Diagnostic Fusion for Dependable Vehicle Architectures

Patrick E. Lanigan, Priya Narasimhan
Carnegie Mellon University
Electrical & Computer Engineering
planigan@ece.cmu.edu, priya@cs.cmu.edu

Thomas E. Fuhrman
General Motors Research & Development
Electrical & Controls Integration Lab
thomas.e.fuhrman@gm.com

Abstract – **Despite extensive design processes, emergent and anomalous behavior can still appear at runtime in dependable automotive systems. This occurs due to the existence of unexpected interactions and unidentified dependencies between independently-developed components. Therefore, system-level mechanisms must be provided to quickly diagnose such behavior and determine an appropriate corrective action. DIAGNOSTIC FUSION describes a holistic process for synthesizing data across design stages and component boundaries in order to provide an actionable diagnosis.**

Keywords – diagnosis; dependable computing; automotive; data fusion

1 Introduction

A growing trend in the automotive industry is toward features that assist the driver in maintaining safe control of the vehicle under a variety of conditions. Previously, such assistance has been provided *passively* in the form of information or warnings. These features are now being given increasing amounts of authority to control the vehicle's motion by *actively* supplementing the driver's inputs. The long term trend is towards fully autonomous operation [17, 19].

This trend has largely been enabled by advances in software-intensive distributed systems. Because these are safety-critical systems, they must be designed to tolerate faults and provide high levels of dependability. Typically, a systematic safety analysis is conducted during the design phase to evaluate both the severity and likelihood of the consequences of possible faults. Formal verification methods are used to analyze system dependability. The upcoming ISO 26262 standard for functional safety in automotive electronics recommends that fault-injection also be included as part of the dependability analysis of critical systems [10].

Despite these extensive design processes, emergent and anomalous behavior can still appear at runtime in dependable automotive systems. This occurs due to unexpected interactions and unidentified dependencies between independently-designed components. These interactions are not readily apparent to the system designers and might not be captured by system models. Therefore, system-level mechanisms must be provided to quickly diagnose such behavior and determine an appropriate corrective action at runtime. Diagnostic approaches that operate strictly at the component or subsystem level and rely mainly on functional models may not provide a satisfactory diagnosis. A holistic approach that analyzes empirical metrics *as well as* functional models, and then synthesize the information *across* component and subsystem boundaries is needed.

2 Diagnostic Fusion

Sensor fusion is a well-known technique for combining multiple sources of sensor information, and then correlating that information to get a composite view of the state of the environment being sensed, as well as the state of health of the sensors being fused. Sensor fusion does not itself come up with new sensing technologies, but combines the existing sensing technologies at the system level. By analogy, *diagnostic fusion* does not define new diagnosis algorithms or methodologies, but finds ways to combine existing diagnosis algorithms and methodologies at the system level to satisfy the goals, requirements, and constraints of the system.

Diagnostic fusion is parameterized by the *instrumentation* that it uses to collect data (see Section 2.1) and the *algorithms* that it uses to extract and combine information from the collected data (see Section 2.2). Defining these parameters involves navigating the tradeoff space discussed in Section 3.

2.1 Instrumentation

Instrumentation refers to a source of data from which information can ultimately be obtained through analysis. The holistic approach that we propose for diagnostic fusion is unique in that it instruments the design process as well as

 ISBN: 978-1-61208-149-6

the developed system. The data obtained through design-time instrumentation ultimately drives the instantiation of run-time instrumentation (see Section 3.1).

Design-time instrumentation points are derived from the artifacts produced at each stage of the design process. Such artifacts include design documents, models and empirical data. For example, Failure Mode and Effects Analysis (FMEA) documents provide information that can be used to develop fault signatures. Fault Tree Analysis (FTA) documents can contain information that characterizes dependencies and interactions between system components. Component and system models (e.g., MATLAB/Simulink models) can be used to identify potential run-time instrumentation points. Fault models, functional requirements, and safety specifications can be used to derive the level of detail that an actionable diagnosis is required to provide. Empirical data from fault-injection processes and vehicle prototypes can provide information on normal versus abnormal system behavior.

