Fourth International Diagnostic Competition – DXC'13

Adam Sweet[1], **Alexander Feldman**[2], **Sriram Narasimhan**[3], **Matthew Daigle**[1], and **Scott Poll**[1]
[1]NASA Ames Research Center, Moffett Field, CA, U.S.A.
email: {adam.sweet, matthew.j.daigle, scott.poll}@nasa.gov
[2]University College Cork, Cork, Ireland
email: alex@llama.gs
[3]University of California, Santa Cruz @ NASA Ames Research Center, Moffett Field, CA, U.S.A.
email: sriram.narasimhan-1@nasa.gov

Abstract

We present the description and results of the Fourth International Diagnostic Competition, which tests and evaluates diagnostic algorithms (DAs). This year's competition offered the industrial, synthetic and software tracks from previous competitions, and a new thermal-fluid track. Only the industrial track competition was held, with a total of 5 DAs participating. The paper briefly reviews the industrial track used in this and previous competitions. The participating DAs are described, and the scoring metrics and competition results are presented.

1 Introduction

Much research has been done in the field of diagnosis, resulting in many types of algorithms capable of detecting and isolating faults in many types of systems. However, until recently there have been few efforts to evaluate and compare these algorithms in a standard way. NASA Ames Research Center (ARC), Palo Alto Research Center (PARC), and Delft University of Technology decided to combine efforts to create a generalized framework that would establish a common platform to evaluate and compare diagnosis algorithms (Kurtoglu et al., 2009a). The objectives for developing this framework were to accelerate research in theories, principles, and computational techniques for monitoring and diagnosis of complex systems; to encourage the development of software platforms that promise more rapid, accessible, and effective maturation of diagnostic technologies; and to provide a forum that can be utilized by algorithm developers to test and validate their technologies on real-world physical systems.

A series of competitions has been held using the framework to test and evaluate diagnostic algorithms (DAs) in a variety of diagnostic problem domains. The First International Diagnostic Competition (DXC'09) was held under the auspices of the DX conference, and the methods and results of the competition are presented in (Kurtoglu et al., 2009b). In that competition, two tracks were defined to present diagnostic problems in different domains: an industrial track focusing on an electrical power system and a synthetic track focusing on logic circuits. A set of metrics was created to quantitatively evaluate the diagnostic algorithms, and the metrics were weighted to determine the overall winner. However, this approach has the weakness that a DA's score in the competition is heavily dependent on the weights assigned to each metric. In practice, the importance of different diagnostic metrics depends on the requirements of the application.

The scoring method was changed for the Second International Diagnostic Competition (DXC'10), (Poll *et al.*, 2010). Instead of the DXC'09 scoring with the same weights on the metrics for all tracks, the scores for each track were determined according to a use case defined for that track. The industrial track used a decision support use case, where the diagnosis would be used to determine a recovery action. The score for a DA depends on the correctness of the recommended recovery action. The synthetic track used a troubleshooting use case, where many internal variables were not observable and probes must be used to determine an unambiguous diagnosis. The goal was to correctly identify the fault with the fewest probes. The metrics were still calculated and used as tiebreakers.

A Third International Diagnostic Competition (DXC'11) was held, and introduced a new track on software diagnosis (Poll *et al.*, 2011). The goal of this track is to provide common ground to evaluate techniques that diagnose failures in software systems. For DXC'11, the focus was on techniques that use coverage data; the algorithms were evaluated on their performance in finding software faults based only on that coverage data.

Finally, the present Fourth International Diagnostic Competition (DXC'13) was held. The most significant addition to this competition was the new thermal fluid track, which presented problems in a building's heating, ventilation, and air conditioning (HVAC) domain. The tracks from previous years were also available. The industrial track's electrical power system competition was performed, with the same format as in previous competitions and with newly acquired competition data.

The paper is organized as follows. Section 2 gives a quick review of the DXC framework. Section 3 describes the diagnostic problems that were presented to the competitors. Section 4 lists the kinds of faults that were injected. Section 5 explains how the evaluation was performed. Section 6 presents the results. Section 7 concludes the paper.

2 DXC Framework

The DXC framework was developed for DXC'09 and modified in subsequent years. It allows DAs to be tested under identical experimental conditions and saves and evaluates the result of the tests. The key components of this framework include representation languages for the physical system description, sensor data and diagnosis results, a run-time architecture for executing DAs and diagnostic scenarios, and an evaluation component that computes performance metrics based on the results from diagnosis algorithm execution.

