Automatic Systems Diagnosis Without Behavioral Models

Shekhar Gupta
Palo Alto Research Center
USA

Rui Abreu
Department of Informatics Engineering
University of Porto, Portugal

Johan de Kleer
Palo Alto Research Center
USA

Arjan J.C. van Gemund
Department of Software Technology
Delft University of Technology, The Netherlands

Abstract— **Recent feedback obtained while applying Model-based diagnosis (MBD) in industry suggests that the costs involved in behavioral modeling (both expertise and labor) can outweigh the benefits of MBD as a high-performance diagnosis approach. In this paper, we propose an automatic approach, called ANTARES, that completely avoids behavioral modeling. Decreasing modeling sacrifices diagnostic accuracy, as the size of the *ambiguity group* (i.e., components which cannot be discriminated because of the lack of information) increases, which in turn increases misdiagnosis penalty. ANTARES further breaks the ambiguity group size by considering the component's false negative rate (FNR), which is estimated using an analytical expression. Furthermore, we study the performance of ANTARES for a number of logic circuits taken from the 74XXX/ISCAS benchmark suite. Our results clearly indicate that sacrificing modeling information degrades the diagnosis quality. However, considering FNR information improves the quality, attaining the diagnostic performance of an MBD approach.**

TABLE OF CONTENTS

1. INTRODUCTION

In model-based diagnosis (MBD) the cost of the diagnostic process can be broken down into *modeling* and *solution* cost. Solution cost includes *algorithmic* as well as *identification* penalty (pointlessly testing any incorrectly diagnosed candidates), where identification cost is often used as diagnostic utility measure. Traditionally, MBD studies the trade-offs between the above cost dimensions in a model-once-diagnose-often context, where modeling cost is amortized over many observations.

While solution cost has been an important success factor (especially in time-critical applications), a recent design and maintenance case study in Dutch industry suggests that modeling cost is much more of a bottleneck for the acceptance of MBD than previously considered. At ASML a LYDIA-based diagnoser has successfully been used to diagnose faults in an important electro-mechanical subsystem that frequently suffered from failures [1]. It was shown that MBD can reduce solution cost from days to minutes for a once-only investment of 25 man-days of modeling effort (approximately 2,000 lines of code (LOC), comprising sensor modeling, electrical circuits, and some simple mechanisms). Despite the obvious financial gains, management discontinued the project once it became clear that only 80% of the model could be obtained *automatically* from the system's source code [2], [3].

In view of the continuous evolution of a large fraction of the subsystems (component upgrading, new lithographic technology, many machine versions) their reluctance to embrace non-automated modeling for anything else than their core business (lithography) seems understandable. Behavioral modeling is primarily a complex, manual process which can be extremely time-intensive and error-prone. For certain systems, it is even impossible to build behavioral models. For simple components, as found in combinatorial logic circuits, a library approach to behavioral component modeling can amortize much of that cost, reducing the modeling process to compiling the structural information of the circuit into a system model. Still, there remains a considerable manual factor as components evolve and compositionality in complex systems is typically limited.

The fact that real-world software of realistic size still cannot be modeled for the purpose of efficient, automatic debugging [4] has led the software engineering community to investigate approaches that are not based on behavioral modeling, such as *spectrum-based fault localization* (SFL). Unlike MBD, in SFL the dynamic program execution profiles of tests (called spectra, hence the name SFL) is correlated with the test outcomes (pass/fail), typically by using statistical similarity coefficients. The components are subsequently ranked in order of the likelihood that they are responsible for test failures. As the spectra are captured by automatic profiling, and as the test oracles are readily implemented from existing specifications, no modeling effort is required. Benchmark studies, as well as case studies by the authors diagnosing embedded software (100 KLOC) from Philips Semiconductors (now NXP) have shown promising results [5], [6], [7]. Recently, a *model-based* approach to SFL has been presented [8] where the statistical approach has been replaced by a reasoning approach. Grounded in (Bayesian) probability theory, the reasoning approach outperforms the statistical approach, in particular for multiple-faults, at polynomial cost due to a number of approximations within the diagnosis algorithm. In particular, as the reasoning approach is based on a *generic* component model *no* software modeling effort is involved.

In industrial situations where software and hardware is constantly evolving, a critical success factor in the mainstream adoption of MBD is whether modeling can be fully *automated*. Given the results of (model-based) SFL in the software domain, in this paper, we study to what extent

978-1-4799-1622-1/14/$31.00 ©2014 IEEE

(model-based) SFL can offer an alternative to MBD in the logic hardware domain. Apart from the above industrial motivation, there are two additional reasons: (1) the motivation to study the relationship between SFL and MBD, where SFL's greater ability to handle large time series of observation data can partly compensate for its inherently limited precision compared to MBD, as well as (2) the benefits of a unified approach to *simultaneously* diagnosing software and hardware, particularly of interest in the embedded systems domain (e.g., abundant in the aerospace industry), where the root cause of software level failures can now be traced down to the hardware level.

