Document Title	Explanation of System Health Monitoring
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	993

Document Status	published
Part of AUTOSAR Standard	Foundation
Part of Standard Release	R20-11

Document Change History			
Date	**Release**	**Changed by**	**Description**
2020-11-30	R20-11	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

1 Introduction 4
1.1 Objectives . . . 4
2 Definition of terms and acronyms 5
2.1 Acronyms and abbreviations . . . 5
2.2 Definition of terms . . . 5
3 Related Documentation 7
4 Master/Client Architecture 8
4.1 Constraints and Assumptions . . . 8
4.2 Overview . . . 8
4.3 Redundancy of System Health Monitoring Master . . . 9
4.4 Supervision of System Health Monitor . . . 10
4.4.1 Scenario 1: One Platform with SHM Deployed on a Single ECU 10
4.4.2 Scenario 2: Multiple Platforms with SHM . . . 11
4.5 Integration of System Health Monitor in AUTOSAR . . . 12
4.6 Interfaces of System Health Monitor . . . 13
4.6.1 Overview of System Health Monitor Interfaces . . . 13
4.6.2 Information Transmission to SHM . . . 13
4.6.3 Health Indicator Transmission . . . 15
5 Example Health Indicator and Use Case 16
5.1 Example Health Indicator: Sensors . . . 16
5.2 Example Use Case: Health of Service (HoS) Indication . . . 17
5.3 Example Use Case: System Degradation . . . 18
A Appendix 20

1 Introduction

System health monitoring (SHM) introduces platform agnostic health monitoring. SHM focuses on system wide coordination of error handling across multiple platforms on multiple controllers and machines. Currently, the health monitoring and the handling of recovery actions is performed at platform level using PHM in Adaptive platform and WdgM in Classic platform. For this concept, a new component called `System Health Monitor` is introduced. The `System Health Monitor` component can be instantiated either as a Master instance or a Client instance. SHM Client is responsible for communicating platform level health data to the Master instance whereas SHM Master is responsible for determination of `Health Indicators`. The `Health Indicators` can be determined at subsystem level, feature level, domain level and eventually at vehicle level. These `Health Indicators` can be used either for platform level recovery actions or to enhance the services with a Health of Service parameter, similar to Quality of Service.

1.1 Objectives

This document is intended to give more details on how System Health Monitoring or Health Indicators can be implemented. All of the content should be understood as an implementation hint and not mandatory specifications. The actual requirements can be found in [1, RS_HealthMonitoring] and the abstract specifications in [2, ASWS_HealthMonitoring].

2 Definition of terms and acronyms

The glossary below includes acronyms and terms relevant to System Health Monitoring that are not included in the AUTOSAR Glossary.

2.1 Acronyms and abbreviations

Abbreviation / Acronym:	Description:
SHM	System Health Monitor
HI	Health Indicator
SOME/IP	Scalable service-Oriented MiddlewarE over IP
PHM	Platform Health Management
SM	State Management
BswM	Basic Software Mode Manager
WdgM	Watchdog Manager
GSS	Global Supervision Status
LSS	Local Supervision Status

2.2 Definition of terms

Terms:	Description:
Health Indicator	Health Indicator provides an evaluation metric of current system performance with regard to safety requirements.
System Health Monitor	System Health Monitor component allows sytem wide coordination of error handling across several Classic, Adaptive or any third party platforms.
Global Supervision Status	Status that summarizes the Local Supervision Status of all Supervised Entities.
Local Supervision Status	Status that represents the current result of Alive Supervision, Deadline Supervision and Logical Supervision of a single Supervised Entity.
Alive Supervision	Kind of supervision that checks if a Supervised Entity executed in a correct frequency.
Deadline Supervision	Kind of supervision that checks if the execution time between two Checkpoints is within minimum/maximum time limit.
Logical Supervision	Kind of online supervision of software that checks if the software (Supervised Entity or set of Supervised Entities) is executed in the sequence defined by the programmer (by the developed code).
Health Channel	Channel providing information about the health status of a (sub)system. This might be the Global Supervision Status of an application, the result any test routine or the status reported by a (sub)system (e.g. voltage monitoring, OS kernel, ECU status).
Checkpoint	A point in the control flow of a Supervised Entity where the activity is reported.

Terms:	Description:
Supervised Entity	A software entity which is included in the supervision. A Supervised Entity denotes a collection of Checkpoints within a software component. There may be zero, one or more Supervised Entities in a Software Component. A Supervised Entity may be instantiated multiple times, in which case each instance is independently supervised.

3 Related Documentation

[1] Requirements on Health Monitoring
AUTOSAR_RS_HealthMonitoring

[2] Specification of Health Monitoring
AUTOSAR_ASWS_HealthMonitoring

[3] Specification of Platform Health Management for Adaptive Platform
AUTOSAR_SWS_PlatformHealthManagement

[4] Specification of Execution Management
AUTOSAR_SWS_ExecutionManagement

[5] Specification of State Management
AUTOSAR_SWS_StateManagement

[6] Specification of Watchdog Manager
AUTOSAR_SWS_WatchdogManager

[7] Specification of Basic Software Mode Manager
AUTOSAR_SWS_BSWModeManager

[8] Specification of Abstract Platform
AUTOSAR_TPS_AbstractPlatformSpecification

[9] SOME/IP Protocol Specification
AUTOSAR_PRS_SOMEIPProtocol

4 Master/Client Architecture

In this chapter a master/client example architecture for System Health Monitor is shown. Distributed architectures are possible as well, but not in the scope of this example. Integration of SHM client on platform level will be described in upcoming releases.