The DXC framework has been extensively described in past publications; the reader may refer to (Kurtoglu *et al.*, 2009a; Feldman *et al.*, 2010; and Poll *et al.*, 2011) for those descriptions and architecture diagrams. A textual description of the main run-time components of the framework is repeated here:

Scenario Loader (SL): Executes the Scenario Data Source, Recorder, and Diagnosis Algorithm. SL ensures system stability and clean-up upon scenario completion. This is the main entry point for performing a diagnostic experiment.

Scenario Data Source (SDS): Provides scenario data from previously recorded datasets. The provenance of the data (whether hardware or simulation) depends on the system in question. A scenario dataset contains sensor readings, commands (note that the majority of classical model-based diagnosis literature does not distinguish commands from observations), and fault injection information (to be sent exclusively to the Scenario Recorder). SDS publishes data following a wall-clock schedule specified by timestamps in the scenario files.

Scenario Recorder (SR): Receives fault injection data and diagnosis data into a scenario results file. The results file contains a number of time-series which are used by the evaluation module for the computation of metrics. SR is the main timing authority, i.e., it timestamps each message upon arrival before recording it to the results file.

Diagnosis Algorithm (DA): A DA receives sensor and command data, performs diagnosis, and sends the diagnosis results back. As long as the DAs comply with the provided API, there are no restrictions on a DA; for example, a DA may read precompiled data, or use external (user supplied) libraries, etc.

Diagnostic Oracle: The Diagnostic Oracle is only relevant to the Industrial Track. It provides a querying capability to the DAs in one of two ways: 1) it takes a diagnostic output produced by a DA and returns the lowest cost action(s) associated with the provided diagnosis, or 2) it takes a diagnostic output and specific actions produced by a DA and returns the corresponding cost.

Evaluator: Takes scenario result file and applies metrics to evaluate DA performance. The metrics and evaluation procedures are detailed in Section 5.

3 Diagnostic Problems

Five diagnostic problems were announced for DXC'13: two industrial track problems (DP-I, DP-II), one synthetic (DP-III), one software (DP-IV), and the new thermal-fluid track problem (DP-V). Unfortunately, the only entries received were in DP-I. The other competition tracks will be maintained and hopefully used in future competitions.

The system used for DP-I, ADAPT-Lite, is based on the Electrical Power System (EPS) testbed in the ADAPT lab located at NASA Ames Research Center (Poll *et al.*, 2007). This system and the DP-I problem are described in detail in previous publications (Kurtoglu *et al.*, 2009a; Feldman *et al.*, 2010; and Poll *et al.*, 2011). It has not been changed for DXC'13. A brief summary is presented in this section.

A subset of the components and sensors on the ADAPT EPS is used to mimic the operation of an electrical system aboard a single-string Unmanned Aircraft System (UAS). DP-I does not represent any particular UAS, but may be thought of as a generic UAS carrying instruments that acquire scientific data. A system schematic for DP-I is given in Figure 1. BAT2 is a battery supplying electrical power to several loads in the UAS system. The power is transmitted through several circuit breakers (with component names beginning in "CB") and relays ("EY"), and an inverter INV2 to supply AC power. The loads are named AC483, FAN416, and DC485. There are also sensors throughout the system to report electrical voltage (names beginning with "E"), electrical current ("IT"), and the positions of relays and circuit breakers ("ESH", "ISH"). Finally there is one sensor to report the operating state of a load (fan speed, "ST").

The DA is used for decision-support during the UAS mission to inform the pilot if faults have occurred, and if so, whether the fault requires aborting the mission and landing the UAS. The necessity of an abort depends on the fault present, and in some cases, on the values of the fault parameters. Any failure which cuts off power to one of the three loads requires an abort. The other failures result in a degraded operation, and while some do not require an abort, others may, depending on the fault parameters. Thus, this diagnostic problem requires the DAs to perform fault detection, isolation, and parameter estimation. In the context of the DXC, the DA will be scored according to the correctness of its abort (or no-

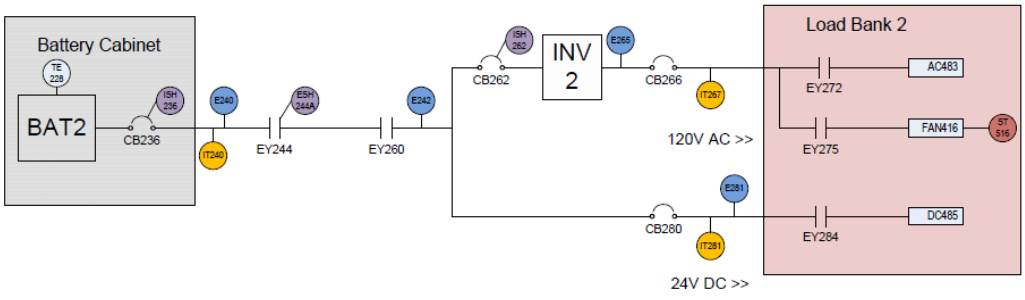

Figure 1: ADAPT-Lite system for DP-I

abort) recommendation as described in later sections.