This paper makes the following contributions:

(1) We present a spectrum-based diagnosis approach to logic circuits, which is part of ANTARES (AutoMAtic systems Diagnosis wIthout behaviOral modelS), to generate diagnosers based on circuit topology without modeling the behavior of the circuit's components.
(2) We describe a particular ANTARES feature that automatically estimates the error propagation characteristics of a circuit, a critical parameter that significantly improves the quality of SFL's Bayesian posterior probability computation.
(3) We compare the performance of ANTARES with a state-of-the-art MBD approach (GDE [9]) using the 74XXX/ISCAS85 benchmark suite of logic circuits.

Our results show that for the logic circuits we studied ANTARES is indeed capable of approaching the performance of MBD, provided accurate information for each component is available on the average pass rate of tests (false negative rate, FNR) that cover the component when faulted.

Approaches that abstract specific component behavior, also known as structural diagnosis, have been proposed in the past, e.g., [10], [11], [12]. None of these approaches are able to deal with intermittent faults. The Analytic Redundancy Relation (ARR) based approach [13] is close to our approach. However, (i) it does not scale well for multiple faults, (ii) it is not studied for probabilistic framework, and (iii) it is also incapable of diagnosing intermittent fault. To the best of our knowledge, we are the first to propose the use of SFL in the multiple-fault diagnosis of logic circuits comprising both persistent and intermittent logic. Note that the technique presented in this paper is orthogonal to techniques for automatic testing (the approach in this paper is started once something is found to be failing in order to pinpoint the root cause of the observed failure), such as automatic testing pattern generation [**?**], [**?**].

The paper is organized as follows. In Section 2 we briefly describe the principles behind SFL, as well as the diagnostic utility metric (identification cost) that we use to compare diagnostic performance. In Section 3 we present the ANTARES approach to modeling hardware, featuring an analytic FNR estimation technique. In Section 4 we compare ANTARES with GDE using the 74XXX/ISCAS85 benchmark circuits. In Section 5 we summarize our contributions.

2. SFL

This section briefly reviews SFL. More detailed descriptions can be found in [8], [14]. In SFL the following is given:

- A finite set $\mathcal{C} = \{c_1, \ldots, c_j, \ldots, c_M\}$ of M components of which M_f are faulted.
- A finite set $\mathcal{T} = \{t_1, \ldots, t_i, \ldots, t_N\}$ of N tests with binary outcomes $O = (o_1, \ldots, o_i, \ldots, o_N)$, where $o_i = 1$ if test t_i failed, and $o_i = 0$ otherwise.
- A $N \times M$ (test) coverage matrix, $A = [a_{ij}]$, where $a_{ij} = 1$ if test t_i involves component c_j, and 0 otherwise. Each row is also called a spectrum.

For a Bayesian approach to SFL, the following additional information is also required:

- The prior fault probability of a component c_j, denoted p_j.
- The *false negative rate* (FNR) of a component, denoted g_j, which expresses the probability that a test involving a component c_j, when *faulted*, will still *pass*. In software FNR is related to *coincidental correctness* [15] and *failure exposing potential* [16], while in hardware FNR is related to *failure intermittency* [17].

The result of SFL is a *component ranking* $R =< c_{r(1)}, \ldots, c_{r(j)}, \ldots, c_{r(M)} >$, ordered in terms of decreasing likelihood $\mathsf{Pr}(c_j)$ that c_j is at fault. In statistical approaches to SFL $\mathsf{Pr}(c_j)$ is approximated using statistical similarity coefficients [18]. In this paper we will consider a reasoning approach where the $\mathsf{Pr}(c_j)$ are posteriors based on Bayesian probability theory.

The diagnostic utility of R is measured in terms of the identification cost C_d, which models the verification effort of a diagnostician, going down the suspect ranking R searching for the actual faults (true positives). In particular, we measure the identification effort *wasted* on false positives (i.e., excluding the components found to be faulted). Let c_r denote the actually faulted component that has the lowest posterior in R, where $r \in \{1, \ldots, M\}$ denotes its rank in R. Then $C_d = r - M_f$. In our studies we will typically consider a normalized value $C_d/(M - M_f)$ which ranges from 0 to 1, in order to compare across varying system sizes. Note that $M - M_f$ is the number of actually non-faulted components. This normalized metric is essentially the inverse of the DXC utility metric [19] for diagnosers that produce no false negatives (R includes all components so it cannot miss any faulted component).

Candidate Generation

In ANTARES R is derived from the multiple-fault *diagnosis* $D =< d_1, \ldots, d_k, \ldots, d_{|D|} >$ which is an ordered set of all $|D|$ *minimal candidates*, ordered by decreasing posterior probability $\mathsf{Pr}(d_k)$. Each candidate d_k comprises a *minimal* set of components c_j that, when faulted, are consistent with all test observations (i.e., a minimal diagnosis).

Candidate generation is based on modeling each component by the generic, weak (i.e., faulty behavior is not specified) model[2] given by

$$h_j \implies (\textit{inputs-ok}_j \implies \textit{output-ok}_j)$$

where h_j denotes component health (true when nominal, false when faulted), while *inputs-ok* and *output-ok* denote whether the component's inputs and output are error-free (an error being produced by some faulted component upstream). Depending on the test outcome, each row i in spectrum matrix A yields either a *pass set* ($\{c_j | a_{ij} = 1, o_i = 0\}$) or a *fail set* ($\{c_j | a_{ij} = 1, o_i = 1\}$). It can be easily seen that a fail set is equivalent to a *conflict* (set). Candidate generation is based on computing the minimal hitting sets (MHS) of all fail sets.

