The Diagnostic Architecture of the PEGASUS Project Car

P. Peti[1], R. Obermaisser[1], and H. Paulitsch[1]

[1]Institute of Computer Engineering,
Real-Time Systems Group,
Vienna University of Technology, Austria
{peti,romano,harald}@vmars.tuwien.ac.at

Abstract — *The automotive industry is at the verve to deploy computer systems not only for safety-related and comfort functionality, but for safety-critical by-wire systems. In the scope of the PEGASUS project a car will be equipped with time-triggered technology in order to provide not only superior car dynamics but also investigate system design and integration on the basis of a series car. As part of this project a diagnostic solution is being developed in order to tackle prevalent diagnostic problems, such as the trouble-not-identified phenomenon in electronic systems, by exploiting the error-containment properties of the underlying architecture.*
In this paper we present the diagnostic architecture of the PEGASUS car that operates on the distributed state of the system in order to trace experienced failures back to the origin and decide on the type of fault (e.g., transient vs. permanent, internal vs. external) that is affecting the system. A necessary prerequisite of such an integrated diagnostic infrastructure is the continuous monitoring and subsequent dissemination of diagnostic information in order to allow a meaningful analysis.

1 Introduction

There is a significant trend in the automotive industry to increase the number of electronic devices in automotive systems. It is estimated that more than 80% of all automotive innovations nowadays stem from electronics [1]. These numbers are underpinned by recent studies of the automotive electronics market that show impressive growth rates in the next four years [2].

Today, the electronic infrastructure of a car consists of a high number of Electronic Control Units (ECUs) (up to more than 70 in today's luxury cars), each typically dedicated to only one application service. The distributed ECUs are interconnected via communication networks with different protocols (e.g., Controller Area Network (CAN) [3]), physical layers, bandwidths (10 kbps–500 kbps), and dependability requirements.

However, despite all the benefits, it is important to state that with the increasing use of electronic devices in transportation systems the likelihood of malfunctions and thus the numbers of defective electronic components will also increase. Nevertheless, diagnostic demands are not solved satisfactorily by current communication systems within the car.

Originally developed to provide simple open/short circuit and abnormal voltage level detection mechanisms, electronic diagnosis evolved into an integral part of every automobile. All modern cars are equipped with On-Board Diagnosis (OBD) systems (OBD-II in USA or EOBD in Europe). OBD, originally developed to continuously monitor the emissions of a car, provides now almost complete engine control and also monitors parts of the chassis, body electronics, and the control network of the vehicle.

However, the development of effective diagnostic systems has stayed behind the recent increase of electronic systems in modern cars. One reason for the diagnostic deficiencies of modern OBD systems is the fact that diagnosis is often treated as add-on to communication systems rather than an integral part of the architecture [4]. Consequently, the problem of the identification of faulty ECUs is one of the predominant challenges that need to be solved.

It takes six months on average for the service technicians at the garages to gain experience with a new car (often due to insufficient technical documentation). However, today's economic pressure in the automotive industry forces the introduction of new car models in decreasing intervals. In recent years the vehicle development cycle was reduced from four to two years to cope with market demands [5]. These technological and business realities underpin the need for effective diagnosis methods.

Though the breakdown logs of the ECUs inform the service technician about detected errors within the system, they do not assist the technician adequately in the identification process [5]. Thus, fully functional units are replaced, or even worse, faulty ECUs remain unchanged in the system. Several factors are relevant to allow for an effective diagnosis, but a pivotal one is the ability to provide error containment that enables the tracing of experienced errors back to the origin. This necessary property is not sufficiently satisfied by the predominant communication infrastructure in the automotive domain, the Controller Area Network (CAN) protocol [3]. These diagnostic deficiencies will become more and more obvious when X-by-wire solutions will be subject to mass production. Emerging X-by-wire solutions will have a lasting effect on the mechanics work, since computer diagnostic will become a standard part of the job [1, 6]. Since a mechanic at a service station is no specialist in automobile electronics, the diagnostic system of the car must provide all necessary information that allows maintenance of faulty components. For this reason it must be possible in modern automotive electronic architectures to trace an entry in a breakdown log back to its source. If this is not possible, as a consequence, fully operational units will be replaced by mistake.