Finally, with the given sensors, DP-I contains four diagnostic ambiguity groups: (i) AC483 failed off and EY272 stuck open, (ii) FAN416 failed off and EY275 stuck open, (iii) DC485 failed off and EY284 stuck open, and (iv) INV2 failed off and CB262 failed open. In each case however, the recovery action is the same for both faults in the ambiguity group.

4 Fault Injection and Scenarios

The DP-I scenarios are a series of approximately four-minute scenarios gathered from the ADAPT EPS. DP-I scenarios only contain a single injected fault, but DAs may report multiple faults as part of an ambiguity group (especially for the ambiguities listed in section 3.1).

The ADAPT EPS is designed to allow repeatable fault injection, in one of several ways. The first method is directly in the hardware, by manually switching components off or on or manipulating the load resistances. The second method is in software, by intercepting user commands or sending extraneous commands to the EPS, both unbeknownst to the DA. The third method is done with postprocessing of a nominal data run; this method is useful for injecting sensor faults, as they don't affect other components of the system.

For DXC'13, 114 new scenarios were gathered from the ADAPT EPS and used for the competition scenarios. All previous training and competition scenarios were available to entrants to use as training scenarios.

5 Evaluation

This section describes the scoring and the computation resources used in this year's competition.

5.1 Scoring

As described in Section 1, the entries in the DP-I diagnostic problem will be evaluated on the basis of the correctness of their abort recommendations. A cost is assigned to each DA's abort or no-abort recommendation for each scenario, and the total cost for all scenarios is summed to determine the DA's final score. The DA with the lowest cost is the competition winner. The costs are summarized in Table 1. As shown in the table, a correct abort recommendation is given a cost of 0, and an incorrect abort recommendation's cost depends on what should have been recommended for that scenario. If a scenario contains a fault which does not require an abort, but the DA recommends an abort, it is regarded as a loss of mission and given the cost $C_{mission}$. If a scenario contains a fault requiring abort but the DA did not recommend abort, it is regarded as a loss of vehicle, $C_{vehicle,}$ and the mission $C_{mission}$, which is a much higher cost. For DP-I, $C_{mission} = 25$, and $C_{vehicle} = 100$. A perfect-scoring DA will thus have an overall competition cost of 0.

Table 1: DP-I Decision Costs ($\mathbf{M}_{dc}$)

Actual Case / DA Rec.	Abort	Non-abort
Abort	0	$C_{mission}$
Non-abort	$C_{mission}$ + $C_{vehicle}$	0

The metrics used in DXC'09 will also be gathered and calculated for purposes of tie-breaking and comparison. Please see (Kurtoglu et al., 2009; Feldman et al., 2010) for detailed definitions and related discussion. These metrics are summarized in Table 2. Note that DXC'09 metric $\mathbf{M}_{ia}$ has been renamed $\mathbf{M}_{err}$ in the table. The metrics in the table are per scenario metrics. To calculate "per system" metrics an unweighted average is taken over all scenarios and is indicated with an overbar.

5.2 Computing platform

DP-I diagnosis algorithms were evaluated using the DXC framework on a Windows 7 computer with an Intel i7-3770S CPU running at 3.10 GHz.

6 Results

This section describes the entrants into the competition and presents the competition results.

6.1 Diagnosis Algorithms

A description of each DA entered in DXC'13 is given below:

1. QED: A model-based diagnosis system based on qualitative event-based fault isolation. Statistically significant deviations of measured from model-predicted values imply the presence of faults. These deviations are abstracted into symbolic event-based descriptions of fault-induced behavior, which are compared to predicted event sequences to isolate faults. Fault identification uses quantitative methods to compute fault parameters and further refine fault hypotheses (Daigle and Roychoudhury, 2010).