4.1 Constraints and Assumptions

As logic for Health Indicator (HI) determination is not standardized, only an example will be provided. Concrete mappings for abstract interfaces to Classic or Adaptive Platform interfaces are not provided in the current release. The architectural solution is based on AUTOSAR's R19-11, but already includes new architectural decision on responsibilities of PHM [3], EM [4] and SM [5] in AUTOSAR Adaptive Platform.

4.2 Overview

For global recovery System Health Monitor shall be used in addition to PHM and WdgM [6]. The workflow depicted in the figure below clarifies responsibilities between PHM, WdgM and System Health Monitor. PHM and/or WdgM execute Local Supervision (LS) functions and determine Local Supervision Status (LSS) and Global Supervision Status (GSS). Necessary health information is transferred via the payload of existing protocols like SOMEIP to the System Health Monitor. By combining information of one or multiple PHM(s), WdgM(s), SM(s), BswM(s) and/or other System Health Monitoring components, the System Health Monitor determines the Health Indicators. Health Indicators include performance, reliability and subsystem health status. These Health Indicators are transmitted via Health Indicators service fields.

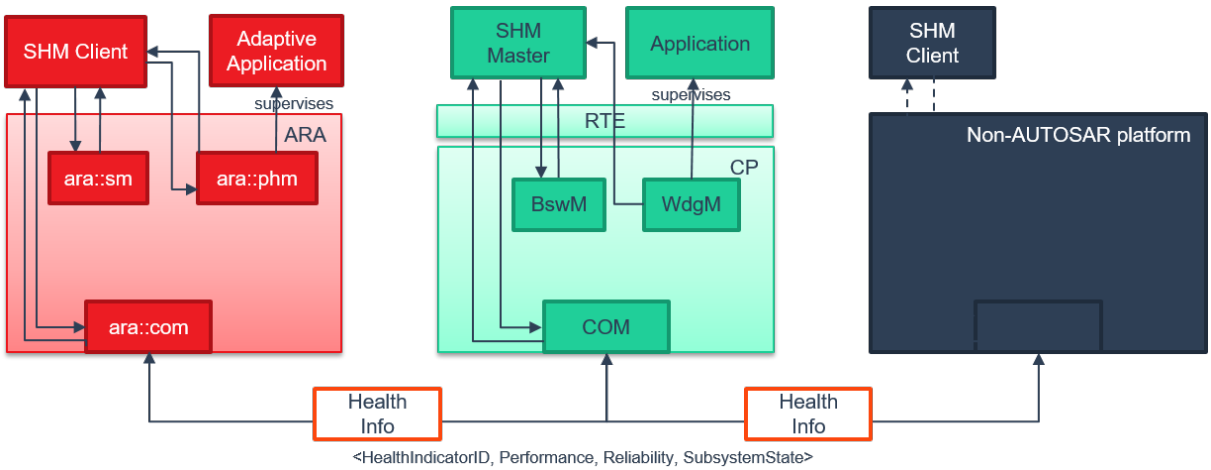

Figure 4.1: Master/Client example across Classic, Adaptive and Non-AUTOSAR platforms.

A master-client distribution is shown in the figure above. In this example the SHM client gathers health related information of the platform, performs some pre-evaluation to build a platform Health Indicator and sends it to the master. The master analyzes the platform HIs and additional supervision results of all its clients to create Health Indicators on a higher abstraction level. Afterwards the Health Indicators are send from the Master to the subscribing clients or other SHM masters. Either SM or applications subscribe to the HI service directly, or the SHM Client can be used to request the HIs instead. This approach can be useful for subscribers which don't support service based communication, so the SHM Client works as a broker. Now, the locally stored HI can be used by platform managers like SM or BswM [7] to degrade the platform. Additionally, the HI can be attached as Health of Service(HoS) to a standardized HI field in services provided by the platform.

The SHM client is responsible for gathering health information within the platform. This health information can then be used to pre-calculate a platform Health Indicator. This platform Health Indicator is send to the SHM master together with the necessary health information (GSS/LSS).The main task of the client is providing this health information as a broker and storing the Health Indicators calculated by the master.

The master calculates the Health Indicators based on all the safety information it receives from all the clients. Clients can subscribe for specific Health Indicators by their IDs. Afterwards the master returns the Health Indicators to all subscribing clients.

System Health Monitor is responsible for determining the Health Indicators. To this end SHM can use software, hardware and context information provided by functional clusters like PHM, WdgM, SM, BswM or other SHM instances. Health Indicators may encapsulate information on Supervision results, Function Group States or Modes, available Hardware information from Health Channel (e.g. voltage monitoring) or system specific knowledge like available redundant components. Depending on the specific project, further abstraction of multiple Health Indicators to define system degradation states or a vehicle wide Health Indicator may be possible.