In the remainder of this document we discuss the diagnostic architecture as part of the electronic infrastructure deployed on board of the PEGASUS project car that aims to tackle prevalent diagnostic problems. In particular, this diagnostic solution focuses on providing the maintenance engineer with the necessary information to decide whether a replacement of a particular component is the correct maintenance action. In the scope of the project we will investigate to which extent the error containment and diagnostic properties (e.g., membership) of TTP facilitate diagnosis for maintenance.

The paper is structured as follows. In Section 2 the objective of the PEGASUS project is described. Section 3 elaborates on today's automotive diagnostic strategy. In Section 4 the requirements of a future diagnostic architecture are discussed and Section 5 gives an overview of the fundamental concepts behind the PEGASUS diagnostic architecture. The

paper is concluded in Section 6.

2 The PEGASUS Project

PEGASUS is a joint project between the companies TTTech Computertechnik AG [1], Audi AG Germany [2], and the Institute of Computer Engineering at the Vienna University of Technology funded by the Austrian Advanced Automotive (A3) Technology program of the Austrian Federal Ministry of Transport, Innovation and Technology [3].

The primary objective of PEGASUS is to develop and to implement an application for advanced electronic vehicle dynamics in a series-production vehicle on the basis of the Time-Triggered Architecture (TTA) [7]. The challenge thereby is the integration of the already existing hardware and software architecture of the car on the basis of CAN [3] with the newly developed TTP [8] network controlling advanced powertrain functionality such as dynamic four-wheel-drive. The ultimate goal is to increase vehicle stability in critical situations and achieve better driving agility. The TTA also establishes the foundation for reducing the number of electronic control units deployed in a modern car. As a consequence thereof the need and cost for connectors and cabling will decrease as well. This substantial reduction of the electronic production costs for vehicle dynamics is of great economic importance for car manufacturers.

Furthermore, in the scope of the PEGASUS project new diagnostic strategies are investigated in order to tackle prevalent maintenance problems such as the *Trouble Not Identified (TNI) phenomenon* [9] and to realize advanced diagnostic strategies such as Condition-Based Maintenance (CBM) [10, 11] in the electronics domain. Thereby the error containment capabilities of the TTA [12] will be fully exploited in order to provide a more accurate diagnosis to substantially reduce the service times at the service stations. The PEGASUS project offers the opportunity to validate the advanced diagnostic concepts not only in a laboratory environment test setup but also in the field.

3 Today's Automotive Diagnosis

Each ECU deployed in a car typically has an OBD subsystem that analyzes the functionality of the constituting parts (e.g., via a Built-In Self Test (BIST)) or performs application specific plausibility checks to detect errors.

Once the OBD system of the car detects a violation of the specification of an ECU, a breakdown log entry is written. In case of a high severity, the driver is informed via the Malfunction Indicator Light (MIL). In case of an error, current diagnostic systems provide a so called *freeze frame* function, that records the condition of the vehicle when a failure occurs. The freeze frame provides important information for the failure cause analysis. The breakdown-log typically stores data on the type of fault, the state of the system, the severity, the environmental conditions, a timestamp, and information on the milage of the car.

The maintenance engineer can use this collected data for getting insight into the context of the system malfunction. However, this information is often insufficient to understand

[1] www.tttech.com
[2] www.audi.de
[3] www.bmvit.gv.at

the complex processes within the system that has caused the system to fail. Depending on the type of inspection (e.g., garage, factory inspection, development) different parts of the breakdown log entry are analyzed.

In maintenance mode the ECUs are accessed using dedicated protocols like ISO-9141, J1850 or the CAN based Keyword Protocol (KWP) 2000 [13, 14]. At the service station the mechanic uses a diagnostic testing device to receive information about pending problems. Since most mechanics are no specialists in automotive electronics, the service technician depends on the accuracy of the diagnostic information provided by the OBD. Based on this information the mechanics must be able to determine which part of the system has caused the failure and whether a replacement restores the intended functionality.

4 Requirements for Future Diagnostic Solutions

Any diagnostic solution needs to meet the following requirements in order to reduce the fault-not-found ratio and its far reaching economic implications in electronic systems.