[2] Often referred to as abstract modeling in related work.

When faulted components are covered in a test, the fact that components have non-zero FNR leads to many pass sets. While not useful for deriving candidates the pass sets do influence a candidate's posterior probability, and are also useful for speeding up (focusing) the MHS computation.

Probability Computation

Given the typically large number of candidates in D that have equal fault cardinality, for large systems the ranking induced by the posterior probability computation is critical to diagnostic accuracy. For each observation $obs_i = (A_{i*}, o_i)$ the posteriors are updated according to Bayes' rule

$$\mathsf{Pr}(d_k|obs_i) = \frac{\mathsf{Pr}(obs_i|d_k)}{\mathsf{Pr}(obs_i)} \cdot \mathsf{Pr}(d_k|obs_{i-1}) \qquad (1)$$

where $\mathsf{Pr}(d_k|obs_0)$ is computed from the priors according to

$$\mathsf{Pr}(d_k|obs_0) = \prod_{\{j|\neg h_j\}} p_j \cdot \prod_{\{j|h_j\}} (1-p_j)$$

assuming components fail independently. The denominator $\mathsf{Pr}(obs_i)$ is a normalizing term that is identical for all d_k and need not be computed directly. $\mathsf{Pr}(obs_i|d_k)$ is defined as

$$\mathsf{Pr}(obs_i|d_k) = \begin{cases} \prod_J g_j & \text{if } o_i = 0; \\ 1 - \prod_J g_j & \text{if } o_i = 1. \end{cases} \qquad (2)$$

where $J = \{j \mid c_j \in d_k, a_{ij} = 1\}$ is the set of component indices in d_k covered by the test. Eq. (2) assumes an OR-model, i.e., the test may fail if either of the faulted components fail. In general, the OR-model is an acceptable approximation [20], [8], not in the least since D's probability mass is often dominated by single faults, even when the system has multiple faults.

R is derived from D by aggregating the posteriors of each d_k into posterior component probabilities according to

$$\mathsf{Pr}(c_r) \approx \sum_{d_k \in D, c_r \in d_k} \mathsf{Pr}(d_k)$$

The approximation is due to the fact that formally D should be expanded with all *non*-minimal candidates for the above equation to be correct. The reason for our approximation is discussed in the next section.

Implementation Details

In ANTARES we use the STACCATO MHS algorithm [8], [21] for computing D from (A, O). STACCATO exploits pass sets in its any-time computation of the most probable minimal candidates d_k. Typically, the first few hundred candidates practically cover all posterior probability mass, after which the MHS algorithm is terminated. As a result, for random problems comprising $N = 1,000$ tests and $M = 1,000,000$ components of which $M_f = 1,000$ are faulted, the MHS is diagnosed in less than 0.1 CPU second on a contemporary PC [8].

As discussed earlier, our performance metric C_d is formally defined in terms of the full (non-minimal) diagnoses rather than the minimal diagnoses D. For the large 74XXX/ISCAS85 circuits we are considering, however, the posterior probability mass covered by D is virtually equal to unity. As the number of minimal diagnoses is considerably smaller than all diagnoses, the huge computational savings outweigh the small estimation error by far. The reason for the small error is that after multiple observations are combined D already involves all components due to the use of a weak component model and the fact that the observations are random (modeling a practical application situation). Even when multiple faults are present, the latter leads to many observations that can already be explained by candidates of lower cardinality. Had we chosen to limit our observations to MFMC (Max-Fault Min-Cardinality) observations (such as generated by MIRANDA [22]) we would have to extend D with non-minimal candidates (e.g., by a low-cost, first-order extension approach such as in SEQUOIA [14]).

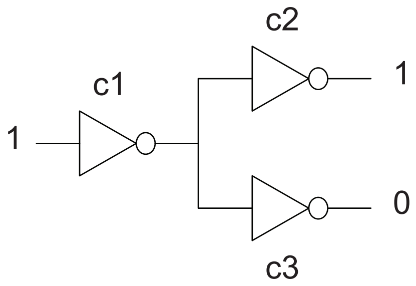

Figure 1. Three-inverter circuit

The posterior probability computation is a straightforward application of Bayes' rule, implemented within the BARINEL toolset [8], using an option to externally read the g_j parameters from file. How the g_j are generated is described in Section FNR Estimation.

3. ANTARES

ANTARES applies SFL to diagnose hardware, exploiting topological information only. Whereas in software the spectra are obtained by tracing the components that are executed per test run (dynamic control flow), in hardware a spectrum originates from a *cone*, i.e., all components involved in the computation of a circuit output (determined by topology). A particular feature of ANTARES is that it includes a method to estimate the g_j parameters, which is vital to diagnostic performance. In the following we outline the principle for logic circuits. Note, however, that the approach generalizes to any causal system.