Health Indicators can be used by PHM or WdgM for more refined health analysis or by SM or BswM to use health knowledge for distributed system level recovery. For Health Indicator determination different approaches like arbitration rules or degradation models can be applied. Less complex scenarios may not require additional logic as Local Supervision Statuses or Global Supervision Statuses and Health Statuses of other platforms would suffice as Health Indicators.

4.3 Redundancy of System Health Monitoring Master

It is needless to say that the System Health Monitoring Master is a very important single point of failure for this concept. Projects should think about redundancy of the SHM Master. The clients might utilize the timeout features provided by E2E to recognize

a failed SHM Master. Similarly a redundant master can detect the failed master by subscribing to the provided HealthIndicators and not receiving them anymore.

There might be different strategies to achieve such a redundancy. One approach could be to have an explicit fallback SHM Master in one of the ECUs in case the originally configured SHM Master fails. The solution is easy to implement and is deterministic, with no need of arbitration to win the SHM Master role. However, the problem with this approach might be that the statically configured fallback SHM Master might not be the best option in run-time, as it might have poor HI than any other ECUs.

Another approach might be to allow an arbitration period after the SHM Master fails. In this arbitration period the client with the best HI will become the SHM Master. This solution would give more confidence that the SHM Master role has been given to the most stable SHM in the vehicle. However, this approach needs all SHMs to have a passive SHM Master logic implemented in them. This approach also means that the System Health Monitoring will be unavailable during the arbitration period. There might also be other criteria, based on which an arbitration could be won.

4.4 Supervision of System Health Monitor

As `System Health Monitor` provides safety-relevant information, functionality of SHM needs to be supervised. As a common implementation option of SHM is in the application layer, supervising SHM with HW Watchdog would not be feasible. Therefore, supervision via PHM/WdgM was decided. Following scenarios were analysed with respect to hierarchical supervision with HW Watchdog.

4.4.1 Scenario 1: One Platform with SHM Deployed on a Single ECU

Depending on platform type (Classic or Adaptive) PHM or WdgM shall monitor SHM functionality by using `Logical Supervision`, `Deadline Supervision` and/or `Alive Supervision`. PHM or WdgM controls the Hardware Watchdog.

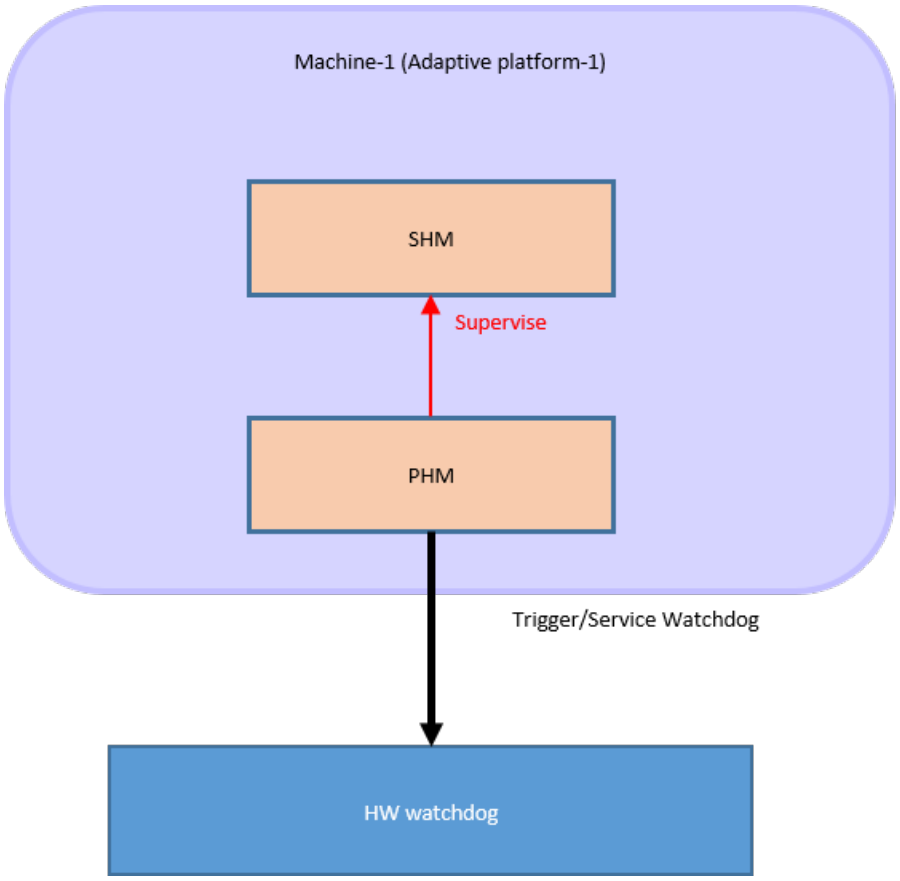

Figure 4.2: SHM for single platform

4.4.2 Scenario 2: Multiple Platforms with SHM

One platform shall monitor SHM using PHM or WdgM. Responsibility between platforms of triggering of Hardware Watchdog shall be configurable. However, if the responsible PHM or WdgM needs information on other platforms, SHM can share information over the `Health Indicator` interface.

