The Diagnostic Competitions

*Alexander Feldman, Johan de Kleer, Tolga Kurtoglu,
Sriram Narasimhan, Scott Poll, David Garcia,
Lukas Kuhn, Arjan J. C. van Gemund*

■ *The international diagnostic competitions provide a set of diagnostic benchmarks to evaluate diagnostic algorithms. This article describes a common diagnostic framework used to evaluate these algorithms. These competitions, started in 2009, have significantly helped shape subsequent diagnostic algorithms.*

A fault in a system is a change in the system that results in it no longer achieving the functionality for which it was originally intended. Diagnostic algorithms (DAs) (1) detect malfunctioning systems, (2) isolate the faulty component or components that cause the malfunction, and possibly (3) repair the system to restore its functionality. The fundamental challenge of diagnosis is that the system is only partially observable. Therefore, diagnostic algorithms must reason backwards from symptoms to causes. For example, determining that a dead battery is the cause of your car not starting in the morning (and not the wiring or the ignition switch). The domains of diagnostic algorithms includes analog and digital circuits, software systems, thermal systems, biological systems, and physical mechanisms. The same classes of diagnostic algorithms can apply in all domains. Diagnostic algorithms make observations, often in real time, of a system being diagnosed. It is impractical to evaluate diagnostic algorithms against physical devices, so an important component of all our benchmarks is a fault simulator that can interact with the diagnostic as if it were the real world.

