]. An application of a fuzzy fault tree, specifically, can be found in [7].

D. AI-ESTATE

Since approximately 1989, the IEEE has been developing the AI-ESTATE standard with the intent of providing the means for exchanging diagnostic models and defining a standard application programming interface (API) for interacting with a diagnostic reasoner. The most recent version of AI-ESTATE, IEEE Std 1232-2010 [1], accomplishes this by specifying four semantic models, defined in the EXPRESS information modeling language [12], for fault diagnosis based on fault trees, *D*-matrices, logic models, and Bayesian networks. The Extensible Markup Language (XML) schemata have been derived from these semantic models conformant to the Standards for the Exchange of Product model data (STEP) from the International Organization for Standardization (ISO) [13]. In addition, several standard services have been defined to enable a test or diagnostic application to interact with a diagnostic reasoner in a standard way. While it is expected that the reasoner would most likely process one of the AI-ESTATE models, this is not a requirement of the standard. The effectiveness of AI-ESTATE in model exchange and reasoner interoperability was demonstrated in 2008 [14] and 2009 [2] respectively.

E. ATML

The goal of the Automatic Test Markup Language (ATML) family of standards (IEEE Std 1671) [15] is to define a collection of XML-based schemata that allows Automatic Test Systems (ATS) and test information to be exchanged in a common format adhering to the XML standard [16]. The 2008 and 2009 demonstrations focused on exchanging XML-based ATS information using the IEEE ATML and AI-ESTATE standards. For these demonstrations, a simple low frequency analog UUT was designed to facilitate both testing and diagnostics [2]. A circuit schematic of the designed UUT for the second phase of this demonstration is shown in Fig. 1.

The UUT consists of seven resistors, five capacitors, and a transistor. In addition, three digital components were added in Phase II to increase the complexity of the demonstration unit. The tests are defined as follows:

- V_{CC} resistance (i.e., continuity) test
- AC voltage test at V_O
- AC voltage test at V_C
- Test for high DC voltage at V_C
- Test for low DC voltage at V_C
- Test for high DC voltage at V_E
- Test for low DC voltage at V_E
- Test for high DC voltage at V_B
- Test for low DC voltage at V_B
- Test for Gain Control
- Test for Output Control

For purposes of creating our fuzzy fault tree, we consider the following fifteen failure modes:

- Resistor 1 - Open (R1.OP)
- Resistor 2 - Open (R2.OP)
- Resistor 3 - Open (R3.OP)
- Resistor 4 - Open (R4.OP)
- Capacitor 1 - Open (C1.OP)
- Capacitor 2 - Open (C2.OP)
- Capacitor 3 - Short (C3.SR)
- Transistor 1 - Open at Collector (Q1.C.OP)
- Transistor 1 - Short at Collector (Q1.C.SR)
- Transistor 1 - Open at Base (Q1.B.OP)
- Transistor 1 - Open at Emitter (Q1.E.OP)
- Transistor 1 - Base–Emitter Short (Q1.BE.SR)
- Transistor 1 - Base–Collector Short (Q1.BC.SR)
- Gain Control Failure
- Output Control Failure

III. A Proposed Extension to AI-ESTATE

One of the limitations facing AI-ESTATE is the current inability to represent gray-scale health information. Presently, the standard limits states of tests, actions, and diagnoses to have discrete outcomes. The inclusion of gray-scale health information would provide a means to "roll up" failure progression to higher levels in the system hierarchy, while still supporting reasoning about the current state of degradation and projecting future failure conditions. By incorporating gray-scale health information, the standard would relax the outcome-based approach to diagnosis to support health estimation based on real-valued or soft test results. The approach described in this paper would support legacy and future diagnostics by having discrete outcomes as a special case while simultaneously being extensible to support requirements for PHM.

We propose that gray-scale health information be represented in the AI-ESTATE standard using a set of soft outcomes that use a basis function or a mixture of several basis functions to determine underlying health state in the presence of uncertainty. This general approach would allow interoperability of the standard utilizing a variety of specific implementations, such as fuzzy logic, artificial neural networks, mixtures of Gaussians, Bayesian networks with continuous random variables, or, as illustrated in this paper, fuzzy fault trees.