- **Service Technician Assistance.** Since a mechanic is usually no specialist in electronics, the diagnostic system must provide all necessary information that allows the identification of faulty Field Replaceable Units (FRUs) [1]. If this is not possible, fully operational units will be replaced by mistake.
- **Focus on Transients.** The types and causes of failures for electronics have changed over the years. Failure analysis in recent years has revealed that permanent failures have been reduced by improvements in technology but due to the higher level of complexity and downsizing other failure classes have emerged [15]. The tremendous improvements made by the IC industry with respect to permanent failure rates are extenuated by increasing transient failure rates for instance due to semiconductor process variations, shrinking geometries, and lower power voltages [16]. Consequently, the diagnostic services must especially be designed to handle transients.
- **Detection of Correlated Errors.** Diagnostic systems operating only on the internal component state preclude the possibility to detect and analyze correlated failures or system anomalies at different nodes. Thus, a diagnostic architecture must provide means to establish a holistic view of the system by operating on the distributed state.
- **Assessment of Fault-Tolerance Mechanisms.** Fault-tolerance mechanisms are required to achieve the necessary degree of dependability for the deployment of electronic systems in safety-critical environments. In order to reduce application complexity and certification efforts, fault-tolerance mechanisms are ideally provided by the architecture and exploited transparently to the application. However, from a diagnostic point of view, this strategy has far reaching implications. Since it is impossible to detect inconsistency of the fault-tolerant replicas at the application level, diagnostic mechanisms must be provided at architecture level.
- **Support for Advanced Maintenance Strategies.** Time-Based Maintenance (TBM) is increasingly being replaced by Condition-Based Maintenance (CBM), to reduce cost and to improve reliability and system performance [10, 11]. In order to adopt CBM for electronic systems suitable indicators for degradation or wearout must be identified and analyzed to detect deviations from sound operation.

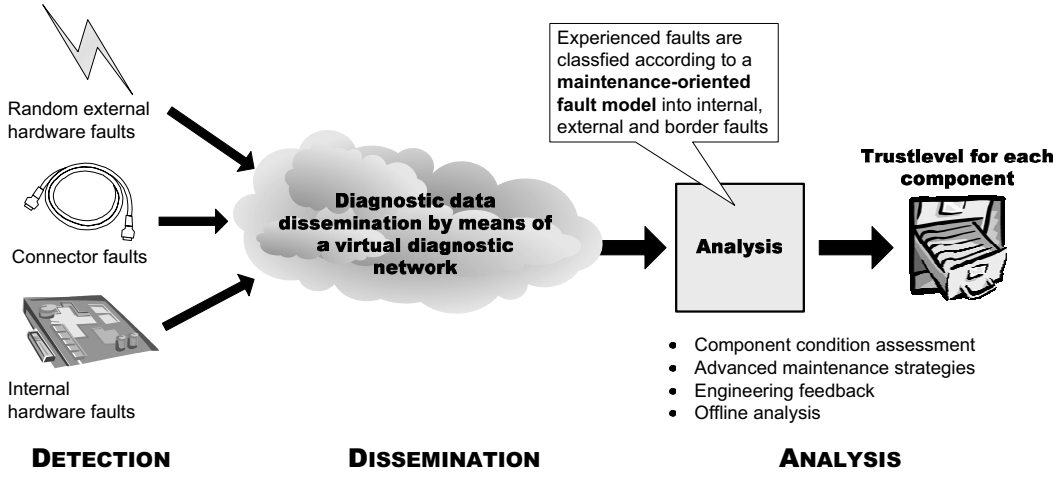

Figure 1: Overview of the Diagnostic Architecture

– **Avoidance of the Probe Effect.** Any diagnostic subsystem must avoid the introduction of probe effects [17] that may forge the outcome of the diagnostic subsystem. This is especially important in case of real-time systems, where the diagnostic messages must not compromise the real-time traffic in any way.
– **Intellectual Property Protection.** Diagnosis is often equated with revealing of internal information. An integrated approach allows the realization of advanced diagnostic strategies by solely operating on the interface state of the linking interfaces [18] without revealing any internals of the application.