Consider the example circuit shown in Fig. 1. Each primary output observation is interpreted as one test. Thus one test vector yields two tests, one which involves c_1 and c_2 (the cone of the top primary output), and one involving c_1 and c_3 (the bottom cone). Since SFL assumes the existence of test oracles, we observe a pass for the above output, and a failure for the bottom output. The observations are given in terms of A and O according to

$$\begin{array}{cccc} 1 & 1 & 0 & + \\ 1 & 0 & 1 & - \end{array}$$

where '+' and '-' denote $o_i = 0$ and $o_i = 1$, respectively, to distinguish O from A. The single conflict (c_1, c_3) will generate two minimal candidates c_1, and c_3. Let $p_j = p =$

0.01 (equal component types), and let $g_j = g = 0.5$, under the assumption that a fault will show up at the output in 50% of the cases. SFL (Eqs. 1, 2) yields the following (minimal) diagnosis $D =< c_3(0.67), c_1(0.33) >$, where $\Pr(d_k)$ is in parentheses.

The component ranking $R =< c_3(0.68), c_1(0.37), c_2(0.05) >$ is computed, where $\Pr(c_r)$ is in parentheses[3]. Note that in this small example we have actually computed R from the full, non-minimal diagnosis, in order to include c_2 in the probability computation (since c_2 is not in D). In our 74XXX/ISCAS85 experiments, however, we simply derive R from D with negligible loss of accuracy.

SFL vs. MBD

Let us compare this diagnosis with a diagnosis from MBD. Exploiting the inverter model we can now also propagate values throughout the circuit, leading to a second conflict (c_2, c_3). In SFL terms both conflicts are expressed as

$$\begin{array}{cccc} 1 & 0 & 1 & - \\ 0 & 1 & 1 & - \end{array}$$

which yields the minimal candidates c_3, and (c_1, c_2). Assuming the same priors and FNR we obtain the following diagnosis $D =< c_3(0.99), c_1 \wedge c_2(0.01) >$. The derived ranking R is $< c_3(0.92), c_1(0.21), c_2(0.21)$ (again, from the full, non-minimal diagnosis). Note that the higher defect density estimation ($M_f = \sum_r \Pr(c_r) = 1.34$) compared to the SFL solution is due to the fact that now two fail sets are found vs. one fail set *and* one pass set. The latter exonerates c_1 and c_2 leading to lower posteriors.

Despite the difference in posterior distribution, the diagnostic accuracy of both approaches are equal in terms of C_d. However, the SFL approach suffers from the fact that no modeling information is exploited. This becomes particularly clear when considering D. While MBD correctly infers a single fault (c_3 with 0.99 probability) or a double fault (c_1, c_2 with 0.01 probability), SFL infers a single fault (c_3 with 0.67 probability), and *another single* fault c_1 (with 0.33 probability). As the latter cannot be true SFL suffers from false positives (in terms of D) compared to MBD. However, note that typically a diagnostician will only consider R, which comprises all components anyway. Thus the diagnostic accuracy is effectively determined by the quality of the posterior computation, which is key in the comparison between SFL and MBD.

An aspect in favor of SFL is that it exploits the information of the pass sets, whereas MBD does not (except internally, e.g., for an MHS engine such as STACCATO, which allows better focusing, yielding computational cost reduction). This explains why c_3 is ranked higher than c_1 although, according to SFL, both are single-fault candidates. Exploiting pass set information is one of the reasons why SFL's diagnostic performance is of practical interest.

Ambiguity Groups

In ANTARES A directly derives from the circuit's topology. Each of the circuit's N' outputs generates 1 row in A, leading to N' rows in A per test vector. For multiple test vectors A simply grows in multiples of N' rows. Since, regardless of the number of test vectors, A only contains N' different rows, many of the columns in A are equal. Consider the well-known systems topology in Fig. 2 which (for a single observation) generates the following matrix rows

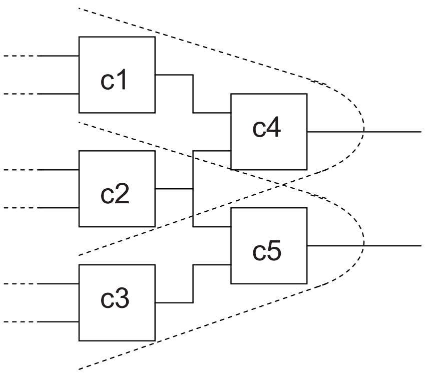

Figure 2. A well-known systems topology

$$\begin{array}{ccccc} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{array}$$

Each of both cones, as well as the cone intersection, generates a set of equal columns in A. The associated components are called an *ambiguity group* (AG). In the above example A has three ambiguity groups $(c_1, c_4), (c_2), (c_3, c_5)$. While the 1-member AG does not pose any problem, the other 2-member AG's introduce a lower bound on C_d since their member components cannot be distinguished unless they have different posteriors. The latter is key to our SFL approach in ANTARES.