Table 2: DXC'09 Metrics

Metric	Name / description
$\bar{\mathbf{M}}_{fd}$	Fault detection time, average
$\bar{\mathbf{M}}_{fn}$	False negative rate, percentage of total scenarios
$\bar{\mathbf{M}}_{fp}$	False positive rate, percentage of total scenarios
$\bar{\mathbf{M}}_{da}$	Detection accuracy, percentage of total scenarios
$\bar{\mathbf{M}}_{fi}$	Fault isolation time, average
$\mathbf{M}_{err}$	Classification errors, sum of all scenarios
$\bar{\mathbf{M}}_{cpu}$	Computer processing time used, average
$\bar{\mathbf{M}}_{mem}$	Computer memory used, average

2. QED-PC: Similar to QED, but uses the Possible Conflicts diagnosis approach (Pulido and Alonso-Gonzalez, 2004). The global system model is de-composed into minimal submodels containing a sufficient analytical redundancy to generate fault hypothesis from observed measurement deviations. (Daigle *et al.*, 2011).
3. QED-PC++: Combines the residual sets of QED and QED-PC to improve fault isolation over each algorithm individually. Residuals from both the global model and PCs are used for fault isolation within the qualitative fault isolation framework. This improves diagnosability and fault isolation times.
4. HyDE: Hybrid Diagnosis Engine (HyDE) is a model based diagnosis engine that uses consistency between model predictions and observations to generate conflicts which in turn drive the search for new fault candidates. The model is used to simulate system behavior which is compared actual system behavior to identify any discrepancies. The discrepancies are used to guide the isolation of possible fault faults that would make simulated and actual system behavior consistent.
5. HyDE-PC: Similar to HyDE, but also uses the Possible Conflicts approach (Pulido and Alonso-Gonzalez, 2004).

6.2 Results

The metrics for all 5 DAs in this year's competition are summarized in Table 3. The DAs are listed in the table in order of their ranking for the competition, based on the decision cost ($\mathbf{M}_{dc}$) metric.

QED was the winner of the DP-1 competition, with a total cost of 250. This is due to missing 2 aborts. The first missed abort seems to be due to a software glitch. QED produced a correct diagnosis but for some reason did not query the oracle or send a recovery action. The same fault is repeated at a different time in another scenario, and in that other scenario QED does query the oracle and issue the correct recovery action to abort. For the second, the injected fault is quite close to the abort vs. no-abort threshold. While QED's determination of the fault parameters was quite close to the exact values used in the fault injection, those values indicate to not abort. The exact values of the injected fault parameters do count as an abort being the correct action for that scenario.

Some other interesting aspects of the results are seen in other metrics. QED had the best overall diagnosis results, with lower false positive ($\bar{\mathbf{M}}_{fp}$) and higher diagnostic accuracy ($\bar{\mathbf{M}}_{da}$) rates than QED-PC and QED-PC++. This is most likely a result of the maturity of the algorithm: QED has been entered in all of the past competitions, while QED-PC and QED-PC++ are similar but newer. Also, QED-PC++ has much lower fault detection times and fault isolation times than QED or QED-PC. This was the expected behavior, borne out by experimental verification in DXC'13.

The HyDE and HyDE-PC DAs were also not as mature as QED, although HyDE was a previous entrant it has not been entered in every competition. Both variants of HyDE incurred higher decision costs than the variants of QED. Looking at Table 3, the HyDE variants had a bias toward false negatives ($\bar{\mathbf{M}}_{fn}$) rather than false positives ($\bar{\mathbf{M}}_{fp}$). Hence, HyDE was biased toward not commanding an abort. This unfortunately is not a good strategy given the scoring for DP-I; far more cost is assigned to incorrectly losing a vehicle (by failing to abort) than

Table 3: DP-I Competition Results

DA	$\mathbf{M}_{dc}$	$\bar{\mathbf{M}}_{fd}$ (s)	$\bar{\mathbf{M}}_{fn}$	$\bar{\mathbf{M}}_{fp}$	$\bar{\mathbf{M}}_{da}$	$\bar{\mathbf{M}}_{fi}$ (s)	$\mathbf{M}_{err}$	$\bar{\mathbf{M}}_{cpu}$ (ms)	$\bar{\mathbf{M}}_{mem}$ (kb)
QED	250	3.255	0.0	0.035	0.9649	71.861	25	10.8	7504
QED-PC++	1350	1.657	0.0	0.439	0.5614	43.837	66	13.5	7847
QED-PC	1500	3.133	0.0	0.406	0.5965	67.146	85	9.9	7687
HyDE-PC	2550	8.157	0.20	0.018	0.807	8.316	112	430	59279
HyDE	2650	8.769	0.19	0.018	0.8158	8.857	121	1311	83189