5 Overview of the Diagnostic Architecture for the PEGASUS Car

In order to cope with industrial demands on diagnosis for automotive systems an architecture with integrated support for diagnosis is needed. Such an architecture provides the necessary prerequisites to allow the effective detection, identification and classification of experienced errors. An integrated solution, in contrast to an addendum solution, allows the realization of advanced diagnostic strategies and supports the gathering of field data for engineering feedback [4].

The model of the diagnostic architecture as illustrated in Figure 1 can be divided into three main parts, the detection of errors and anomalies on the distributed state of the system, the acquisition of diagnostic information via a dedicated *Virtual Diagnostic Network (VDN)*, and the subsequent analysis in order to determine nature of the experienced fault with respect to a maintenance-oriented fault model. The pivotal strategy of the diagnostic architecture is the establishment of a holistic view on the system by operating on the distributed state. In combination with the error containment mechanisms provided by the TTA [19], this strategy allows tracing correlated system anomalies back to the FRU responsible for the experienced system behavior.

5.1 The Maintenance-oriented Fault Model

As stated in [20] the concept of fault is introduced to stop the recursion of the "fault-error-failure" chain. From a maintenance point of view, we are only interested in categorizing

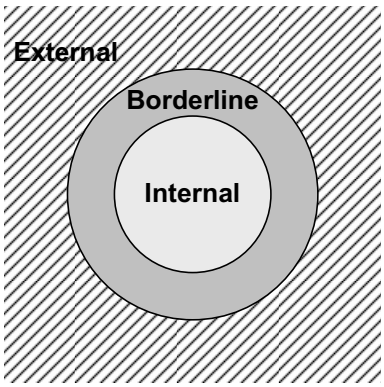

Figure 2: Component Fault Model

the type of fault of the experienced failure into classes that allow a determination whether a replacement is the correct maintenance strategy. Thus, by reversing the fault-error-failure chain, it must be possible for the diagnostic subsystem to determine whether a change of a FRU can eliminate the experienced problem, or if a replacement will prove to be ineffective.

Consequently, we stop the recursion at Field Replaceable Unit (FRU) level. In the context of the PEGASUS car architecture in case of hardware faults the FRU is considered to be a complete node computer. The fault classification for each FRU needs to be derived by analyzing the prevalent types of faults affecting the given FRU.

Consider for instance a crack in a Printed Circuit Board (PCB). Such a crack may originate from wear-out of the material due to environmental stress, such as vibration (e.g., rough roads), shock (e.g., chuckholes, hard landings) and changes in temperature (i.e. expansion and contraction). Depending on operational conditions this crack may cause the component to fail transiently. From a maintenance point of view (at the service station) the first level cause – due to environmental – stress is not of interest. In analogy the exact element of the FRU that is subject to failure is of limited interest for a service technician. By taking a maintenance-oriented view the most important fact we are interested in is that the hardware fault can only be eliminated by replacement of the FRU. The analysis, which part of the FRU caused the malfunction is in the scope of the inspection of faulty nodes at the Original Equipment Manufacturer (OEM) (and not part of the maintenance action at the service station). For more information on the rationale behind the model see also [21].

The proposed maintenance-oriented fault model takes the component-based nature of today's distributed systems into account by considering a component as a FRU for hardware faults. Consequently, we devise the following fault classes as illustrated in Figure 2. Faults that originate outside the component boundaries are denoted as *external faults*. External faults are characterized by having no permanent effect on the functionality of the component. A restart of the component with subsequent state synchronization is a typical strategy to restore a correct state. An example for an external fault is Electromagnetic Interference (EMI) [22]. So-called *borderline faults* are the class of faults that cannot be judged to be external or internal with respect to the component boundary. An example for such a fault is a connector fault (a connector consist of two parts, one attached to the

component, the other attached to the cable loom). Since this class is responsible for a significant number of system failures [23], we extend the boundary classification of faults as introduced by Laprie [20] by adding the class of borderline faults. Borderline faults require a closer inspection by the service technicians. Connector problems are difficult to trace, since the inspection itself can be the corrective action [24]. In case of connectors showing wearout phenomena such as fretting or corrosion, a replacement will be necessary. Finally, *internal faults* cover those faults that originate from within the FRU boundary (e.g., crack in the PCB). In contrast to external faults, these faults can only be eliminated by a replacement of the component.