Ambiguity Reduction

As shown by the 3-inverter example MBD generates additional fail sets (conflicts) compared to SFL. Consequently, SFL gives rise to AGs that can be much larger than MBD. In software the AG problem can often be resolved by adding better tests (e.g., distinguishing components by introducing different control flow, unless they belong to the same basic block). In hardware, however, the AG problem is determined by circuit topology (which we assume static). As the ratio between the number of components and primary outputs is determined by area vs. circumference, the AG problem scales with the size of the circuit, leading to very large AGs. This implies that there are also large AGs in the ranking R (equal posteriors) if the p_j and g_j would be equal, which can greatly affect C_d. As p_j is typically not available often one assumes p_j to some arbitrary value p. Even when the p_j would be different, the diagnostic performance of SFL is still largely determined by the quality of g_j, as has been shown in the software domain [23]. The reason is that g_j is involved in the Bayesian update every time a new observation is processed. In software g_j is typically measured using mutation analysis [24]. While these measurements significantly increase SFL accuracy, the cost of a Monte-Carlo approach scales linearly with system size. In ANTARES we therefore also consider an *analytic* approach to the estimation of g_j since the estimation quality is sufficient to tackle the ambiguity problem.

Gate EPP Estimation

Computing the g_j of a component c_j can be framed as the problem of computing the error propagation probability (EPP) through a logic circuit [25], [26]. For example, consider the simple logic circuit according to Figure 3, comprising an INV gate (c_1) connected to an AND gate (c_2). Suppose

[3] Note that the total probability mass (M_f) may well exceed unity.

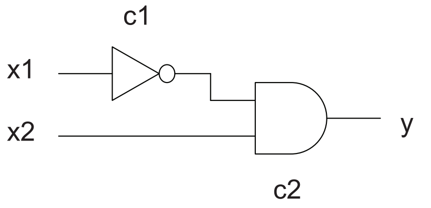

Figure 3. Example logic circuit

the INV gate is faulted. For input $x = (\mathrm{X}, 0)$ (X = don't care) an error at the output of c_1 will be masked by the fact that c_2 will always produce $y = 0$. However, for input $x = (X, 1)$ an inverter error will always propagate to y. Assuming a uniform input value probability distribution, the EPP at y is 0.5, and consequently, $g_1 = 1 - 0.5 = 0.5$.

EPP in logic circuits has been studied in the context of reliability studies, primarily motivated by an increasing soft error rate due to ever decreasing gate sizes [25], [27]. While there exists a deterministic approach to compute the EPP through circuits of arbitrary topology given the model of each gate involved, in this paper we use a novel, *probabilistic* approach to EPP computation, since in the ANTARES approach we refrain from modeling the actual components. While our method produces exact estimates of the *mean value* of the EPP over the corresponding gate space, the EPP value found for a particular circuit output may differ from the correct value. However, a certain error is acceptable provided the posterior probability *ranking* by our diagnosis algorithm is not too seriously affected.

Due to space limitation, we refrain from explaining in detail how the EPP is computed. For interested readers, refer to [28]. However, in this paper, we use the general EPP model for a binary gate as derived in [28]. Let e_1 and e_2 denote that the probability that inputs x_1 and x_2 of the binary gate have an error respectively. The EPP value (probability e that the gate's output has an error) is given by

$$\mathsf{E}[e] = \frac{e_1 + e_2 - e_1 \cdot e_2}{2} \tag{3}$$

Note that Eq. (3) is averaged over the space of all 16 conceivable binary gate functions. Computing the EPP at circuit level simply requires composition of Eq. (3) per gate between the faulted gate(s) and the circuit output.

FNR Estimation

The above EPP model allows us to directly compute $\mathsf{Pr}(obs_i|d_k)$ for multiple-fault candidates d_k in the Bayesian update (Eq. (1)), circumventing the approach based on the single-fault g_j parameters in combination with the OR-model (Eq. (2)). However, the decreasing accuracy with increasing cardinality makes the EPP model less attractive for multiple faults. Despite the fact that the OR-model assumes failure independence, its accuracy in practice outweighs the inaccuracy of a direct computation [28]. Consequently, in the following we outline an FNR estimation procedure based on the g_j parameters obtained through single-fault EPP modeling.

Obtaining the EPP for single faults is straightforward. In the following we assume that a gate is either SA0 (stuck at 0) or SA1 (stuck at 1), leading to an average failure probability of 0.5. As this gate is the only faulted gate in the circuit every subsequent gate downstream to the primary output has exactly one input that has an error. Consequently, Eq. (3) can be simplified. Without loss of generality, assume that for each gate in the path between faulted gate and primary output, x_1 is in the error path. Consequently, $e_2 = 0$ and Eq. (3) reduces to $\mathsf{E}[e] = e_1/2$, which implies halving the EPP per stage[4]. Let m_j denote the 'depth' of the gate c_j relative to the output considered (for the gate at the output $m_j = 0$). Assuming that the faulted gate produces $e = 1/2$ it follows that g_j is given by

$$g_j = 1 - 2^{-(m_j+1)} \tag{4}$$

which is substituted in Eq. (2). Thus, in this model the FNR for a component is only determined by the relative topological depth of the component relative to a particular circuit output, where all (intermediate) components are modeled by the generic model of Eq. (3).