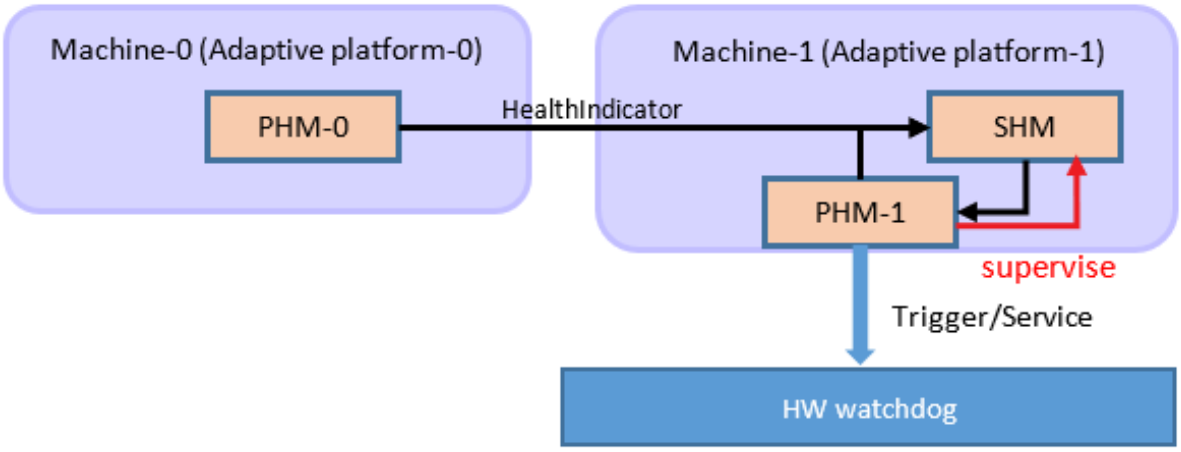

Figure 4.3: SHM for multiple platforms

4.5 Integration of System Health Monitor in AUTOSAR

As currently SHM is only standardized with platform agnostic interfaces, the most sensible solution is implementing SHM on an application layer in AUTOSAR. The standardized interfaces can then be mapped to concrete Classic or Adaptive Interfaces. The platform mapping will be elaborated in later releases. A project-specific solution implementing SHM at System Service layer shall be possible, but not recommended and hence not an AUTOSAR standardized way.