A. Soft Outcomes

To include gray-scale health information in AI-ESTATE, the majority of the proposed changes would be limited to the

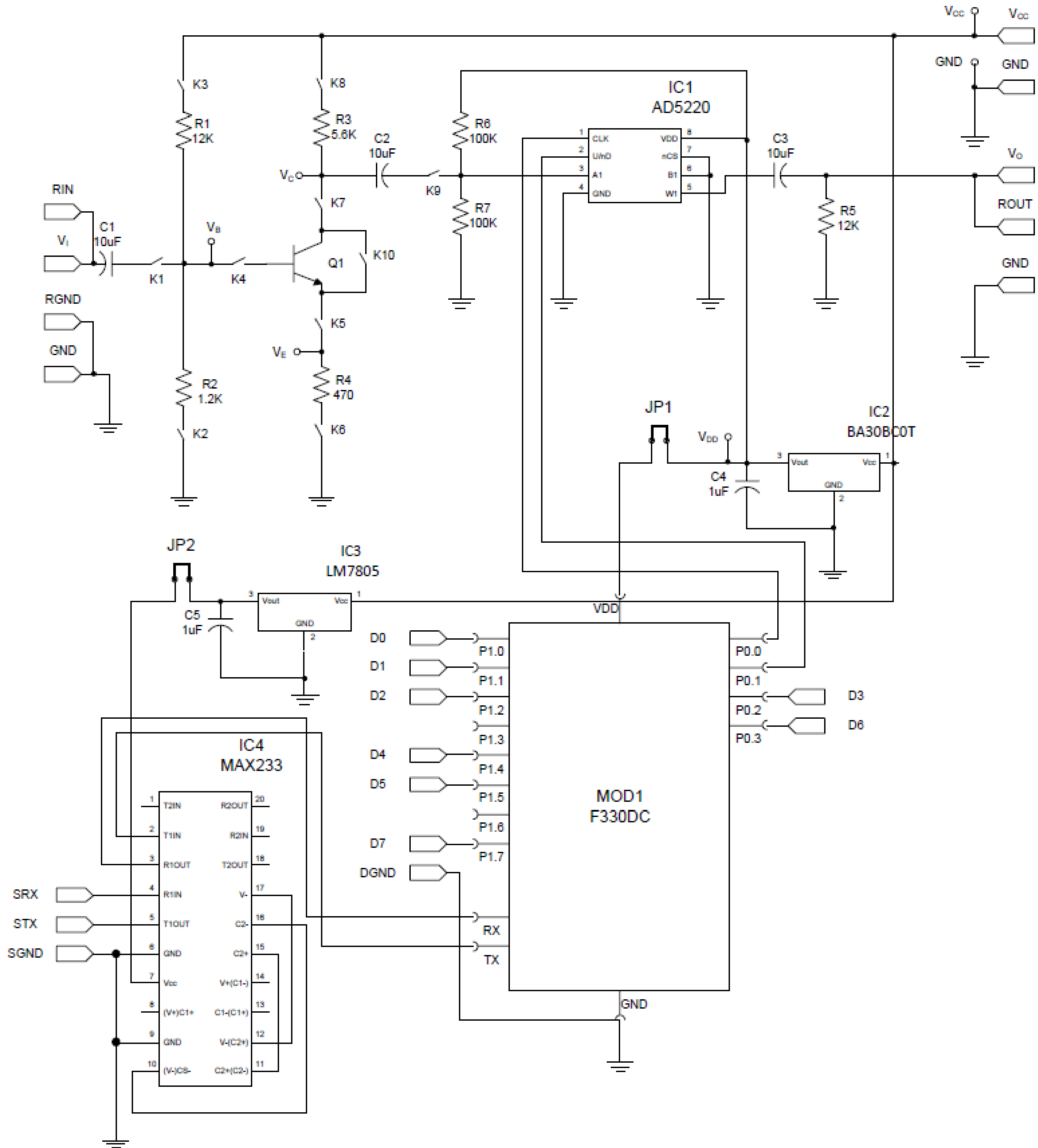


Fig. 1. ATML Phase II Demonstration UUT circuit diagram.