4. EXPERIMENTAL RESULTS

In this section we evaluate the diagnostic performance of ANTARES for the circuits described earlier in comparison to an MBD approach. Tables 1, 2, and 3 list the diagnostic performance results for 50 random test vectors, averaged over 200 randomly injected fault sets with $M_f = 1, 2, 3$ faults, respectively. Instead of C_d we quote $C_d/(M - M_f)$ which allows comparison between different values of M_f (a value of 0 indicates no identification effort, i.e., all faulted components are ranked at the top, whereas a value of 1 indicates maximum identification effort, i.e., all faulted components are at the bottom of the list).

We consider three versions of ANTARES:

- A version, denoted $\mathrm{A_{BAR}}$, where g_j is *not* determined by topology, but is computed internally by BARINEL based on (A, O) [8]. This reference version [29] is intended to assess the added value of using topology-specific g_j information.
- A version, denoted $\mathrm{A_{MC}}$, where g_j is determined from the circuit using Monte Carlo (MC) simulation and is externally supplied to BARINEL. This version uses the most accurate g_j information.
- A version, denoted $\mathrm{A_{EPP}}$, where g_j is estimated from the circuit using the analytical EPP model and is externally supplied to BARINEL.

In order to compare ANTARES to MBD we include results for GDE, a state-of-the-art MBD engine [9]. Since GDE does not provide posterior probabilities we have incorporated GDE within our SFL approach as follows. The additional conflicts that GDE infers due to its MBD capability are appended as rows to the A matrix obtained by ANTARES, with $o_i = 1$. The extended (A, O) are processed as usual using STACCATO and BARINEL.

Comparison of the $\mathrm{A_{BAR}}$ and MBD results shows that some diagnostic accuracy is lost by merely taking into account structure (topology) with a standard, weak component model without FNR information. The results for $\mathrm{A_{MC}}$ show that knowledge of the g_j has a significant impact on ANTARES' diagnostic performance. Although the results are for logic

[4] As mentioned before, this differs for particular gates (1 for XOR, 1/2 for AND, etc.). Our EPP model represents the average over all 16 conceivable gates. One can improve the model once statistics are known regarding gate types.

Table 1. C_d for ANTARES and MBD ($M_f = 1$)

Circuit	$\mathrm{A_{BAR}}$	$\mathrm{A_{MC}}$	$\mathrm{A_{EPP}}$	MBD
74181	0.176	0.014	0.247	0.030
74182	0.090	0.061	0.080	0.025
74L85	0.320	0.105	0.250	0.060
74283	0.096	0.034	0.155	0.050
c499	0.164	0.092	0.129	0.004
c880	0.065	0.021	0.052	0.006
c1355	0.196	0.135	0.162	0.004
c2670	0.166	0.067	0.089	0.033
c7520	0.112	0.087	0.110	0.003

Table 2. C_d for ANTARES and MBD ($M_f = 2$)

Circuit	$\mathrm{A_{BAR}}$	$\mathrm{A_{MC}}$	$\mathrm{A_{EPP}}$	MBD
74181	0.314	0.230	0.502	0.110
74182	0.275	0.201	0.250	0.140
74L85	0.550	0.483	0.545	0.144
74283	0.303	0.210	0.355	0.210
c499	0.182	0.123	0.164	0.010
c880	0.461	0.265	0.386	0.011

circuits only, these results suggest that in quite a number of cases the modeling cost associated with MBD may well outweigh the limited loss of ANTARES's diagnostic utility.

The results for $\mathrm{A_{EPP}}$ show that analytically estimating the g_j using our generic component model does not always improve ANTARES' performance compared to not using them ($\mathrm{A_{BAR}}$). Given the impact of g_j as shown by $\mathrm{A_{MC}}$, however, there is a great potential in developing more elaborate, analytic schemes. An obvious extension of the analytic EPP model takes into account information on the truth probability of a gate (in terms of its truth table). For instance, the EPP characteristics of an AND (truth probability 1/4) and an OR (truth probability 3/4) are equal to Eq. (3), while an XOR (truth probability 1/2) has higher EPP (a single error on one of its inputs always propagates to the output). In some cases the truth probability of components may be known by design, or can be measured in isolation using MC simulation.

Single faults dominate the ranking yielded by random vectors since there are many single-fault diagnosis candidates that explain all failed observations. In other words, unlike MBD, our study does not make use of Max-Fault Min-Cardinality (MFMC) observation vectors, but instead use random vectors in an attempt to mimic reality. As a consequence, ANTARES suffers from a limitation in presence of multiple faults. Even for large circuits (c1355, c2670 and c7552) only one of the faulty components appears in the diagnosis and therefore ANTARES could not isolate all of them. Hence we do not include those circuit results in the paper for $M_f = 2$ and 3. However, in many real life scenarios, a diagnostician looking for the root cause of a system failure does not know in advance how many faults are in the system; every time a faulty component has been found and replaced, the system is typically re-tested, to ascertain that all faults have been found. With such iterative process, multiple faults often can be detected using the single fault diagnosis approach, making ANTARES a useful approach.