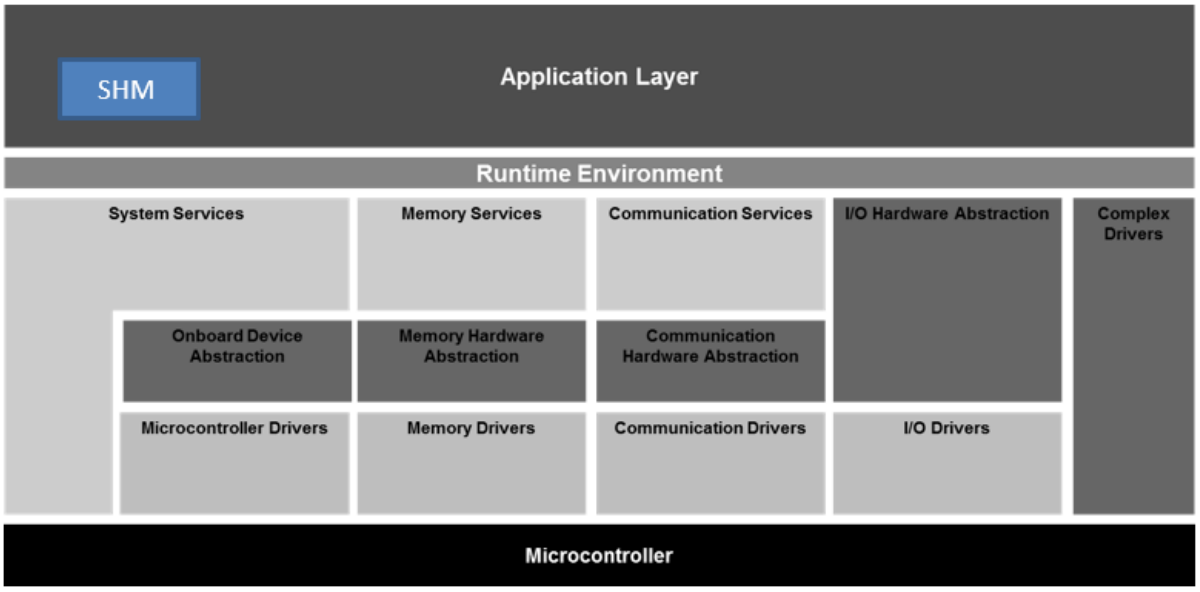

Figure 4.4: Integration of `System Health Monitor` in Classic Platform

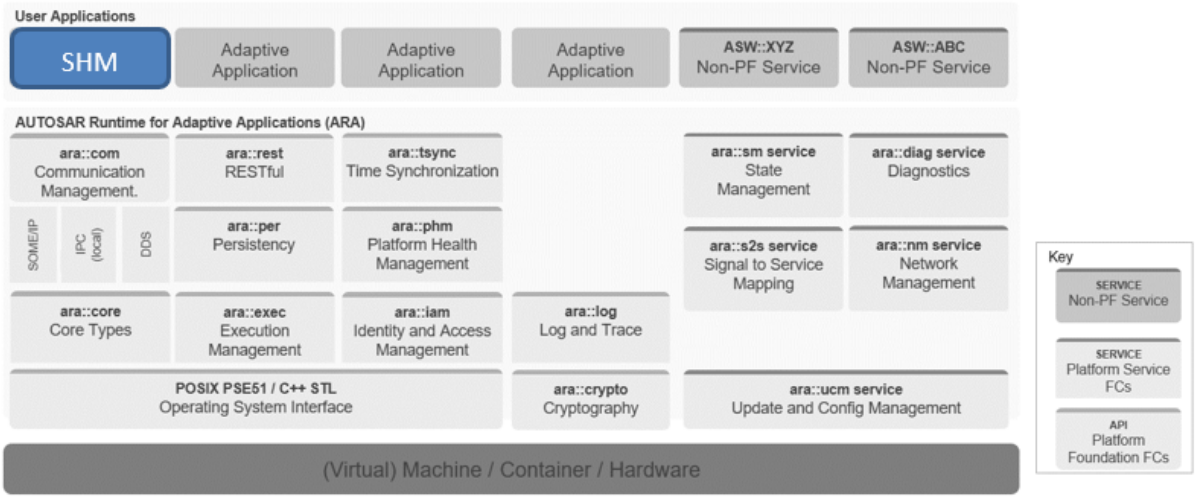

Figure 4.5: Integration of `System Health Monitor` in Adaptive Platform

4.6 Interfaces of System Health Monitor

4.6.1 Overview of System Health Monitor Interfaces

As System Health Monitor shall be standardized as platform agnostic component, all interface descriptions stay on abstract interface descriptions of AUTOSAR_TPS_AbstractPlatformSpecification [8]. All platform interfaces and payload configuration solely provide examples and are not part of the standardization in this release. The examples show existing interfaces that could be used to gather the health information of PHM/SM or WdgM/BswM. The following picture gives an overview of communication with software components of AP and CP. Illustrated Ports of CP and AP are already existing, for AP only the interface of PHM for reporting health channel information to applications will be specified.

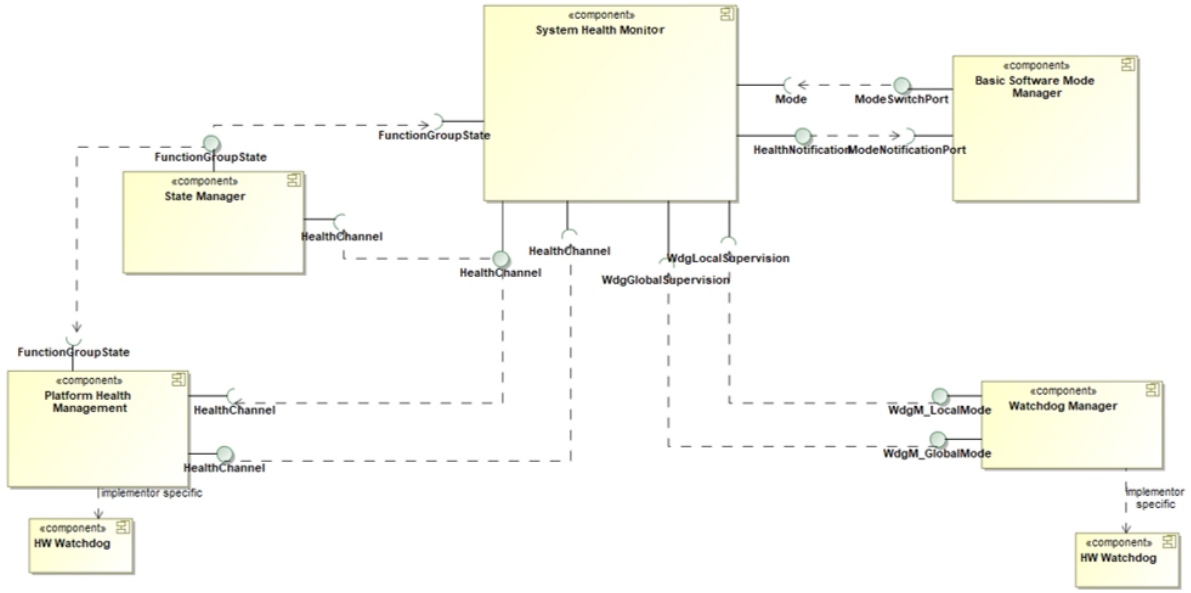

Figure 4.6: Example interfaces of SHM in Classic and Adaptive Platform

4.6.2 Information Transmission to SHM

Information to SHM shall be transmitted as data according to the abstract interfaces of SHM. For communication existing communication modules (ara::com, COM) can be used as well as existing protocols like SOME/IP. As cyclic messaging requires a certain bandwidth, the message content shall be configurable to contain heart beat information, a global degradation state or information on changed supervision parameter. As a very high frequent heart beat will be required to ensure availability, there is no need to introduce additional acyclic messaging.