Common Element Model (CEM). Currently the standard restricts outcomes to discrete outcomes (e.g., GOOD, BAD, and CANDIDATE for diagnoses; PASS, FAIL, and UNKNOWN for tests). The abstract entity Outcome contains the attribute allowedValue that represents a single discrete value associated with the specific outcome based on the subtype instantiating the Outcome entity.

Our proposed changes to allow gray-scale health information would require modification of the allowedValue attribute of the Outcome entity (Fig. 2). Specifically, we propose redefining allowedValue to utilize a new SELECT type—CrispSoft—that serves as a selector of the type the outcome value is to be. This allows the Outcome to instantiate either the current implementation of a single discrete OutcomeValue attribute or alternately instantiate a new SoftOutcomeValue that represents individual possible gray-scale outcomes. The SoftOutcomeValue entity permits an Outcome to specify membership in one or more possible SoftOutcomeValues, each with a linguistic variable name. An associated degree of membership would be recorded at test time.

The new entity SoftOutcomeValue contains two attributes. Attribute “name” of type NameType represents the linguistic label assigned to the soft outcome, such as Failed, Degraded, or Good. The names of these linguistic variables could be

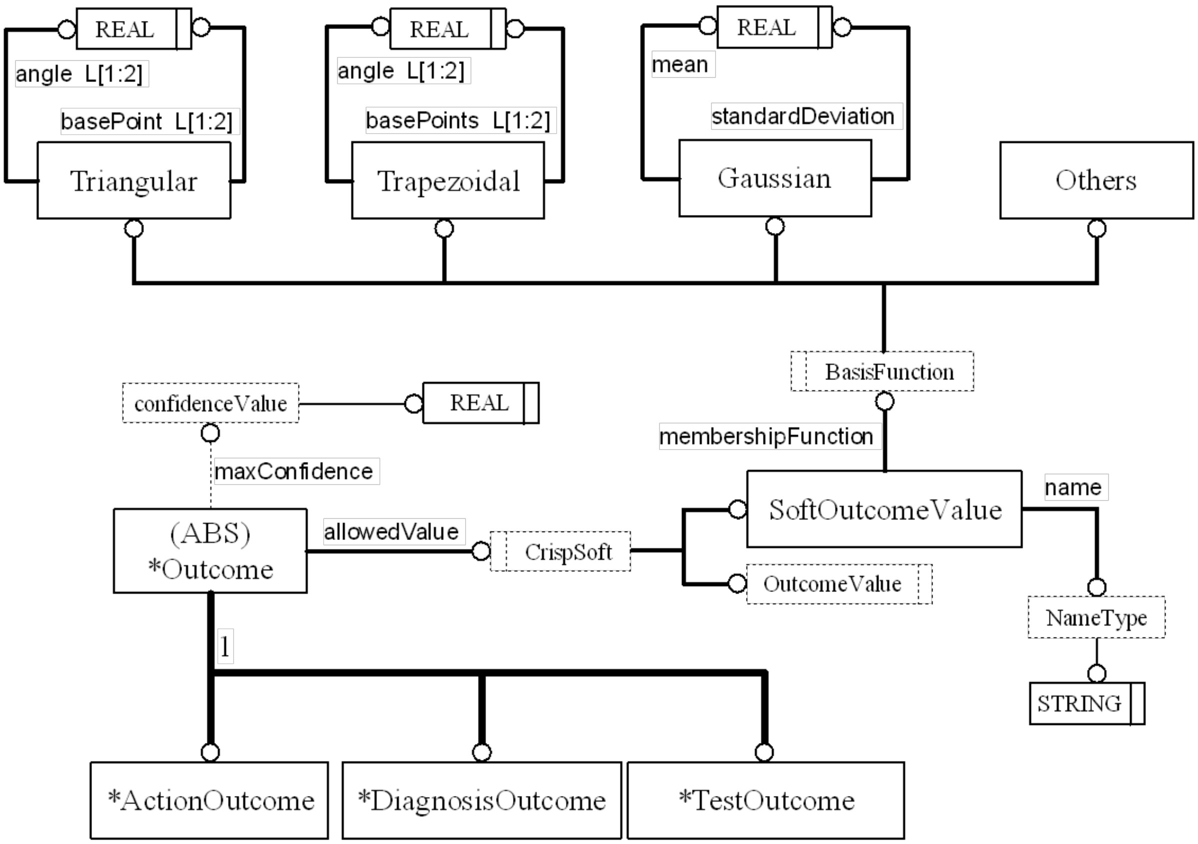


Fig. 2. Suggested modifications to the AI-ESTATE Common Element Model.