At run-time, there are numerous discrete instrumentation points that can provide diagnostic data. These instrumentation points can provide black-box, white-box or gray-box views and exist at the system-level as well as the component-level. Examples of potential run-time instrumentation points are the error indicators provided by the communication controller, hooks inserted into software components [13], and Operating System (OS) metrics such as context-switch rates. Another approach demonstrates the extent to which diagnosis is possible using only passive monitoring in FlexRay-based networks [2]. Several aspects of the FlexRay protocol that can be used to aid diagnosis under such restrictions, such as syntactic failures in the value domain (e.g., Cyclic Redundancy Check (CRC), header values), semantic failures in the value domain (e.g., application specific plausibility checks) and failures in the time domain (e.g., early, late or missing messages).

2.2 Algorithms

By combining and analyzing the trends of data at the component-level, subsystem-level and system-level, the diagnostic fusion module can detect escalating anomalous behavior in the system, localize the source of the problem and provide an actionable diagnosis. This approach involves employing data analysis through machine-learning and data-mining techniques on the generated error logs and the instrumented data that is extracted out of the system and its components. The diagnostic fusion process will need to correlate time-stamped data across multiple Electronic Control Units (ECUs) and subsystems, not only to localize the source of a failure, but also to examine possible propagation paths that can lead to additional, related failures.

Failure diagnosis approaches in enterprise systems typically localize anomalous system behavior through statistical analysis of time-series data [6, 8, 9, 11, 14] or through control-flow and data-flow analysis [1, 3, 5, 7, 12]. However, the failure diagnosis approaches developed for enterprise systems might not be directly applicable to automotive systems because automotive systems have limited processing and storage capacity and might not support the level of instrumentation and processing needed by the enterprise approach. Automotive systems also generally require a higher degree of accuracy and lower diagnosis latencies than enterprise systems due to the safety-critical and interactive nature of chassis and powertrain subsystems.

For example, peer comparison is a valuable tool for anomaly-detection, especially in enterprise environments with fluctuating workloads. However, peer comparison is less effective with correlated failures, which occur when a fault originates in one node and then propagates to other nodes in the system. Emergent behavior is likely to arise from correlated failures between components with unidentified dependencies. Peer comparison also requires a certain level of homogeneity to exist between the compared peers, whereas automotive distributed systems are largely heterogenous.

The classic formulation of system diagnosis is the *Preperata-Metze-Chien (PMC) model* [16]. Under the PMC model, components test each other according to predetermined assignments. The set of test results (called the **syndrome**) can then be collected and analyzed to determine the health of each component. Subsequent work extended the PMC model by addressing limitations that made it impractical for application in real fault-tolerant systems; an extensive survey of such work is available [4]. System-diagnosis algorithms developed for automotive systems leverage local status-indicators provided to produce a global view of the network's health [15, 18]. This is accomplished by disseminating and aggregating diagnostic information via diagnostic messages, and then performing analysis on the aggregated data.

3 Diagnostic Requirements and Tradeoffs

Developing a run-time instantiation of the diagnostic fusion process will require navigating a complex tradeoff space, which is comprised of the relationships between *coverage*, *latency*, *accuracy*, and *cost* requirements.

- The diagnostic **latency** is the time between the activation of a fault and the output of an actionable diagnosis.
- Safety process standards define **coverage** as the percentage of possible errors that can actually be detected by a system. In functional safety standards such as ISO26262 [10], the required coverage will vary with

 ISBN: 978-1-61208-149-6

Automotive Safety Integrity Level (ASIL). For example, an ASIL D system may require 99% coverage of all memory errors, while an ASIL C system may require only 90% coverage.

- Diagnostic **accuracy** is the probability that any given diagnosis is correct, and can be expressed in terms of false-positive and false-negative rates.

- Diagnosis will impose some overhead, or **cost**, on the system. The cost can also be expressed in economic terms if additional resources are required to compensate for or implement diagnostic functions.

Some very simple examples serve to illustrate the tradeoff space. Clearly, a highly accurate diagnosis might take longer to produce, increasing latency. If you need low latency, you might have to allow for a reduction in accuracy. On the other hand, in some instances, the latency could be decreased by adding resources, thereby raising the cost. High coverage might require more resources or instrumentation, which would also increase the cost of diagnosis. The diagnostic fusion process could cover a large space of faults with reduced accuracy, or a small space of faults with greater accuracy, especially when trying to discriminate between fault types.

