Development of TCU Diagnostic System in Application Software Based on AUTOSAR

Feng Luo¹, Xiaoying Fang²*

¹School of Automotive Studies, Tongji University, Shanghai, 201804, China
²School of Automotive Studies, Tongji University, Shanghai, 201804, China
³Corresponding author’s e-mail: shirley_fang2004@163.com

Abstract. Based on the diagnostic architecture of AUTOSAR, this paper presents the architecture of AUTOSAR diagnostic system and implementation of application fault diagnostic function. The interaction between the application software components and the BSW modules is emphasized. The hardware-in-the-loop test platform verifies that the development of application diagnostic system fulfills the requirements of design.

1. Introduction
AUTOSAR is an open system architecture standard for automobiles, including important diagnostic systems. The development of automotive electronic system based on AUTOSAR achieves the separation of software design from underlying hardware, which is the trend of automotive electronic embedded system development [1][2]. In this paper, the diagnostic mechanism of AUTOSAR is discussed. Aiming at the transmission control unit (TCU) diagnostic requirement, the mechanism and strategy of AUTOSAR diagnostic system is studied deeply. The design and implementation of fault diagnosis and on application software component (SWC) are emphasized. Finally, the development results are verified by hardware in the loop test.

2. Diagnostic Architecture of AUTOSAR
The BSW modules related to diagnostic system in AUTOSAR include Diagnostic Event Manager (Dem), Diagnostic Communication Manager (Dcm), Diagnostic Log and Trace (Dlt), Function Inhibition Manager (Fim), NVRAM Manager (NvM), ECU Status Manager (EcuM) [3].

Dem is the core modules of AUTOSAR diagnostic system. Dem processes and stores diagnostic events, errors and related data. Application software component and other BSW modules can access Dem to update and retrieve current event state information [4]. Additionally, Fim module evaluates and assigns the handling of the event to the SWC. When Dem informs Fim the change state of event, Fim will decide whether stop or release the functionality upon the assignment [5].

The interfaces and dependencies between Dem with other BSW modules and SWCs are indicated in Figure 1 [6].
3. Fault Diagnosis

Fault diagnosis of TCU application software strategy, like Shift fork position does not match speed ratio, shall be implemented by SWC, Dem and Fim coordinately. Dem stores the monitor result from SWC and informs Fim. Then Fim accomplishes the fault handling and return the assigned action to SWC. The development of application SWC includes two major steps. Firstly, event state of diagnostic monitoring shall be produced by SWC SST and sent to the underlying Dem. Secondly, SWC IPF receives the fault handling instruction from Fim to suppress specific functionality.

3.1 Diagnostic Monitoring

In Dem, Client-Server (C-S) interface defines the standard operation. According to the AUTOSAR, Meanwhile, Provide-Port (PPort), consisting the interface, is defined in Dem. Application SWC SST is responsible for collecting and packing the fault and relevant information. It transfers the event state to Dem via Receive-Port (RPort). The connection is established between PPort and RPort via assembly connector. So SST can send the current event state to Dem by calling the operation.

The detailed design of interface between the ASW and is shown in Table 1, where x represents 1, 2, 3… That is the serial number of application fault state signal.

Table 1. Interface of diagnostic monitoring.

| Interface     | BSW module | Application software component |
|---------------|------------|-------------------------------|
| Diagnostic Monitor | SWC instance | PPort | SWC instance | RPort |
|               | Comp_Service_Dem | PP_a_ASW_x | CPT_SST | RP_a_ASW_x |

The diagram of connection between SST and Dem is shown in Figure 2:
3.2 Function Inhibition
Application retrieves the fault handling instruction from Fim when the fault occurs (including BSW fault and ASW fault), so as to trigger each suppression of relevant functionality in application software module. Fim is responsible for providing the control mechanism for SWC functionality. A functionality can be composed of one, several or parts of software module by setting the same permission/inhibit condition. With Fim, some functionalities can be suppressed by configuring and calibrating.

In Fim, C-S interface defines the standard operation. Meanwhile, PPort, consisting the interface, is defined in Fim. Application SWC IPF is responsible for retrieving the permission from Fim via RPort. Then it informs other SWCs. The connection is established between PPort and RPort via assembly connector. So IPF can read the handling instruction from Fim by calling the operation.

The detailed design of interface between ASW and BSW is shown in Table 2, where x represent 1, 2, 3…That is the serial number of application suppression signal.

Table 2. Interface of function inhibition.

| Interface | BSW module | Application software component |
|-----------|------------|--------------------------------|
| Function Inhibition | Comp_Fim | PP_q_FID_ASW_x |
| | CPT_IPF | RP_q_FID_ASW_x |

The diagram of connection between IPF and Fim is shown in Figure 3:
4. Test Result
Take the application fault ‘shift fork position does not match event speed ratio’ as an example. The test result is shown in Figure 4. When the even shift fork position does not match the calculated speed ratio, the application SWC SST will send the “test pre-failed: 3” as the event state on signal SST_St_PostFrkGearSutEvn to the Dem via RTE. Dem informs Fim according to the change of the event state. Fim then trigger the assignment function inhibition through the signal DINH_stFid.FID_ASW_001 to IPF. Application SWC responds immediately and react the fault handling process. Similarly, test result of “Shift fork position does not match odd speed ratio” is demonstrated as Figure 5. The test results are consistent with the expected requirement and design.
5. Conclusion
After several rounds of HIL test, the implementation of fault diagnosis in application layer based on AUTOSAR basically meets the requirements. That is a good foundation for the next development of OBD and function safety. Furthermore, the diagnostic mechanism of AUTOSAR is extensive and profound. The usage of more functionality is still in further exploration. At the same time, it is also necessary to supplement more specified diagnostic functionality in combination with the vehicle test.
results. It can be predicted that with the deepening of the project, the development of diagnostic system based on AUTOSAR will be further improved and optimized.

References
[1] Oliver S. (2015) AUTOSAR Compendium Part1 Application & RTE. Createspace Independent Publishing Platform, Charleston.
[2] AUTOSAR Development Partnership, 2011. Layered Software Architecture V3.2.0. http://www.autosar.org.
[3] AUTOSAR Development Partnership, 2011. Requirements on Diagnostic V2.4.0. http://www.autosar.org.
[4] AUTOSAR Development Partnership, 2011. Specification of Diagnostic Event Manager V4.2.0. http://www.autosar.org.
[5] AUTOSAR Development Partnership, 2011. Specification of Function Inhibition Manager V2.2.0. http://www.autosar.org.