left to be defined entirely by a specific implementation or could be defined within the standard. The second attribute, "membershipFunction," is defined using the BasisFunction SELECT type. This SELECT type associates a particular SoftOutcomeValue with a single basis function entity that models how the soft outcome's value associated with that linguistic variable will be determined at test time. Essentially, this function bounds the associated degree of uncertainty.

In the present standard, the Outcome entity contains the optional attribute maxConfidence, which is of type Confidence. The semantics for this attribute were defined relative to the original crisp outcome. Consequently, given the potential ambiguity that might arise from incorporating a "maximum confidence" with a soft outcome when the shape of the basis function determines this property, we propose adding a rule to the standard whereby, if the type of the outcome is soft, then this attribute would not be instantiated. Thus, for purposes of our discussion here, the maxConfidence attribute is not used.

Since actually modeling Outcomes occurs at the subtype level, soft outcomes would be inherited down to all implementing subtypes. This would include the DiagnosisOutcome, permitting flexible representation of gray-scale health information. Additionally, TestOutcome and ActionOutcome would also inherit this ability to represent soft outcomes, permitting the potential to handle the continuous representation of test measurements. Note that the proposed modification would permit outcome types to be "mixed and matched" as needed by the end model.

B. Basis Functions

As described above, the SoftOutcomeValue entity contains a SELECT type that associates the SoftOutcomeValue with a single basis function. Each basis function contains common attributes that define its individual shape. For example, a triangular basis function requires a list of two angles and the length of the base. A Gaussian basis function, on the other hand, requires the mean and standard deviation of the function. The Other entity is a placeholder to demonstrate that any number of other common basis functions (e.g., a sigmoidal or sinusoidal function) could be included to reflect the ability of this approach to extend to a specific target application.

Including a variety of common basis functions permits flexibility to any specific implementation using gray-scale soft outcomes. For instance, triangular and trapezoidal functions are common membership functions in a fuzzy logic system, while the hard limiter, linear threshold, and logistic functions are common activation functions used in neural networks. Additionally, a mixture of Gaussian functions could be used