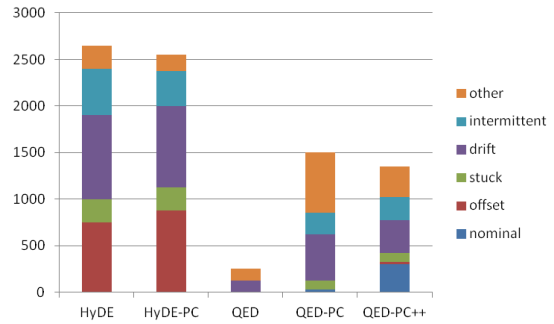

Figure 2: DP-I Cost by scenario type

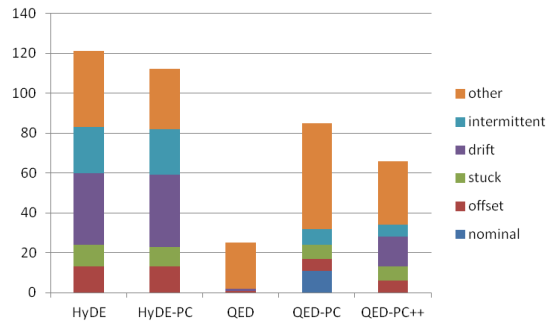

Figure 3: DP-I classification errors by scenario type

to incorrectly losing a mission (by aborting when unnecessary). The HyDE variants had the lowest false positive rates of all the entrants; if the HyDE variants were tuned to report fewer false negatives (even at a cost of increased false positives) it is likely their scores would have been similar to QED-PC and QED-PC++.

We show the breakdown of decision cost ($\mathbf{M}_{dc}$) by fault type for each DA in Figure 2. Offset, drift, and intermittent faults include hardware (AC483, DC485) and sensor (e.g., IT267, IT281, etc.) fault injection scenarios. Category "other" includes circuit breaker, inverter, fan, and AC and DC load failed-off fault scenarios. The QED variants are all good at detecting the "offset" type of fault, as Figure 2 shows: a very small slice (or no slice) is red indicating the "offset" type. The HyDE variants are fairly uniform in their ability to detect different types of faults, and had a considerably higher overall cost due to the immaturity of the diagnoser.

We also show the breakdown of classification errors ($\mathbf{M}_{err}$) by fault type in Figure 3. In a scenario, the number of classification errors is the number of misclassified components. Ruling out guessing, a perfect DA would have 23 classification errors, all in the category "other", because of the ambiguity groups.

7 Conclusion

We presented the implementation of the Fourth International Diagnostic Competition, DXC'13. This year's competition featured 5 DAs each competing in the industrial track problem DP-1. The winner had an excellent score, showing the maturity of the algorithm having been developed and entered in several competitions.

The authors hope that future competitions will garner more interest than this year's competition. The authors also hope that the DXC framework and data are useful to the research community outside of the competition, as a standard means to compare and benchmark diagnosis algorithms. Finally, we continue to hope that the framework is applied to more physical systems and diagnostic algorithms in the future.

Acknowledgments

As the competition and framework have been developed over many years, the authors would like to acknowledge the contributions of all previous DXC organizers and developers of the DXC framework, in particular Tolga Kurtoglu (PARC), David Garcia (PARC), and Johan De Kleer (PARC). We would also like to thank Amarnath Raveendran (Georgia Tech) and David Nishikawa (NASA) for their efforts in collecting new competition data for DP-I.

References

[Abreu et al. 2009] R. Abreu, P. Zoeteweij, A.J.C. van Gemund. A New Bayesian Approach to Multiple Intermittent Fault Diagnosis. In Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI'09), pp. 653-658, Pasadena, CA, USA, July 2009.

[Abreu et al., 2011] R. Abreu, P. Zoeteweij, A.J.C. van Gemund. Simultaneous Debugging of Software Faults. In Journal of Systems and Software (JSS), vol. 84(4), pp. 573-586, Elsevier, 2011.

[Daigle and Roychoudhury, 2010] M. Daigle and I. Roychoudhury. Qualitative Event-based Diagnosis: Case Study on the Second International Diagnostic Competition. In Proceedings of 21st International Workshop on Principles of Diagnosis, Portland, OR, 2010.