3.1 Diagnosis Advisor

Manually balancing these tradeoffs can be a significant challenge for system designers. Therefore, we further propose a Diagnosis Advisor (DA) that characterizes the system at design-time and develops set of parameters that are used to instantiate the diagnostic fusion process at run-time. Artifacts from each stage of the design process are provided as inputs to the DA. Such inputs could include fault models, FMEA documents, dependability requirements, feature specifications, or functional models. The DA analyzes these inputs, and provides the system designer with a set of parameters that can be used by the diagnostic fusion process to fulfill the diagnostic requirements of the system.

The analysis performed by the DA is aimed at determining an appropriate set of parameters for instantiating the diagnostic fusion process at run-time. The DA does not develop new algorithms by itself. Rather, it aids the system designer in choosing from a set of existing algorithms that can later be combined by the diagnostic fusion process at run-time. The DA performs its analysis at design-time, and so can be a centralized tool. However, the DA could output either a centralized or distributed configuration for the diagnostic fusion module, depending on the system's diagnostic and dependability requirements.

4 Research Questions and Challenges

This work seeks to address three key research questions.

Research Question 1 *Given specific input requirements relating to cost, dependability and performance, how can a configuration of instrumentation-sources and analysis-algorithms be synthesized, which can diagnose emergent behavior effectively and within the latency required for an actionable response?*

Research Question 2 *What is the coverage of the diagnostic configuration output by the DA, and how do we handle cases when such a configuration is impossible due to the constraints imposed on the system?*

Research Question 3 *What is the trade-off space of the cost (i.e., the overhead of increased instrumentation) vs. the accuracy of fault-localization?*

There are challenges related to each of the following aspects: *holistic*, *data-driven* and *diagnosis*. For instance, diagnostic fusion aims to extract data from every phase of the development cycle. However, it is even more important to extract the *right* data. Moreover, this data comes at the expense of human overhead (e.g., development hours) as well as system and communication overhead. This overhead will need to be minimized, as well as balanced with the benefits provided by diagnostic fusion. Appropriate analytic techniques will then need to be developed, applied and evaluated in order to provide actionable diagnostic output. Finally, the the dependability of the DA and its outputs must be analyzed.

Potential sources of data have to be identified, even though it is not clear what instrumentation points will be available in future automotive architectures. Each potential instrumentation point will then need to be characterized with respect to the utility it provides and the overhead it imposes. Moreover, the relationships between instrumentation points have to be studied. For example, if some sources of instrumentation are redundant or synergistic, can they be correlated as a sanity check? On the other hand, can sources of instrumentation that are disjoint or independent be leveraged to provide a more complete picture of the vehicle's health?

Identifying algorithms that can detect specific kinds of failures based on the instrumentation available to them is a key issue. Once these algorithms have been identified, they will need to be implemented in a resource-constrained environment. For algorithms developed in enterprise environments, this could be a significant challenge. Just as with instrumentation points, the utility provided and overhead imposted by the algorithms will also need to be characterized. Further, the diagnostic accuracy and granularity (e.g.,

 ISBN: 978-1-61208-149-6

component-level, subsystem-level, etc.) provided by various combinations of algorithms and instrumentation should be shown experimentally as well as analytically.

5 Summary

Despite extensive design processes, emergent behavior can still appear at runtime in dependable automotive systems. The holistic approach used in diagnostic fusion can address this problem in two ways. First, by synthesizing data across design phases, dependencies and interactions that could have otherwise been undetected can be identified. Second, by coherently characterizing the expected behavior of the system, diagnostic fusion will provide a more robust means of detecting and diagnosing emergent behavior as it occurs.

References

[1] M. K. Aguilera, J. C. Mogul, J. L. Wiener, P. Reynolds, and A. Muthitacharoen. Performance debugging for distributed systems of black boxes. In *Proceedings, 19th ACM Symposium on Operating Systems Principles*, SOSP '03, pages 74–89, New York, NY, USA, October 2003. ACM.

[2] E. Armengaud and A. Steininger. Pushing the limits of online diagnosis in flexray-based networks. In *Proceedings, 2006 IEEE International Workshop on Factory Communication Systems*, WFCS '06, pages 44–53, Piscataway, NJ, USA, June 2006. IEEE.