Table 3. C_d for ANTARES and MBD ($M_f = 3$)

Circuit	$\mathrm{A_{BAR}}$	$\mathrm{A_{MC}}$	$\mathrm{A_{EPP}}$	MBD
74181	0.480	0.460	0.608	0.287
74182	0.365	0.303	0.350	0.203
74L85	0.671	0.610	0.675	0.195
74283	0.447	0.375	0.516	0.410
c499	0.203	0.158	0.182	0.020
c880	0.502	0.321	0.452	0.020

5. CONCLUSION

Results clearly show that MBD outperforms every variant of ANTARES which demonstrates the importance of modeling information in diagnosis. However, there are situations where it is impossible to create behavioral models. For instance, in software of realistic size and complexity the choice not to model is typically borne out of necessity. Our industrial feedback suggests that there is a business proposition in sacrificing some diagnostic performance in an approach where modeling is no longer required.

In this paper we addressed the trade-off between the modeling/identification costs in diagnosis. We also propose to exploit FNR information to boost the diagnosis quality without actually using the behavioral models. Our results show that ANTARES using detailed FNR information is capable of approaching the performance of MBD. While in software mutation analysis is relatively easy to implement, measuring FNR data in hardware can only be done when simulators are available. Consequently, we also studied a simple, abstract EPP modeling technique to analytically estimate the FNR data. Our results show that a more detailed EPP model is required to attain the performance of Monte Carlo measurements.

Future work will address improved EPP modeling to further exploit the potential of ANTARES. Currently, we use a generic component EPP model, and we will investigate whether we can reach the quality of the Monte Carlo approach by dynamically measuring each component's EPP. The great significance of such *empirical* study is to measure EPP without needing to inject faults in the circuit, while still not modeling component's behavior.

ACKNOWLEDGMENTS

This work was partially funded by FEDER/ON2 and FCT project NORTE-07-124-FEDER-000062.

REFERENCES

[1] J. Pietersma and A. van Gemund, "Benefits and costs of model-based fault diagnosis for semiconductor manufacturing equipment," in *Proc. INCOSE'07*, 2007.

[2] B. Reeven, "Model-based diagnosis in industrial context," 2011, mSc thesis, Delft University of Technology,

The Netherlands.

[3] E. Schoemaker, ASML Netherlands, Personal Communication, 2008.

[4] M. Nica, J. Weber, and F. Wotawa, “How to debug sequential code by means of constraint representation,” in *Proc. of DX’08*, September 2008.

[5] R. Mathijssen, Embedded Systems Institute, Personal Communication, 2008.

[6] P. Zoeteweij, R. Abreu, R. Golsteijn, and A. van Gemund, “Diagnosis of embedded software using program spectra,” in *Proc. of ECBS’07*, 2007.

[7] P. Zoeteweij, J. Pietersma, R. Abreu, A. Feldman, and A. van Gemund, “Automated fault diagnosis in embedded systems,” in *Proc. of SSIRI’08*, 2008.

[8] R. Abreu and A. van Gemund, “Diagnosing intermittent faults using maximum likelihood estimation,” *Artificial Intelligence Journal*, 2010.

[9] J. de Kleer, “Minimum cardinality candidate generation,” in *Proc. of DX’09*, 2009.

[10] R. Bakker, D. van Soest, P. Hogenhuis, and N. Mars, “Fault models in structural diagnosis,” in *Proc. of DX’89*, 1989.

[11] A. Ducoli, G. Lamperti, E. Piantoni, and M. Zanella, “Circular pruning for lazy diagnosis of active systems,” in *Proc. of DX’09*, 2009.

[12] L. Gianfranco and M. Zanella, “Distributed consistency-based diagnosis without behavior,” in *Proc. of DX’10*, October 2010.

[13] M. Staroswiecki and G. Comtet-Varga, “Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems,” *Automatica*, vol. 37, no. 5, pp. 687–699, May 2001. [Online]. Available: http://dx.doi.org/10.1016/S0005-1098(01)00005-X

[14] A. Gonzalez-Sanchez, R. Abreu, H.-G. Gross, and A. van Gemund, “Spectrum-based sequential diagnosis,” in *Proc. of DX’10*, 2010.

[15] X. Wang, S. Cheung, W. Chan, and Z. Zhang, “Taming coincidental correctness: Coverage refinement with context patterns to improve fault localization,” in *Proc. of ICSE’09*, 2009.

[16] G. Rothermel, R. Untch, C. Chu, and M. Harrold, “Prioritizing test cases for regression testing,” *IEEE TSE*, 2001.

[17] J. de Kleer, “Diagnosing multiple persistent and intermittent faults,” in *Proc. of DX’07*, 2007.

[18] R. Abreu, P. Zoeteweij, R. Golsteijn, and A. van Gemund, “A practical evaluation of spectrum-based fault localization,” *Journal of Systems and Software*, 2009.

[19] A. Feldman, T. Kurtoglu, S. Narasimhan, S. Poll, D. Garcia, J. de Kleer, L. Kuhn, and A. van Gemund, “Empirical evaluation of diagnostic algorithm performance using a generic framework,” *International Journal of Prognostics and Health Management*, 2010.