ISSN 0738-4602

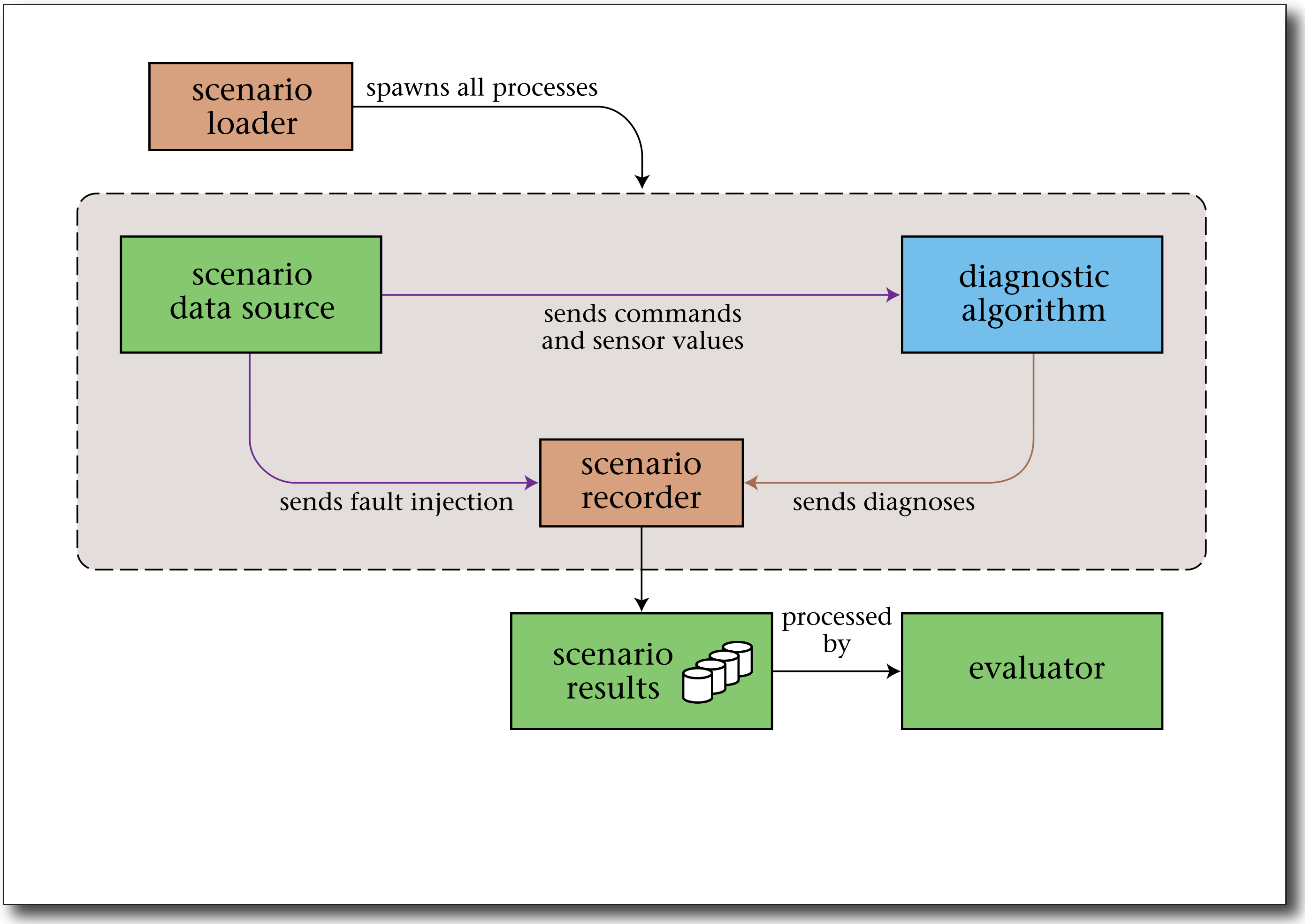

Figure 1. Basic Framework Architecture.

History

The International Diagnostic Competition (DXC) is part of the International Workshop on Principles Diagnosis (DX). The first diagnostic competition was held in 2009 and the results reported at the 20th DX workshop held in Stockholm, Sweden. At this first competition 13 diagnostic algorithms were submitted. At each workshop implementers of DX algorithms are encouraged to present their algorithms and benchmark results. The results of the first competition are described at length by Feldman et al. (2010).

Framework

One of the substantial achievements of the Diagnostic Competition is the design and implementation of the diagnostic framework (see figure 1). The diagnostic framework has been designed to mimic natural conditions for the implementations of the diagnostic algorithms. All decision making and communication, for example, is in real time.

Participants provide their diagnostic algorithm. The remainder of the framework is provided by the competition organizers. The competition benchmarks contains scenarios each of which consists of injected faults, inputs, and outputs. The inputs and outputs are provided to the DA (the faults themselves remain hidden to the DA). The results of the DA are compared to the actually injected faults and, based on that, several performance metrics are computed. In the case of the synthetic track there may be thousands of different fault/system combinations. Some tracks require the DA to actively measure new quantities or modify the system (this is not captured in figure 1).

The diagnostic framework is a collection of software components, libraries, programming interfaces, and protocols that allow the execution of diagnostic scenarios. Further, the diagnostic framework collects performance data about the diagnostic algorithms and computes performance metrics that are combined into a final ranking. The diagnostic framework specifies a text-based communication protocol consisting of time-stamped events. Its implementation is on top of the transmission control protocol and supports Linux and Windows.

Participants in the Diagnostic Competition submit diagnostic algorithms that implement the framework API. A diagnostic algorithm must implement a callback function that is called when new sensor data is available. We have provided dummy diagnostic algorithms written in C and in Java. For other languages, participants have to write their own code that communicates with the remaining framework components. Participants also submit diagnostic algorithms written in Matlab and in LISP.

The main goal of the software components in the diagnostic framework is to perform diagnostic experiments, or to play scenarios to the diagnostic algorithms. The scenarios are stored in text files. The scenario data source framework component reads the scenario files and sends commands and sensor data to the diagnostic algorithms. Both the scenario data source and the diagnostic algorithm send data to the scenario recorder. The latter is in charge of creating output scenarios that contain all events in the original scenarios but also the diagnoses computed by the diagnostic algorithms. These output scenarios are evaluated by a scenario evaluator to compute several performance metrics. All framework components are managed (started, stopped, monitored, and so on) by the scenario loader. The framework is designed with security in mind. The fault injection events, for example, are not submitted to the diagnostic algorithms, to prevent cheating.

We envision the diagnostic framework as a step toward the adoption of a wider diagnostic standard and, we hope, one day, the design of a universal industrial protocol for exchange of diagnostic information.

Competition Tracks

Diagnostics has deep roots in the aerospace industry. Users from this industry require fast and accurate reasoning in order to achieve autonomy or to provide decision support to human flight operators. The system in this track is the electrical power system (EPS) test bed in the ADAPT lab at the NASA Ames Research Center. The system is a hardware test bed of an electrical power and distribution system of a satellite and illustrates many of the design properties that present in real-world satellites. These properties are controllability and observability as well as redundancy. A subcircuit of ADAPT is called ADAPT-Lite and is used as a lightweight competition track. Diagnostic algorithms in the ADAPT and ADAPT-Lite tracks are benchmarked with a large number of diagnostic scenarios, including single and multiple faults. Some of these scenarios are created by hardware sensor data, others programmatically by using a software simulator.