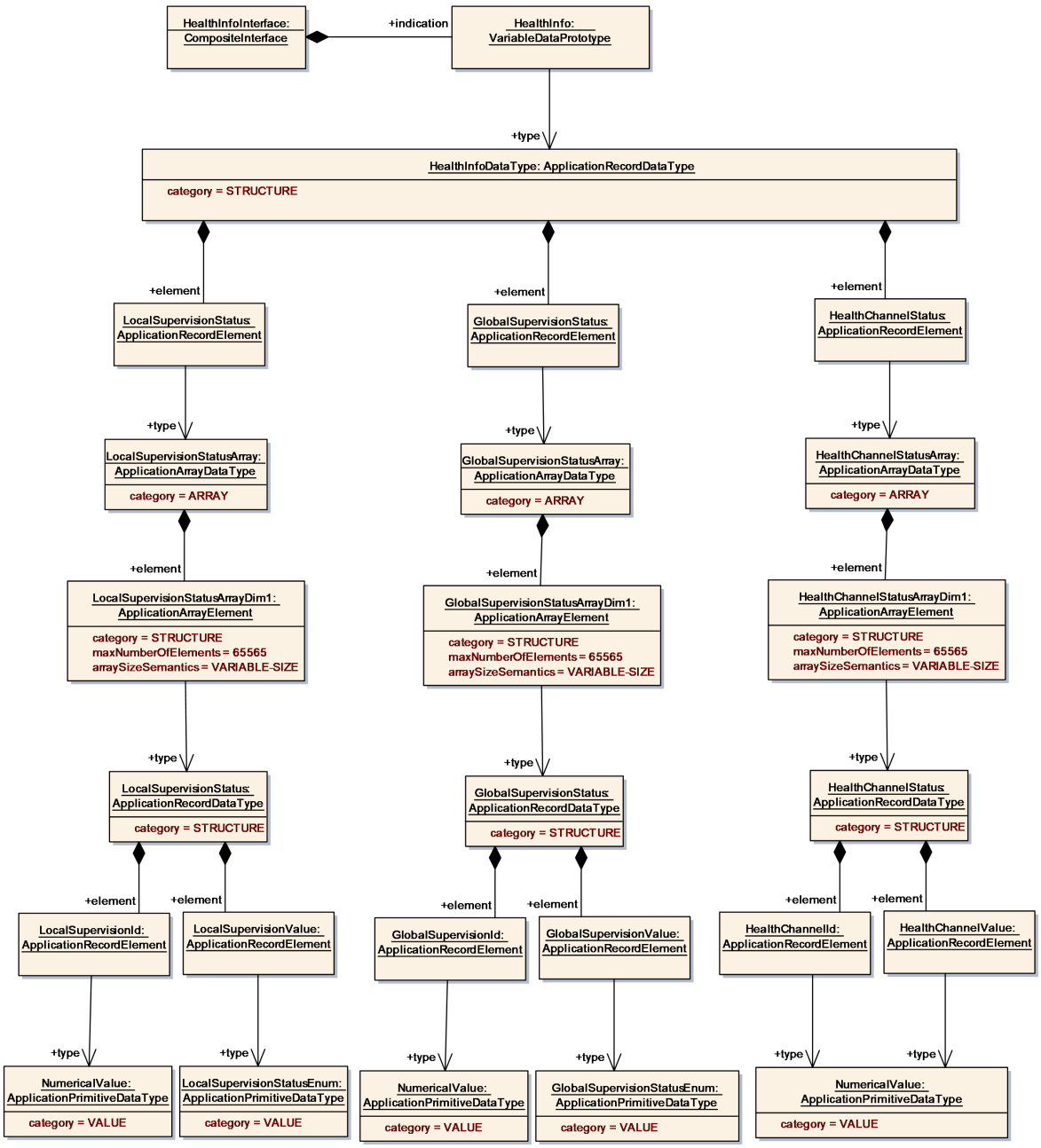

Figure 4.7: SHM abstract datatype modeling

Health information of PHM and WdgM could be described on Abstract Platform level like depicted in figure above. One possible solution to describe the Health Information is to use a Record datatype that contains VariableSizeArrays as subelements for the different payload parts. Each VariableSizeArray inside of the record can have a variable size starting from 0. After the serialization, such a Record datatype may result in a payload structure that is shown in Figure 4.8 where each VariableSizeArrays results in a length field and the array content itself.

As some project specific payload might be send to the SHM as well, TLV or unserialized data could be attached to the rest of the payload. The used payload could be transmitted in a format similar to: <NumberOfElements, ElementType, ElementID, ElementValue>

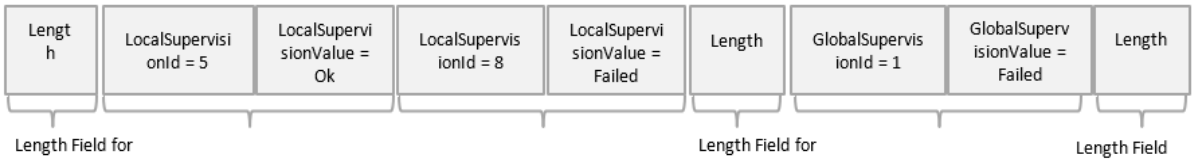

Figure 4.8: Example of an SHM payload

The example in the figure above just shows how the payload could look like when using SomeIP as base protocol. The actual payload description can be found in PRS_SomeIP (Section: Dynamic Length Arrays) [9].

4.6.3 Health Indicator Transmission

For `Health Indicator` transmission a service field shall be specified. This field has the purpose of introducing `Health Indicator` as part of the transmitted service. The `Health Indicator` shall be of structure: <HealthIndicatorID, Performance, Reliability, Subsystem Health Status>.