5.2 Operation on the Distributed State

Informally speaking, the notion of state is introduced in order to separate the past from the future (i.e. a decoupling). *The idea is that if one knows what state the system is in, he could with assurance ascertain what the output will be* [25, p. 45]. Hence, the state accumulates all past history of the given system. Apparently, this definition of state by Mesarovic and Takahara is only meaningful, if the notion of past and future (time) is relevant for the considered system. Central to this definition is the inseparable nature of time and state.

If the time base in a distributed system is dense (the events are allowed to occur at any instant of the timeline), then it is in general not possible to generate a consistent temporal order on the basis of the time-stamps [26]. Due to the impossibility of synchronizing clocks perfectly and the denseness property of real time, there is always the possibility that a single event is time-stamped by two clocks with a difference of one tick. By introducing the concept of a *sparse time base* the ordering of events can be restored without execution of agreement protocols only based on timestamps [27]. In the sparse time model the continuum of time is partitioned into an infinite sequence of alternating durations of activity and silence. Thereby, the occurrence of significant events is restricted to the activity intervals of a globally synchronized action lattice. The interval of silence on the sparse time base is a system wide consistent dividing line between the past and the future and the interval when the state of the distributed system can be defined.

Whenever a fault affects one or more constituting parts of the distributed system, a change of state can occur that leads to an unintended state denoted as an error [20]. Depending on the type of fault (e.g., internal or external fault, software or hardware fault), the unintended state will exhibit a characteristic manifestation in time, value and space. To capture the characteristics of the fault-induced distributed state changes, we introduce the concept of *fault pattern* [28]. A fault pattern is the set of state variables that has been identified as subject to fault-induced state changes along with corresponding properties in value, space, and time. Different types of faults show different fault patterns on the distributed state. For example, a wearout failure due to monotonic accumulation of incremental damage beyond the endurance of the material [29] will exhibit a fault pattern typical for intermittent type faults [20]. This type of fault affects only a single component (space dimension) and reoccurs repeatedly at the same location at higher rates with decreasing intervals [16]. By contrast, a massive transient disturbance (e.g., due to EMI) is an example for the class of faults typically affecting multiple components at the same time. EMI causes correlated effects on the entire system that usually cause no physical

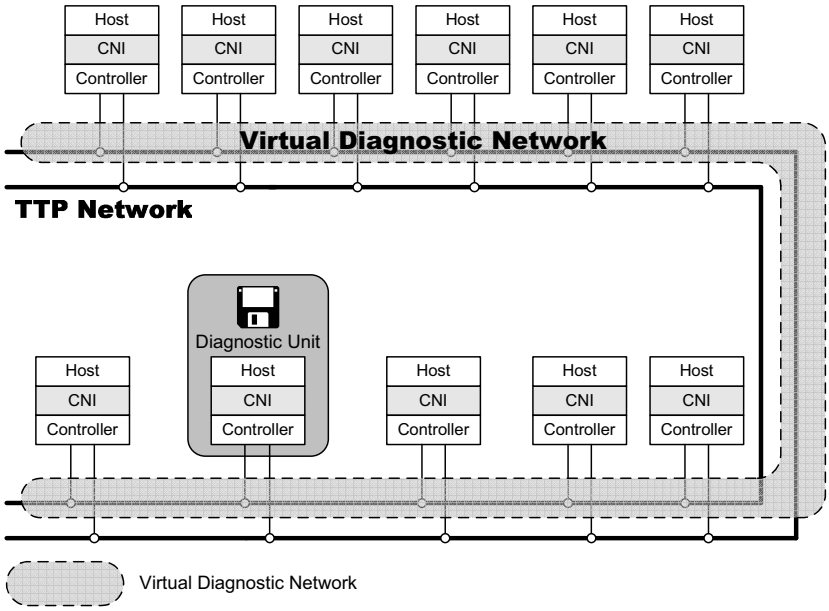

Figure 3: The Virtual Diagnostic Network

damage to hardware [22].