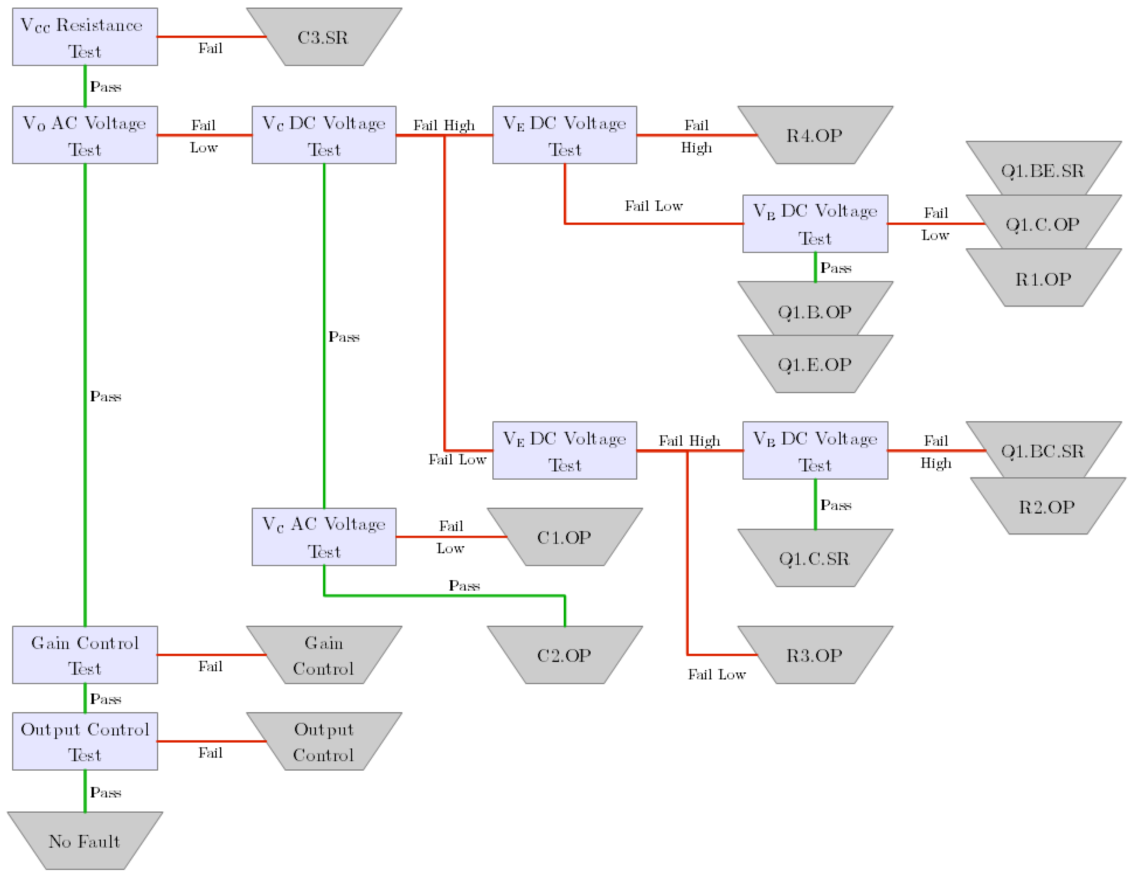


Fig. 3. Fault tree for the ATML circuit.

in the implementation of a Bayesian network with continuous random variables.

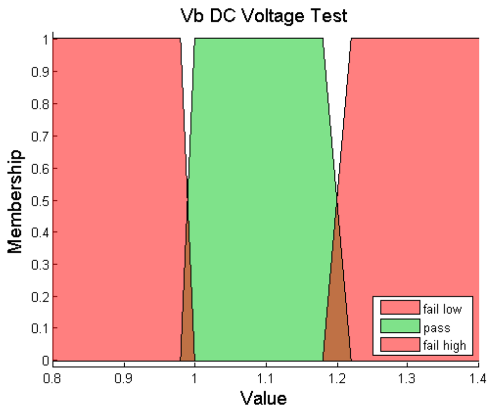
While the standard should include a number of general common basis functions, an additional user-defined basis function should be included to permit flexibility to any implementation incorporating gray-scale health, replacing the Other entity in the model. The specification of this user-defined basis function is to be determined based on feedback from the standards committee, but including such an entity would then permit users of AI-ESTATE to apply the standard extension mechanism to incorporate whatever basis function they need.

IV. A Fuzzy Fault Tree for the ATML UUT

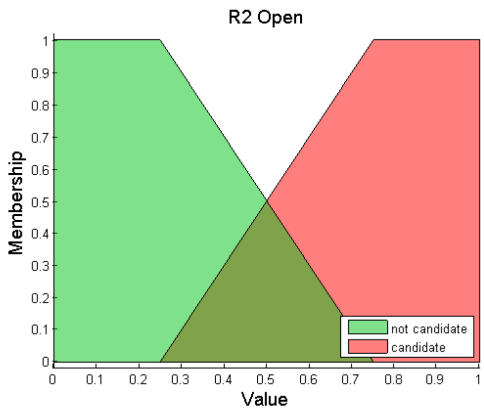
To illustrate the utility of our proposed extensions to AI-ESTATE to support gray-scale health, we have encoded a concrete example. Beginning with the fault tree from the ATML demonstration discussed in Section II-E, we have created a fuzzy fault tree (Fig. 3) of the ATML UUT. The underlying assumption in building this tree (which may or may not be valid) is that the degree of membership associated with a test result arises due to that test detecting a possible level of degradation in the UUT.