[20] L. Kuhn, B. Price, J. de Kleer, M. Do, and R. Zhou, “Pervasive diagnosis: Integration of active diagnosis into production plans,” in *Proc. AAAI’08*, 2008.

[21] R. Abreu and A. G. Gemund, “A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis,” in *Proceedings of the 8th Symposium on Abstraction, Reformulation, and Approximation*, ser. SARA’09, 2009.

[22] A. Feldman, G. Provan, and A. van Gemund, “Computing observation vectors for max-fault min-cardinality diagnoses,” in *Proc. AAAI’08*, 2008.

[23] A. Gonzalez-Sanchez, R. Abreu, H.-G. Gross, and A. van Gemund, “An empirical study on the usage of testability information to fault localization in software,” in *Proc. SAC’11*, 2011.

[24] J. Voas, “Pie: A dynamic failure-based technique,” *IEEE TSE*, 1992.

[25] H. Asadi, M. Tahoori, and C. Tirumurti, “Estimating error propagation probabilities with bounded variance,” in *Proc. of DFT’07*, 2007.

[26] N. Mohyuddin, E. Pakbaznia, and M. Pedram, “Probabilistic error propagation in logic circuits using the boolean difference calculus,” in *Proc. of ICCD’08*, 2008.

[27] K. Parker and E. McCluskey, “Probabilistic treatment of general combinational networks,” *IEEE Trans. Computers*, 1975.

[28] S. Gupta, A. J. C. van Gemund, and R. Abreu, “Probabilistic error propagation modeling in logic circuits,” in *Proceedings ICST’11 Workshops*, ser. ICSTW ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 617–623. [Online]. Available: http://dx.doi.org/10.1109/ICSTW.2011.40

[29] M. Wilson, “BACINOL: Bayesian circuit analysis by topology,” 2011, mSc thesis, Delft University of Technology, The Netherlands.

Biography

Shekhar Gupta *Shekhar Gupta is a 3rd year PhD student at Delft University of Technology (TUDelft), The Netherlands. He is on an extended research leave at Palo Alto Research Center (PARC) in Palo Alto, CA, USA. He is working in the field of Model Based Diagnosis (MBD) with Dr. Johan DeKleer. Earlier he has pursued MSc in Computer Engineering from TUDelft in 2011 with Cum Laude. He did his Master thesis with Prof. Arjan. J. C. Van Gemund on embedded system fault diagnosis. He has pursued bachelor degree (BTech) in Information and Communication Technology (ICT) from Dhirubhai Ambani Institute of Information and Communication Technology (DAIICT), Gujarat India.*

Rui Abreu *graduated in Systems and Computer Engineering from University of Minho, Portugal, carrying out his graduation thesis project at Siemens S.A., Portugal. Between September 2002 and February 2003, Rui followed courses of the Software Technology Master Course at University of Utrecht, the Netherlands, as an Erasmus Exchage Student. He was an intern researcher at Philips Research Labs, the Netherlands, between October 2004 and June 2005. He received his Ph.D. degree from the Delft University of Technology, the Netherlands, in November 2009, and he is currently an assistant professor at the Faculty* *of Engineering of University of Porto, Portugal. He is also with the School of Computer Science of Carnegie Mellon University (CMU), USA, as a Visiting Faculty Member.*

Johan de Kleer *is a Principal Scientist in the Embedded Reasoning Area in PARC's Intelligent Systems Laboratory. His core interest is building a system which can reason about the physical world as well as he can. Until recently, he was Laboratory Manager of PARC's Systems and Practices Laboratory of Xerox's Palo Alto Research Center. This interdisciplinary laboratory conducted research ranging from social science to robotics. Two foci of the laboratory were: (1) Smart Matter - which exploits trends in miniaturization and integration to create a new generation of products and processes that benefit from the coupling of computational and physical worlds, and (2) Knowledge - knowledge management, which includes social science research on organizations, knowledge representation and understanding images and video streams. Johan received his Ph.D. from M.I.T. in 1979 in Artificial Intelligence. He has published widely on Qualitative Physics, Model-Based Reasoning, Truth Maintenance Systems, and Knowledge Representation. He has co-authored three books: Readings in Qualitative Physics, Readings in Model-Based Diagnosis, Building Problem Solvers. In 1987 he received the prestigious Computers and Thought Award at the International Joint Conference on Artificial Intelligence. He is a fellow of the American Association of Artificial Intelligence and the Association of Computing Machinery.*

Arjan J.C. van Gemund *received a BSc in Physics, an MSc degree (cum laude) in Computer Science, and a PhD (cum laude), all from Delft University of Technology. He has held positions at the R & D organization of the Dutch multinational company DSM as an Embedded Systems Engineer, and at the Dutch TNO Research Organization as a High-Performance Computing Research Scientist. From 1992 he was at the Electrical Engineering, Mathematics, and Computer Science Faculty of Delft University of Technology, the last 6 years serving as Full Professor in the area of fault diagnosis of embedded hardware and software systems. He has (co)authored over 200 scientific publications, and is (co)recipient of 8 best paper awards, and a best demo award.*