5 Example Health Indicator and Use Case

5.1 Example Health Indicator: Sensors

This section describes an example calculation for sensors with a definition of all three parameters of the `Health Indicator` (HI) .

A “Sensor application” processes and evaluates the measurements. The application implements voting and plausibility checks to decide which sensor value to forward. Redundancy is applied as a means to improve reliability. Hence, voting and plausibility information is combined with redundancy information to define reliability levels. For the numerical results, the receiver of the HI can define individual threshold values for different levels. The presented equation considers the following constraints: A higher number of voters agreeing with the selected voter increases the reliability; a larger number of implausible voters decreases reliability. As an implausible voter can be ignored, the reliability value shall distinguish the case of one plausible voter disagreeing from the case of one implausible voter disagreeing. On the other hand, a plausible voter disagreeing shall decrease the reliability. Therefore, the number of plausible agreeing voters and the number of implausible voters are the main reliability indicators. Both values are set into ratio with the number of plausible voters and are weighted with factors α and β to tailor the reliability to specific projects:

$$Rel_{Sen} = \alpha * \frac{\#agreeWithVoted}{\#plausibleVoters} - \beta * \frac{\#implausibleVoters}{\#plausibleVoters} \tag{5.1}$$

The “Sensor application” is considered safety-critical and `WdgM` or `PHM` supervises the timely arrival of sensor information and whether logical and deadline constraints are satisfied. The supervision state of the “Sensor application” is used as performance indicator. An error tolerance for failed reference cycles can be configured for `Alive Supervisions`. All supervision results are summed up in the `Local Supervision Status`, which can have one of four states:

- OK: No supervision failed.
- FAILED: An Alive Supervision failed and the error counter is below the configured error tolerance.
- EXPIRED: A Deadline or Logical Supervision failed or the error counter is equal or above the configured error tolerance.
- DEACTIVATED: A mode switch deactivated the Supervised Entity.

Consequentially, OK and DEACTIVATED suggest sensor performance is good (0). Depending on the use case, a delayed sensor input might decrease overall performance but might not be considered a safety risk. Thus, FAILED is mapped to medium performance (1). EXPIRED indicates a severe error or even functionality loss and is mapped to poor performance (2). To summarize the following formula can be stated:

$$Per_{Sen} = \begin{cases} 0 & \text{if } LSS = OK \vee LSS = DEACTIVATED \\ 1 & \text{if } LSS = FAILED \\ 2 & \text{if } LSS = EXPIRED \end{cases}$$

SubsystemState evaluation shall indicate the sensor availability. The minimum requirements for sensor availability demand one plausible sensor with good or medium performance:

$$Sub_{Sen} = \begin{cases} 0 & \text{if } \exists s(Plausible(s) \wedge Per_{Sen}(s) < 2) \\ 1 & \text{else} \end{cases}$$

5.2 Example Use Case: Health of Service (HoS) Indication

As first use case serves the interface between the domains Automated Driving and Chassis. The automated driving domain's task is to provide a safe drivable trajectory in the vehicle's current environment. Motion control takes these trajectories and the current vehicle state into account to transform the driving strategy into commands to control the actors. Therefore, motion control implements additional safety checks. Depending on the active mode, different safety requirements apply. When ADAS is active the main goal is the control quality. In case of poor control quality, the driver assistance functionality is shut down and the driver seamlessly takes over. In contrast, deactivating assistance functionalities is no safe option in FAD and HAD mode. Adequate degradation actions need to be activated to guarantee fail-degraded behavior in case of poor system performance. For this purpose, the trajectory input is checked for plausibility. If an input signal is unknown, safe default values are provided.
Exactly at this step, plausibility checking, the `Health Indicator` of the currently active feature, e.g. highway pilot, can be attached to the sent trajectory. The extensions of the presented motion control platform with the HoS Health Indicator approach are depicted in figure 5.1. This way the `Health Indicator` information provides additional safety information. The Degradation parameter can give an indication to which automation mode the trajectory complies. As described above, for different degradation levels different safety requirements apply. This way, the Health Indicator gives run-time information about which safety criteria to check and the safety requirements can be adapted at run-time. Furthermore, the Reliability and the Performance parameters can be included as part of the plausibility check. To conclude, the safety of the decision whether to keep the trajectory, to deactivate driver assistance or to choose substitute values increases with `Health Indicators` as run-time input.

Using a HealthService with the standardized ServiceHI field for the trajectories allows the attachment of the right HI directly to the service. In case above it is not necessary for the receiver in the Chassis Domain to have knowledge of the architecture and available HIs of the Automated Driving Domain. Especially for receivers who have no direct access to a specific HI, because they have no access to the services of the SHM, don't use an AUTOSAR platform or have knowledge of the senders architecture the HealthService provides a way of transmitting the right HI, which was decided as suitable on the sender side, along the actual data.