Three of the tests being applied to this circuit have three possible test outcomes: *Pass*, *Fail Low*, and *Fail High*. In the crisp version of the ATML fault tree each of these is represented as two separate tests, each with the binary outcomes of *Pass* or *Fail*. One test asks if the test fails low or passes while the other asks if the test fails high or passes. In the fuzzy fault tree, however, these tests are represented with three possible outcomes. Each of these outcomes is represented by a trapezoidal membership function with overlapping areas of uncertainty. Fig. 4(a) shows an example trapezoidal function for the V$_B$ DC Voltage Test.

As an example of gray-scale health being derived from a fuzzy fault tree, suppose we perform the tests and get the results as shown in Table I. The result of each test shown in the Value column determines the corresponding membership value for each respective test outcome shown in the Outcome column. The first four tests have a membership value of 1 to the outcome shown, while the final test has non-zero membership value to the two outcomes shown. Notice that this list of tests follows a distinct path through the fault tree in Fig. 3. In this example, the first test, V$_{CC}$ Resistance Test, passes while the next test, V$_O$ AC Voltage Test, fails. The



(a) Membership function for Test Vb DC Voltage.



(b) Membership function for Diagnosis R2-OP

Fig. 4. Examples of fuzzy membership functions for a Test (a) and a Diagnosis (b).

Test	Value	Outcome (Membership value)
V_{CC} Resistance Test	12.5 KΩ	Pass (1)
V_O AC Voltage Test	0.85 V	Fail (1)
V_C DC Voltage Test	4.5 V	Fail Low (1)
V_E DC Voltage Test	0.6 V	Fail High (1)
V_B DC Voltage Test	1.21 V	{Pass (0.25), Fail High (0.75)}

TABLE I
VALUES OF THE TESTS. OUTCOMES WITH MEMBERSHIP VALUES OF ZERO ARE NOT SHOWN.

Diagnosis	Value
R2.OP	0.72
Q1.BC.SR	0.72
Q1.C.SR	0.66

TABLE II
AMBIGUITY GROUP SHOWING CANDIDATE DIAGNOSES USING COA DEFUZZIFICATION.

third test, V_C DC Voltage, fails low leading to the fourth test, V_E DC Voltage, which fails high. Now consider the final test, V_B DC Voltage. If V_B DC Voltage yields a value of 1.21 V, this value resides in the overlapping region between the membership functions of *Pass* and *Fail High*. Using the common defuzzification technique of "center of area" (COA) as described above, this results in two paths, each with a different level of uncertainty as shown in Table II.

Given the potential modifications to AI-ESTATE discussed in Section III, it would be possible to represent this fuzzy fault tree using the standard. For example, the V_B DC Voltage Test would have three associated TestOutcomes: *Pass*, *Fail Low*, and *Fail High*. The TestOutcomes would use the new CrispSoft SELECT type to store a SoftOutcomeValue instead of the single OutcomeValue currently prescribed in the standard. Each TestOutcome would store its membership function. In this example, these are the trapezoidal membership functions, such as the one shown in Fig. 4(a). Likewise, the Diagnosis associated with Resistor 2 failing open (R2-OP) would have two associated DiagnosisOutcomes: *Candidate* and *Not Candidate*, each storing a SoftOutcomeValue that represents its membership function (Fig. 4(b)).

During the diagnostic reasoning process, using the Dynamic Context Model (DCM), the V_B DC Voltage Test, when applied, would result in an ActualOutcome for each of its outcomes. This is different from a standard fault tree in that we need to record the associated membership values for both *Pass* and *Fail High*. The current Step in the DCM then stores the set of ActualOutcomes in its attribute, outcomesObserved, and these ActualOutcomes point to the corresponding soft TestOutcomes with their associated membership functions. The ActualOutcome also stores the corresponding membership values in the confidence attribute.

Given the information stored in the ActualOutcome entity, the user can apply any defuzzification technique desired. We show the defuzzified values using the "center of area" technique in Table II. This represents the gray-scale health corresponding to the diagnosis. Within the DCM, diagnosis outcomes are stored at each step using the outcomesInferred attributes, which is also of type ActualOutcome. The process is identical to the way test outcomes are captured. Thus, these defuzzified values would be stored as the confidence attribute in the ActualOutcome.