The synthetic track consists mainly of the ten ISCAS-85 logic circuits that were initially used for automated test pattern generation (ATPG). These are net lists of (old) real-world integrated circuits (ICs) such as arithmetic-logic units (ALUs), error correction codes (ECCs), adders, and multipliers. We have also added four smaller circuites from the 74XXX IC family as simpler cases for the DAs. The modeling of the systems in this track is trivial, and there are several easy algorithms that can be used for diagnosis. The challenge in this track, however, is not modeling, but globally optimizing the computational performance vs. diagnostic accuracy. The largest of these circuits (c7552), for example, has 3512 logic gates, which presents problems for many DAs. Further, there is no restriction on the number of faults that we inject. They may mask or may not mask, making the optimization of computational performance versus diagnostic accuracy nontrivial. The DAs must report results within 30 seconds of CPU time.

Diagnostic Approaches

FACT (Roychoudhury, Biswas, and Koutsoukos 2009) is a model-based diagnosis system that uses hybrid bond graphs, and models derived from them, at all levels of diagnosis, including fault detection, isolation, and identification. Faults are detected using an observer-based approach with statistical techniques for robust detection. Faults are isolated by matching qualitative deviations caused by fault transients to those predicted by the model. For systems with few operating configurations, fault isolation is implemented in a compiled form to improve performance.

Fault Buster, uses a combination of multivariate statistical methods for the generation of residuals. Once the detection has been done, a neural networkperforms classification for computing isolation.

GoalArt Diagnostic System (Larsson 1996) is based on multilevel flow models, which are crisp descriptions of flows of mass, energy, and information. It performs fast root-cause analysis with linear computational complexity. Its main advantage is that it is very efficient to engineer a model. The algorithm has been proven in several commercial applications.

The hybrid diagnosis engine (HyDE) (Narasimhan and Brownston 2007) is a model-based diagnosis engine that uses consistency between model predictions and observations to generate conflicts that in turn drive the search for new fault candidates. HyDE uses discrete models of the system and a discretization of the sensor observations for diagnosis. HyDE-S uses the HyDE system but runs it on interval valued hybrid models and the raw sensor data.

LYDIA is a declarative modeling language specifically developed for model-based diagnosis (MBD).

The language core is propositional logic, enhanced with a number of syntactic extensions for ease of modeling. The accompanying tool set currently comprises a number of diagnostic engines and a simulator tool (Feldman, Provan, and van Gemund 2009).

NGDE is an Allegro Common Lisp implementation of the classic general diagnostic engine (GDE). NGDE (de Kleer 2009) uses a minimum-cardinality candidate generator to construct diagnoses from conflicts. For ADAPT-Lite it uses interval constraints.

ProADAPT (Mengshoel 2007) processes all sensor data and then acts as a gateway to a probabilistic inference engine. The inference engine uses an arithmetic circuit evaluator that is compiled from Bayesian network models. The primary advantage of using arithmetic circuits is speed, which is key in resource-bounded environments.

RacerX is a detection-only algorithm that detects a percentage change in individual filtered sensor values to raise a fault detection flag.

RODON (Karin, Lunde, and Münker 2006) is based on the principles of the GDE as described by de Kleer and Williams (1987) and the G+DE (Heller and Struss 2001). RODON uses contradictions (conflicts) between the simulated and the observed behavior to generate hypotheses about possible causes for the observed behavior. If the model contains failure modes in addition to the nominal behavior, these can be used to verify the hypotheses, which speeds up the diagnostic process and improves the results.

RulesRule is a rule-based isolation-only algorithm. The rule base was developed by analyzing the sample data and determining characteristic features of faults. There is no explicit fault detection though isolation implicitly means that a fault has been detected.

StanfordDA is an optimization-based approach to estimating fault states in direct current power systems. The model includes faults changing the system topology along with sensor faults. The approach can be considered as a relaxation of the mixed estimation problem. The authors have developed a linear model of the circuit and use convex optimization to estimate the faults and other hidden states. A sparse fault vector solution is computed by using L1 regularization (Zymnis, Boyd, and Gorinevsky 2009).

Wizards of Oz (Grastien and Kan-John 2009) is a consistency-based algorithm. The model of the system completely defines the stable (static) output of the system in case of normal and faulty behavior. Given a new command or new observations, the algorithm waits for a stable state and computes the minimum diagnoses consistent with the observations and the previous diagnoses.

Future Work

In 2013 we started to prepare a thermal track. The creation of this track has been motivated by a survey of the U.S. Department of Energy, which states that 54 percent of the total energy consumption of the United States in 1986 was for space heating, ventilation, and air conditioning. Modern air-handling units share design properties such as compensating control. Early detection of faults in heating units will lead to timelyrepair and subsequently decrease the total energy consumption. The sampling frequency of the thermal track scenarios is one minute. The typical scenario is 24 hours. The thermal scenarios also depend on environmental factors such as outside temperature and humidity, which are supplied as inputs to the diagnostic algorithms.