[3] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. *ACM SIGCOMM Computer Communication Review*, 37(4):13–24, August 2007.

[4] M. Barborak, M. Malek, and A. Dahbura. The consensus problem in fault-tolerant computing. *ACM Computing Surveys*, 25(2):171–220, June 1993.

[5] P. Barham, A. Donnelly, R. Isaacs, and R. Mortier. Using magpie for request extraction and workload modelling. In *Proceedings, 6th USENIX Symposium on Operating Systems Design and Implementation*, OSDI '04, pages 259–272, Berkeley, CA, USA, December 2004. USENIX Association.

[6] S. Bhatia, A. Kumar, M. E. Fiuczynski, and L. L. Peterson. Lightweight, high-resolution monitoring for troubleshooting production systems. In *Proceedings, 8th USENIX Symposium on Operating Systems Design and Implementation*, OSDI '08, pages 103–116, Berkeley, CA, USA, December 2008. USENIX Association.

[7] R. P. J. C. Bose and S. H. Srinivasan. Data mining approaches to software fault diagnosis. In *Proceedings, 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications*, RIDE-SDMA 2005, pages 45–52, Los Alamitos, CA, USA, April 2005. IEEE Computer Society.

[8] I. Cohen, S. Zhang, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox. Capturing, indexing, clustering, and retrieving system history. *ACM SIGOPS Operating Systems Review*, 39(5):105–118, December 2005.

[9] M. Hauswirth, P. F. Sweeney, A. Diwan, and M. Hind. Vertical profiling: Understanding the behavior of object-oriented applications. In *Proceedings, 19th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications*, OOSPLA '04, pages 251–269, New York, NY, USA, October 2004. ACM.

[10] *ISO/DIS 26262: Road vehicles – Functional safety*, volume 4–6. International Organization for Standardization, Geneva, Switzerland, 2009.

[11] S. Kavulya, R. Gandhi, and P. Narasimhan. Gumshoe: Diagnosing performance problems in replicated file-systems. In *Proceedings, 2008 IEEE Symposium on Reliable Distributed Systems*, SRDS '08, pages 137–146, Los Alamitos, CA, USA, October 2008. IEEE Computer Society.

[12] E. Kiciman and A. Fox. Detecting application-level failures in component-based internet services. *IEEE Transactions on Neural Networks: Special Issue on Adaptive Learning Systems in Communication Networks*, 16(5):1027–1041, September 2005.

[13] P. E. Lanigan, P. Narasimhan, and T. E. Fuhrman. Experiences with a CANoe-based fault injection framework for AUTOSAR. In *Proceedings, 2010 IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '10, pages 569—574, Los Alamitos, CA, USA, June 2010. IEEE Computer Society.

[14] S. Pertet, R. Gandhi, and P. Narasimhan. Fingerpointing correlated failures in replicated systems. In *Proceedings, 2nd USENIX Workshop on Tackling Computer Systems Problems with Machine Learning Techniques*, SysML '07, pages 9:1–9:6, Berkeley, CA, USA, April 2007. USENIX Association.

[15] P. Peti, R. Obermaisser, and H. Kopetz. Out-of-norm assertions. In *Proceedings, 11th IEEE Real Time and Embedded Technology and Applications Symposium*, RTAS '05, pages 280–291, Los Alamitos, CA, USA, March 2005. IEEE Computer Society.

[16] F. P. Preperata, G. Metze, and R. T. Chien. On the connection asssignment problem of diagnosable systems. *IEEE Transactions on Electronic Computers*, EC-16(6):848–854, December 1967.

[17] J. D. Rupp and A. G. King. Autonomous driving – a practical roadmap. SAE Technical Paper Series 2010-01-2335, SAE International, Warrendale, PA, USA, October 2010.

[18] M. Serafini, N. Suri, J. Vinter, A. Ademaj, W. Brandstäter, F. Tagliabò, and J. Koch. A tunable add-on diagnostic protocol for time-triggered systems. In *Proceedings, 2007 IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '07, pages 164–174, Los Alamitos, CA, USA, June 2007. IEEE Computer Society.

[19] C. Urmson et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, July 2008.

 ISBN: 978-1-61208-149-6