V. CONCLUSION

In this paper, we proposed modifications to AI-ESTATE to extend the ability of the standard to represent gray-scale health. Furthermore, we have translated the ATML fault tree into a fuzzy fault tree containing gray-scale uncertainty. Lastly, we have demonstrated the ability of our proposed extensions to AI-ESTATE to represent this gray-scale uncertainty.

As future work, we will focus on evaluating the ability our proposed changes to the standard to represent uncertainty in the other models supported by AI-ESTATE. These include continuous state approximations of Bayesian networks, fuzzy Bayesian networks [17], and fuzzy D-matrices.

By way of a disclaimer, in any standardization effort, requirements are brought before a standards development organization, such as the IEEE, from which either new standards are developed or existing standards are revised. For standardization efforts to be successful, existing applications should drive the requirements, rather than standards requirements drive the applications. Currently, considerable work is being performed in developing new applications for PHM. These applications are demonstrating the need for diagnostic standards to incorporate capabilities such as the ones presented here. The purpose of this paper is not to propose a need for standardizing on fuzzy fault trees for PHM. Rather, the purpose is to propose general extensions to AI-ESTATE to support PHM requirements while using the fuzzy fault tree as an example of how these extensions might be used.

ACKNOWLEDGMENTS

We thank Mike Malesich, Jennifer Fetherman, and Patrick Kalgren for the many discussions leading to developing the proposed extensions to this standard. We also thank members of the IEEE SCC20 Diagnostic and Maintenance Control Subcommittee for reviewing and discussing an early proposal for these extensions.

REFERENCES

[1] IEEE Std 1232-2010, *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)*. Piscataway, NJ: IEEE Standards Association Press, 2010.

[2] J. Sheppard, S. Butcher, and P. Donnelly, "Demonstrating semantic interoperability of diagnostic reasoners via AI-ESTATE," in *Proceedings of the IEEE Aerospace Conference*, 2010, pp. 1–10.

[3] W. Fenton, T. McGinnity, and L. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 31, no. 3, pp. 269–281, 2001.

[4] J. Sheppard and S. Butcher, "A Formal Analysis of Fault Diagnosis with *D*-matrices," *Journal of Electronic Testing: Theory and Applications*, vol. 23, pp. 309–322, 2007.

[5] J. Pietersma, A. van Gemund, and A. Bos, "A model-based approach to sequential fault diagnosis," in *IEEE AUTOTESTCON Conference Record*, 2005, pp. 621–627.

[6] L. Zadeh, "Fuzzy logic and approximate reasoning," *Synthese*, vol. 30, no. 3, pp. 407–428, 1975.

[7] M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita, "Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems," in *Proceedings of the Third IEEE Conference on Fuzzy Systems*, 1994, pp. 2113–2118.

[8] K. Cios and L. Sztandera, "Continuous ID3 algorithm with fuzzy entropy measures," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, 1992, pp. 469–476.

[9] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 221–254, 2003.

[10] Y. Yuan and M. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 69, no. 2, pp. 125–139, 1995.

[11] M. Dong and R. Kothari, "Look-ahead based fuzzy decision tree induction," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 461–468, 2001.

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

[13] ISO 10303-28:2007, *Industrial Automation Systems–Product Data Representation and Exchange–Part 28: XML Representation of EXPRESS Schemas and Data Using XML Schemas*. Geneva, Switzerland: The International Organization for Standardization, 2007.

[14] J. Sheppard, S. Butcher, P. Donnelly, and B. Mitchell, "Demonstrating semantic interoperability of diagnostic models via AI-ESTATE," in *Proceedings of the IEEE Aerospace Conference*, 2009, pp. 1–13.

[15] IEEE Std 1671-2010, *IEEE Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML*. Piscataway, NJ: IEEE Standards Association Press, 2010.

[16] C. Gorringe, M. Seavey, and T. Lopes, "ATML and dot standards status," in *IEEE AUTOTESTCON Conference Record*, 2008, pp. 69–73.

[17] H. Tang and S. Liu, "Basic theory of fuzzy Bayesian networks and its application in machinery fault diagnosis," in *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 4, 2007, pp. 132–137.