In the diagnostic architecture so-called *Out-of-Norm Assertions (ONAs)* [28] are deployed that are checked against the distributed state established by the use of a sparse time base [27]. We define an *Out-of-Norm Assertion (ONA)* as a predicate on the distributed system state that encodes a *fault pattern* in the value, time and space domain. Out-of-Norm Assertions (ONAs) are deterministically triggered, whenever all *symptoms* of a particular fault pattern are detected on the distributed state. A *symptom* is a condition on a set of interface state variables of a particular component that is monitored to detect deviations from the Linking Interface (LIF) specification [18]. An ONA will likely be composed of more than one symptom, each operating on the interface state of different components. Since ONAs operate solely on the *interface state* [30], the internals of jobs remain hidden. This ensures the protection of intellectual property as required by industry.

ONAs do not provide a definite classification whether a component is correct or incorrect in case only a subset of the specified symptoms fires. In this case, we speak of an *anomaly*, i.e. we cannot ascribe the behavior of the component to a specific fault pattern. In order to decide on the correctness of a component, an assessment over time is necessary. The repeated evaluation of evidence gathered by ONAs provides the foundation for the analysis process that ultimately decides whether a component is correct. ONAs are gathering evidence in order to decide on a particular pattern affecting the state of the system. This process can be compared with gathering evidence of different diagnostic techniques in medicine (e.g., temperature measurement, computer tomography, X-ray). In case sufficient evidence is gathered, a suspicion for a particular disease is confirmed or falsified.

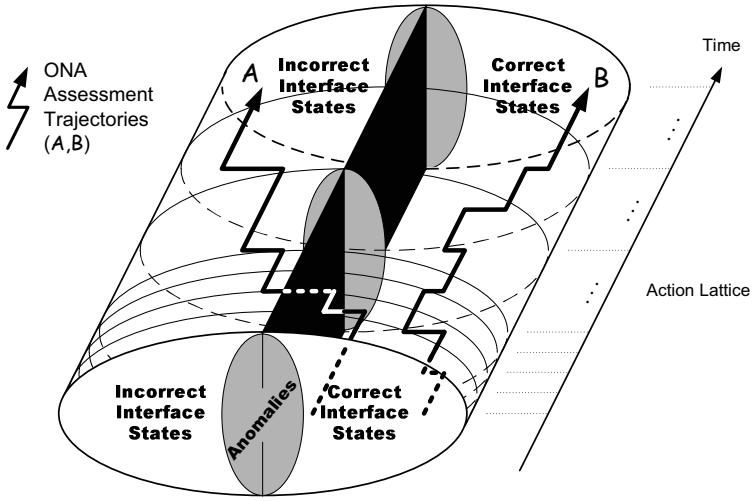

Figure 4: Assessment Process

5.3 Dissemination of Diagnostic Information

The acquisition and dissemination of diagnostic information occurs through high-level architectural services. By exploiting these services, a so-called Virtual Diagnostic Network (VDN) is established. A *virtual network* is an overlay network that is established on top of a physical network [31]. In the PEGASUS system architecture, we provide a virtual network on top of the time-triggered core communication service provided by TTP. This virtual network is tailored to the requirements of the diagnostic subsystem via the provided functionality, the operational properties, the namespace, and the dependability properties.

Furthermore, this virtual network is encapsulated so that communication activities are neither visible nor have any effect (e.g., performance penalty) on the exchange of messages for the dynamic four-wheel-drive in the time-triggered network. Figure 3 illustrates this system architecture. By using only elementary interfaces [32] to establish the virtual diagnostic network, no back-propagation of the diagnostic dissemination service to the application subsystem controlling the car dynamics is possible. This way, the diagnostic architecture need not be validated to the dependability of the highest application criticality class.

Such a virtual solution has two main advantages. At first, real-time traffic is not compromised in any way since the bandwidth for the exchange of diagnostic information is fixed a priori at design time. This way a deterministic message exchange for all non-diagnostic subsystems is guaranteed. Secondly, the purely virtual solution ensures that no additional hardware faults are introduced due to wiring or connector problems. Consequently, no probe effect can be introduced [17].

5.4 Analysis

The analysis subsystem executes algorithms on the gathered diagnostic information in order to assess the condition of each component. By taking a *maintenance-oriented fault model* (see Section 5.1) as the basis for the assessment process, the ultimate goal of the

analysis is to decide whether a FRU (as the smallest replaceable unit with respect to maintenance) should be replaced or remain in the given system. Therefore, the maintenance-oriented fault model defines a classification of faults into internal (e.g., crack in PCB), borderline (e.g., connector failure) or external (e.g., EMI).