[Daigle et al., 2011] M. Daigle, A. Bregon, and I. Roychoudhury. Qualitative Event-based with Possible Conflicts: Case Study on the Third International Diagnostic Competition. In Proceedings of 22nd International Workshop on Principles of Diagnosis, Munich, Germany, 2011.

[de Kleer and Williams, 1987] J. de Kleer and B. C. Williams. Diagnosing Multiple Faults. Artificial Intelligence, 32(1):97-130, 1987.

[de Kleer, 2008] J. de Kleer. An Improved Approach for Generating Max-Fault Min-Cardinality Diagnoses. In Proceedings of 19th International Workshop on Principles of Diagnosis, Blue Mountains, Australia, 2008.

[de Kleer, 2011] J. de Kleer. Hitting Set Algorithms for Model-based Diagnosis. In Proceedings of 22nd International Workshop on Principles of Diagnosis, Munich, Germany, 2011.

[Feldman et al., 2008] A. Feldman, G. Provan, A. van Gemund. Computing observation vectors for Max-Fault Min-Cardinality diagnoses. In Proc. AAAI'08, pp. 919–924.

[Feldman et al., 2010] A. Feldman, T. Kurtoglu, S. Narasimhan, S. Poll, D. Garcia, J. de Kleer, L. Kuhn, A. van Gemund. Empirical Evaluation of Diagnostic Algorithm Performance Using a Generic Framework. In International Journal of Prognostics and Health Management, Vol. 1 (2), 2010.

[Gonzalez-Sanchez et al., 2011] A. Gonzalez-Sanchez, R, Abreu, H.G. Gross, A.J.C. van Gemund. Prioritizing Tests for Fault Localization through Ambiguity Group Reduction. In Proceedings of the 26th International Conference on Automated Software Engineering (ASE'11). Lawrence, KA, November 2011.

[Kurtoglu et al., 2009a] T. Kurtoglu, S. Narasimhan, S. Poll, D. Garcia, L. Kuhn, J. de Kleer, A. van Gemund, and A. Feldman. Towards a Framework for Evaluating and Comparing Diagnosis Algorithms. In Proceedings of 20th International Workshop on Principles of Diagnosis, Stockholm, Sweden, 2009.

[Kurtoglu et al., 2009b] T. Kurtoglu, S. Narasimhan, S. Poll, D. Garcia, L. Kuhn, J. de Kleer, A. van Gemund, A. Feldman. First International Diagnosis Competition – DXC'09. In Proceedings of 20th International Workshop on Principles of Diagnosis, Stockholm, Sweden, 2009.

[Mange et al., 2011] J. Mange, D Daniszewski, and A. Dunn. Artificial Immune Systems for Diagnostic

Classification Problems. In Proceedings of 21st International Workshop of Principles of Diagnosis, Munich, Germany, 2011.

[Mosterman and Biswas, 1999] P. J. Mosterman and G. Biswas. Diagnosis of Continuous Valued Systems in Transient Operating Regions. In IEEE Trans. on Systems, Man and Cybernetics, vol. 29, no. 6, pp. 554-565, Nov. 1999.

[Narasimhan and Brownston, 2007] S. Narasimhan and Lee Brownston. HyDE – A General Framework for Stochastic and Hybrid Model-based Diagnosis. In Proceedings of 18th International Workshop on Principles of Diagnosis, Nashville, TN, 2007.

[Poll et al., 2010] S. Poll, J. de Kleer, A. Feldman, D. Garcia, T. Kurtoglu, and S. Narasimhan. Second International Diagnostic Competition – DXC'10. In Proceedings of 21st International Workshop on Principles of Diagnosis, Portland, OR, 2010.

[Poll et al., 2011] S. Poll, J. de Kleer, R. Abreu, M. Daigle, A. Feldman, D. Garcia, A. Gonzalez-Sanchez, T. Kurtoglu, S. Narasimhan, and A. Sweet. Third International Diagnostic Competition – DXC'11. In Proceedings of 22st International Workshop on Principles of Diagnosis, Murnau, Germany, 2011.

[Pulido and Alonso-Gonzalez, 2004] B. Pulido and C. Alonso-Gonzalez. Possible conflicts: a compilation technique for consistency-based diagnosis. IEEE Trans. Syst. Man Cy. B., 34(5):2192–2206, 2004.

[Siddiqi and Huang, 2007]. S. Siddiqi and J. Huang. Hierarchical Diagnosis of Multiple Faults. In Proc. IJCAI'07, pp. 581–586.