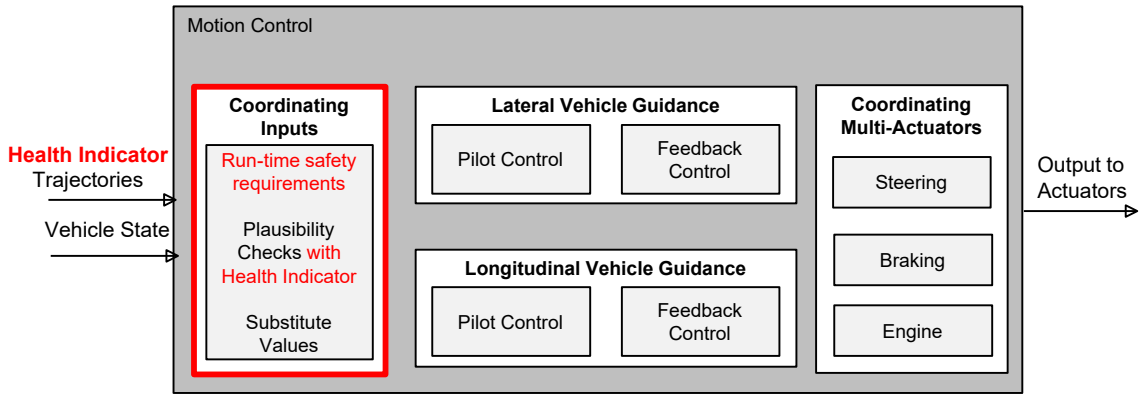

Figure 5.1: Extensions of motion control with HoS Health Indicator

5.3 Example Use Case: System Degradation

This use case demonstrates how `Health Indicators` can support system degradation strategies in automated driving context. As example serves a logical architecture of the automated driving domain as shown in figure 5.2.

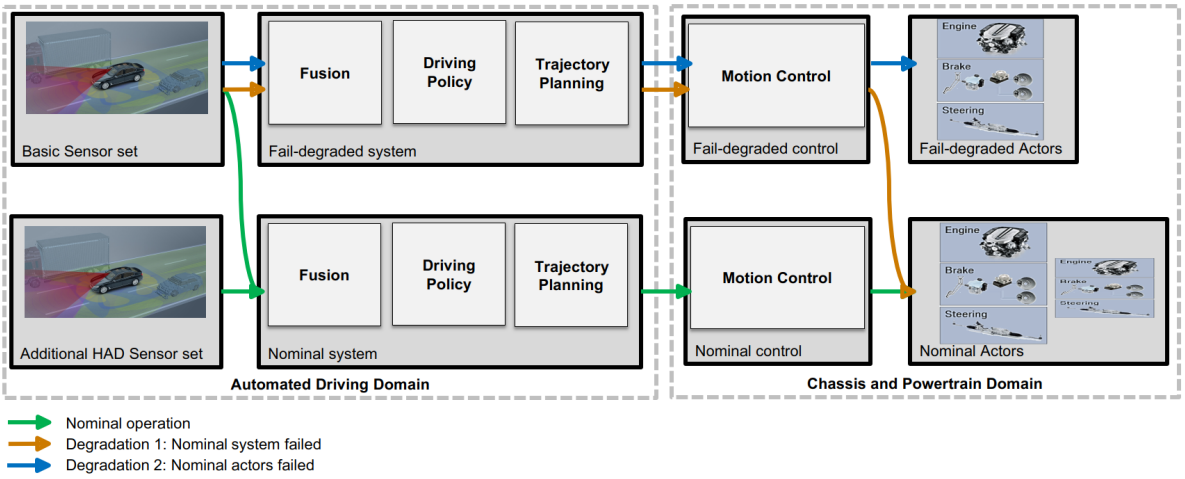

Figure 5.2: System architecture of an automated vehicle

It is realized via two redundant fail-safe systems, the nominal integration system (lower part) and the fail-degraded system (upper part). During operation, only ECUs of the same system are allowed to be active to guarantee that no contradicting commands are transported to braking and steering systems. At all times exactly one system must be active, except when switching between systems. The nominal and fail-degraded system must be independent by design. Otherwise, no fail-degraded capability can be guaranteed. The nominal system is responsible for calculating nominal trajectories, the fail-degraded system determines so-called minimum risk maneuvers.

The SHM collects all information on the nominal and the fail-degraded system, calculates the HIs, and sends them back to the platforms.

For coordinating system degradation, a decentralized approach is applied. Each platform implements its own state machine. Those state machines use Health Indicators as decision criteria to determine the platform's next operation mode. This way the HI completes the heartbeat signal exchanged between the different platforms: In addition to checking platform availability, the HI offers the opportunity for more refined degradation strategies. The state machines determine the platforms next operation state and execute the degradation. To avoid conflicting operation states of different platforms each state machine implements the same adaption logic and all health states are exchanged cyclically. Furthermore, Health Indicators allow identifying and handling reliability and performance shortcomings of the active platform.

Runtime HIs enable realizing performance and functional degradation. While the Performance and Reliability parameters indicate performance shortcomings, the SubsystemState parameter can give context information for the functional degradation.

Either the nominal integration platform or the fail-degraded platform needs to be active. The same criteria apply for nominal and fail-degraded motion control. In case the minimal requirements of one subsystem cannot be fulfilled, this subsystem is marked as inactive and the redundant system is activated. If both redundant systems fail, the driver is the last fallback level. The degradation paths mirror the reduction of functional features from FAD to HAD to ADAS functionality. The nominal integration platform provides FAD and HAD functionality, the fail-degraded platform ADAS functionality.

A Appendix

This chapter is empty