The evaluation process performed by the diagnostic subsystem is illustrated in Figure 4. The evaluation process is based on a consistent notion of state, which is provided through the *action lattice* of the sparse time base established by the core services. The arrows in Figure 4 indicate the assessment trajectories. At first both arrows show component conformance with the specification, i.e. correct interface states. As time progresses arrow A exhibits an increasing confidence for a violation of the specification, while arrow B indicates a component behavior in accordance with the specified service.

As a result of the diagnostic algorithms, a *trust level* for each replaceable component of the system is determined that forms the basis for decision-making process of the maintenance engineer. Thus, the system itself analyzes its current condition and informs the service technician about the current health status. This diagnostic infrastructure supports the realization of advanced maintenance strategies such as CBM [10] to be employed in the context of electronic systems.

6 Conclusion

In the scope of the PEGASUS project a series production vehicle is equipped with a TTP network to realize advanced powertrain functionality in order to improve driving comfort and the handling of the car in critical situations. The use of the Time-Triggered Architecture (TTA) as the communication infrastructure excels not only with dependability and composability properties in comparison with today's used technology but also allows investigating and implementing new maintenance strategies. Since the introduction of X-by-wire systems requires new diagnostic concepts to improve capabilities and effectiveness, the field data that will be derived from the experiences gained with a series production car will have significant impact on effective maintenance solutions.

In order to cope with new diagnostic demands of on-board diagnostic subsystems the *design for diagnosis* principle must be obeyed when designing new architectures. Ad-hoc solutions that treat diagnosis as an addendum rather than an integrated part of every system are no longer applicable. By following this principle the introduced diagnostic architecture is designed to reduce the fault-not-found ratio that is currently causing negative media coverage indicating electrical problems as number one reason for car breakdown. By operating on the distributed state of the system a holistic view on the system can be established that allows correlation of experienced failures. Since the majority of today's ECU failures are transient, only an online analysis service is capable of answering the question whether the replacement of a particular ECU will prove to be effective. An accurate diagnosis with short repair time is a prerequisite for binding a customer to the manufacturer.

Future fault injection campaigns will provide interesting data on the effectiveness of the deployed architecture and give important feedback for the implementation of accurate analysis algorithms.

Acknowledgments

This work has been supported in part by the Austrian Advanced Automotive Technology Project under project No. 809437 and the European IST project ARTIST2 under project No. IST-004527 and the European IST project DECOS under project No. IST-511764.

References

[1] G. Leen and D. Heffernan. Expanding automotive electronic systems. *Computer*, 35(1):88–93, January 2002.

[2] The Hansen Report on Automotive Electronics, November 2002. Portsmouth NH USA, `www.hansenreport.com`.

[3] Robert Bosch Gmbh, Stuttgart, Germany. *CAN Specification, Version 2.0*, 1991.

[4] M. Sachenbacher, P. Struss, and R. Weber. Advances in design and implementation of OBD functions for diesel injection based on a qualitative approach to diagnosis. In *Proceedings of SAE 2000 World Congress*, Detroit, MI, USA, 2000. SAE.

[5] J. Barkai. Vehicle diagnostics–are you ready for the challenge? In *Proceedings of Automotive & Transportation Technology (ATT) Congress & Exhibition*, volume 5. SAE, October 2001.

[6] E. Bretz. By-wire cars turn the corner. *IEEE Spectrum*, 38(4):68–73, April 2001.

[7] H. Kopetz and G. Bauer. The time-triggered architecture. *IEEE Special Issue on Modeling and Design of Embedded Software*, January 2003.

[8] H. Kopetz. *Specification of the TTP/C Protocol*. TTTech, Schönbrunner Straße 7, A-1040 Vienna, Austria, July 1999. Available at `http://www.ttpforum.org`.

[9] D.A. Thomas, K. Ayers, and M. Pecht. The 'trouble not identified' phenomenon in automotive electronics. *Microelectronics Reliability*, 42:641–651, 2002.