We are planning to introduce a robotic track. The subject of this track is going to be a rover. Its position is supplied by a localization system (for example an overhead camera). The goal of this diagnostic track would be to infer motor failure from changes of trajectory.

There is more planned work on control systems such as the ones in chemical plants, software, and others.

Lessons Learned

Communication between the diagnostic algorithm and diagnostic framework is complex. DAs receive sensor streams, report tentative results as a stream, possibly propose information-gathering actions to take on the simulated faulty system, possibly taking repair actions, and so on. This raised two serious challenges. First, designing a scoring metric that could not be gamed and was close to real costs incurred in diagnosing actual systems. For example, how should one score a DA that first reports a correct result at t_1 and an incorrect result at t_2. Or, consider a null DA that always reported there was nothing wrong at every time. As components fail rarely, this null DA might have a very high score. Second, most participants underestimated how much effort it would take to modify their algorithm to interact with the world as represented by the diagnostic framework. To ameliorate this we found it important to distribute the full diagnostic framework to participants well before the competition deadline. Nevertheless, the number of participants declined over the years, which we believe is due to the inherent complexity of the framework and the fact that the best algorithms turned out be very hard to beat, even with more years of research. We believe we need to publicize this competition more widely. This article is one way to achieve this.

Details for participating in the fifth competition are described at dxc-2014.org.

References

de Kleer, J. 2009. Minimum Cardinality Candidate Generation. Paper presented at the 20th International Workshop on Principles of Diagnosis (DX-09), Stockholm, Sweden, 14–17 June.

de Kleer, J., and Williams, B. 1987. Diagnosing Multiple

Faults. *Artificial Intelligence* 32(1): 97–130. dx.doi.org/10.1016/0004-3702(87)90063-4

Feldman, A.; Kurtoglu, T.; Narasimhan, S.; Poll, S.; Garcia, D.; de Kleer, J.; Kuhn, L.; and van Gemund, A. 2010. Empirical Evaluation of Diagnostic Algorithm Performance Using a Generic Framework. *International Journal of Prognostics and Health Management* 1(1): 40.

Feldman, A.; Provan, G.; and van Gemund, A. 2009. The LYDIA Approach to Combinational Model-Based Diagnosis. Paper presented at the 20th International Workshop on Principles of Diagnosis (DX-09), Stockholm, Sweden, 14–17 June.

Grastien, A., and Kan-John, P. 2009. Wizards of Oz — Description of the 2009 DXC Entry. Paper presented at the 20th International Workshop on Principles of Diagnosis (DX-09), Stockholm, Sweden, 14–17 June.

Heller, U., and Struss, P. 2001. G+DE—the Generalized Diagnosis Engine. Paper presented at the 12th International Workshop on Principles of Diagnosis (DX-01), Via Latte, Italian Alps, Italy 7–9 March.

Karin, L.; Lunde, R.; and Münker, B. 2006. Model-Based Failure Analysis with RODON. In *Proceedings of the 17th International Conference on Artificial Intelligence* (ECAI 06). Amsterdam, The Netherlands: IOS Press.

Larsson, J. E. 1996. Diagnosis Based on Explicit Means-End Models. *Artificial Intelligence* 80(1). dx.doi.org/10.1016/0004-3702(94)00043-3

Mengshoel, O. 2007. Designing Resource-Bounded Reasoners Using Bayesian Networks: System Health Monitoring and Diagnosis. Paper presented at the 18th International Workshop on Principles of Diagnosis (DX-07), Nashville, TN, 29–31 May

Narasimhan, S., and Brownston, L. 2007. HyDE: A General Framework for Stochastic and Hybrid Model-Based Diagnosis. Paper presented at the 18th International Workshop on Principles of Diagnosis (DX-07), Nashville, TN, 29–31 May.