[10] V. Polimac and J. Polimac. Assessment of present maintenance practices and future trends. In *Proceedings of the Transmission and Distribution Conference and Exposition, IEEE/PES*, pages 891–894, 2001.

[11] C. Teal and D. Sorensen. Condition based maintenance [aircraft wiring]. In *Proceedings of the 20th Conference on Digital Avionics Systems, DASC*, volume 1, pages 3B2/1–3B2/7, October 2001.

[12] H. Kopetz. Fault containment and error detection in the time-triggered architecture. In *Proceedings of the International Symposium on Autonomous Decentralized Systems*, Pisa, Italy, April 2003.

[13] W. Waldeck. Diagnostic protocol challenges in a global environment. In *Convergence International Congress & Exposition On Transportation Electronics*, Detroit, MI, USA, October 2002. SAE.

[14] International Organization for Standardization. *Keyword Protocol 2000, ISO 14230*, 1999.

[15] M. Pecht and V. Ramappan. Are components still the major problem: a review of electronic system and device field failure returns. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 15(6):1160–1164, December 1992.

[16] C. Constantinescu. Impact of deep submicron technology on dependability of VLSI circuits. In *Proceedings of the International Conference on Dependable Systems and Networks*, pages 205–209. IEEE, 2002.

[17] J. Gait. A probe effect in concurrent programs. *Software Practice and Experience*, 16(3):225–233, March 1986.

[18] H. Kopetz and N. Suri. Compositional design of RT systems: A conceptual basis for specification of linking interfaces. In *Proceedings of Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing*, pages 51–60, May 2003.

[19] H. Kopetz. Fault containment and error detection in the time-triggered architecture. In *Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems*, April 2003.

[20] A. Avizienis, J.C. Laprie, and B. Randell. Fundamental concepts of dependability. Research Report 01-145, LAAS-CNRS, Toulouse, France, April 2001.

[21] P. Peti, R. Obermaisser, and H. Kopetz. A maintenance-oriented fault model for the DECOS integrated diagnostic architecture. In *Proceedings of the Workshop on Parallel and Distributed Real-Time Systems 2005 (WPDRTS)*. IEEE, April 2005.

[22] H. Kim, A.L. White, and K.G. Shin. Effects of electromagnetic interference on controller-computer upsets and system stability. *IEEE Transactions on Control Systems Technology*, 8(2):351–357, March 2000.

[23] J. Swingler, J.W. McBride, and C. Maul. Degradation of road tested automotive connectors. *IEEE Transactions on Components and Packaging Technologies*, 23(1):157–164, March 2000.

[24] K. Kimseng, M. Hoit, N. Tiwari, and M. Pecht. Physics-of-failure assessment of a cruise control module. *Microelectronics Reliability*, 39:1423–1444, 1999.

[25] M. D. Mesarovic and Y. Takahara. *Abstract Systems Theory*, chapter 3. Springer-Verlag, 1989.

[26] H. Kopetz. *Real-Time Systems, Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1997.

[27] H. Kopetz. Sparse time versus dense time in distributed real-time systems. In *Proceedings of 12th International Conference on Distributed Computing Systems*, Japan, June 1992.

[28] P. Peti, R. Obermaisser, and H. Kopetz. Out-of-norm assertions. In *Proceedings of the 11th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, March 2005.

[29] A. Ramakrishnan. *The Avionics Handbook*, chapter Electronic Hardware Reliability. CRC Press LCC, 2001.

[30] M.-C. Gaudel, V. Issarny, C. Jones, H. Kopetz, E. Marsden, N. Moffat, M. Paulitsch, D. Powell, B. Randell, A. Romanovsky, R. Stroud, and F. Taiani. Final version of the DSoS conceptual model. *DSoS Project (IST-1999-11585) Deliverable CSDA1*, December 2002. Available as Research Report 54/2002 at http://www.vmars.tuwien.ac.at.

[31] R. Obermaisser. *Event-Triggered and Time-Triggered Control Paradigms – An Integrated Architecture*. Real-Time Systems Series. Kluwer Academic Publishers, November 2004.

[32] H. Kopetz. Elementary versus composite interfaces in distributed real-time systems. In *Proceedings of ISADS'99*, Tokyo, Japan, March 1999.