Roychoudhury, I.; Biswas, G.; and Koutsoukos, X. 2009. Designing Distributed Diagnosers for Complex Continuous Systems. *IEEE Transactions on Automation Science and Engineering* 6(2): 277–290. dx.doi.org/10.1109/TASE.2008.2009094

Zymnis, A.; Boyd, S.; and Gorinevsky, D. 2009. Relaxed Maximum a Posteriori Fault Identification. *Signal Processing* 89(6): 989–999. dx.doi.org/10.1016/j.sigpro.2008.11.014

Alexander Feldman is a founder and a president of General Diagnostics, a privately owned AI lab in Delft, The Netherlands. Alexander Feldman worked as a postdoc at University College Cork and a visiting researcher at Ecole Polytechnique Fédérale de Lausanne (EPFL), Delft University of Technology, and PARC Incorporated (former Xerox PARC). He obtained his Ph.D. (cum laude) in computer science/artificial intelligence and M.Sc. (cum laude) in parallel and distributed systems from the Delft University of Technology. He has published in leading conference proceedings and international journals covering topics in artificial intelligence, model-based diagnosis, and engineering. In cooperation with NASA Ames Research Center and PARC, Alexander Feldman coorganized the International Diagnostic Competitions (DXC). Feldman's interest cover a wide spectrum, including topics such as model-based diagnosis, automated problem solving, software and hardware design, design of diagnostic space applications, digital signal processing, and localization.

Johan de Kleer is a research fellow and area manager of the model-based reasoning area at Xerox's Palo Alto Research Center. His core interest is building a system that can reason about the physical world as well as he can. de He received his Ph.D. from the Massachusetts Institute of Technology in 1979 in artificial intelligence. He has published widely on qualitative physics, model-based reasoning, truth maintenance systems, and knowledge representation. He has coauthored three books: *Readings in Qualitative Physics*, *Readings in Model-Based Diagnosis*, and *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 theAssociation for the Advancement of Artificial Intelligence and the Association of Computing Machinery.

Tolga Kurtoglu is the director of PARC's Design and Digital Manufacturing Program, where he leads business development, strategy, execution, and technology commercialization for a portfolio of software technologies serving computer aided-design (CAD), product life-cycle management (PLM), and digital manufacturing markets. Recently, he has been the project lead and PI for four DARPA projects: META, iFAB, C2M2L, and iFoundry. His research focuses on design and development of complex systems, applied intelligence for engineering systems, design theory and methodology with a specialization in design creation and innovation, and design automation and optimization. His research spans the areas of model-based systems engineering, automated reasoning, prognostics and health management, and risk and reliability-based design.

Sriram Narasimhan is a project scientist with the University of California, Santa Cruz working as a contractor at the NASA Ames Research Center in the Discovery and Systems Health area. His research interests are in model-based diagnosis with a focus on hybrid and stochastic systems. He is the technical lead for the Hybrid Diagnosis Engine (HyDE) project. He received his M.S. and Ph.D. in electrical engineering and computer science from Vanderbilt University. He also has an MS in economics from Birla Institute of Technology and Science.

Scott Poll received a BSE degree in aerospace engineering from the University of Michigan, Ann Arbor, in 1994, and an MS degree in aeronautical engineering from the California Institute of Technology, Pasadena, in 1995. He is a researcher at the National Aeronautics and Space Administration (NASA) Ames Research Center in Moffett Field, CA, where he is the deputy lead for the Diagnostics and Prognostics Group in the Intelligent Systems Division. He is coleading a laboratory designed to enable the development, maturation, and benchmarking of diagnostic, prognostic, and decision technologies for systems health management applications.

David Garcia is a software developer in the Innovation, Design, and Engineering Analytics (IDEA) team at the Palo Alto Research Center, where he designs and implements software solutions for various research projects. His current focus is on data analytics and software solutions for health care. Prior to joining PARC, Garcia worked at NASA Ames Research Center, where he wrote software to aid research performed by the Diagnostics and Prognostics group, and led the development of the Diagnostic Competition Framework, a suite of software tools for evaluating and comparing diagnostic technologies. Garcia received his BS in mathematics from Santa Clara University and is working toward an MS in computer science from Stanford University.

Lukas Kuhn is a cofounder of Zenhavior. Prior to Zenhavior, he worked as a senior system engineer at Qualcomm R&D on the first generation of mass-scale embedded contextual computing. He received a Ph.D. and a diploma (equivalent to a Master's) in computer science from the Technical University of Munich and the University of Munich, respectively. Before joining Qualcomm in 2010, he was a research assistant with the Embedded Reasoning Area at the Palo Alto Research Center, where he worked on the integration of model-based diagnosis and planning. His current work focuses on reasoning for contextual awareness, mobile computing, and behavior modeling.

Arjan J. C. van Gemund received a BSc in physics, an MSc degree (cum laude) in computer science, and a Ph.D. (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 has been a member of the Electrical Engineering, Mathematics, and Computer Science faculty ofDelft University of Technology, the last 6 years serving as a full professor in the area of fault diagnosis of embedded hardware and software systems. He has coauthored more than 200 scientific publications, and is corecipient of 8 best paper awards.