Document Title	Feature Specification of the BSW Architecture and the RTE
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	294
Document Classification	Auxiliary

Document Version	1.1.0
Document Status	Final
Part of Release	4.0
Revision	3

Document Change History

Date	Version	Changed by	Change Description
20.12.2011	1.1.0	AUTOSAR Administration	Corrected wrong usage of term “module short name”
30.11.2009	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Contents

1 Scope of Document 8
2 Conventions to be used 9
3 Requirements Specification 10
3.1 AUTOSAR Scheduler harmonization Concept 11
3.1.1 [BRF00020] Integration of existing BSW Scheduling into the RTE 11
3.1.2 [BRF00260] Support of at Runtime dynamically schedulable BSW Modules 11
3.1.3 [BRF00261] Enable Integrator to optimize Startup/ Shutdown Behavior of BSW Modules 12
3.2 RTE API enhancement Concept 14
3.2.1 [BRF00023] Additional RTE Status ‘Never Received’ 14
3.2.2 [BRF00024] Additional RTE read API using Return Value 14
3.2.3 [BRF00092] Extension of the Receive Queue Behavior 14
3.2.4 [BRF00259] Extend RTE API with Rte_IFeedback 15
3.3 Triggered Event Concept 16
3.3.1 [BRF00031] Triggered Event 16
3.4 Enhance Measurement and Calibration Enabling per-Instance Memory as measurable Concept 17
3.4.1 [BRF00076] Enhance Measurement and Calibration Enabling per-Instance Memory as measurable 17
3.5 Avoidance of duplicated Type Definitions in RTE Types Header File Concept 18
3.5.1 [BRF00078] Avoidance of duplicated Type Definitions in RTE Types Header File 18
3.6 Integrity and Scaling at ports Concept 19
3.6.1 [BRF00101] Self Scaling Signals at Port Interfaces 19
3.6.2 [BRF00097] Conversion between internal and Network Data Types for InterECU Communication 19
3.6.3 [BRF00074] DataSemantics Ranges Check during Runtime 20
3.7 Implicit Communication Enhancement Concept 22
3.7.1 [BRF00079] Optimization of Semantic of implicit Communication due to Resource Need 22
3.8 A2L Generation Support Concept 23
3.8.1 [BRF00021] A2L Generation Support 23
3.9 DEM Behavior Concept 24
3.9.1 [BRF00230] DEM behavior requirement 24
3.9.2 [BRF00231] Forwarding of notifications about new freeze frame data 25
3.9.3 [BRF00232] Control of event handling 25
3.9.4 [BRF00233] Memory Overflow indication 25
3.10 Support of large data types Concept 27
3.10.1 [BRF00004] Support for variable-length Data Types 27
3.10.2 [BRF00005] Support of Data Type Size > 8 Bytes (Signal Size) 27
3.10.3 [BRF00290] Support of Array Type with dynamic Size 28
3.11 VMM AMM Concept 29
3.11.1 [BRF00045] Support Disable ‘normal’ Communication 29
3.11.2 [BRF00060] Control Runtime Changeable LIN Schedule Tables 29
3.11.3 [BRF00073] Port Groups 30

3.11.4 [BRF00103] Control of Mode dependent IPDU Groups 30
3.11.5 [BRF00104] Mode Dependent Reset of Initial Values 31
3.11.6 [BRF00189] Enable SWCs to request dedicated Modes 31
3.11.7 [BRF00190] Configurable BSW internal Evaluation of Mode Requests 31
3.11.8 [BRF00191] Propagation of Mode Information 32
3.12 Fixed Data Exchange Concept 33
3.12.1 [BRF00105] System Parameter 33
3.12.2 [BRF00157] Fixed Values for R-Ports 33
3.13 Variant Handling Concept 35
3.13.1 [BRF00029] Variant Handling on VFB Level 35
3.13.2 [BRF00155] Support for different Interfaces of a SW-C 35
3.13.3 [BRF00167] Macro Value to set Variant 36
3.14 Bus Monitoring Issues Concept 37
3.14.1 [BRF00087] API to query the FlexRay Rate and Offset Correction 37
3.14.2 [BRF00093] Detection of missed FlexRay Startup Frames 37
3.14.3 [BRF00264] FlexRay Transceiver Errors reported for Diagnostics 38
3.14.4 [BRF00265] Reporting Applied FlexRay Clock Correction Terms 38
3.14.5 [BRF00266] Report of Aggregated FlexRay Channel Status Error 39
3.14.6 [BRF00267] Report of FlexRay Status Data for number of Sync Frames. 39
3.14.7 [BRF00299] Report of FlexRay Status Data for list of IDs 39
3.15 Support of SAE J1939 Protocol Features Concept 41
3.15.1 [BRF00168] Support of SAE J1939 Protocol Features 41
3.16 LIN 2.1 Std Concept 43
3.16.1 [BRF00184] Adaptation of the LIN Stack at LIN 2.1 Specification 43
3.17 FlexRay ISO TP Concept 44
3.17.1 [BRF00192] Support of FlexRay Message IDs 44
3.17.2 [BRF00252] ISO 10681-2 conform FlexRay Communication 44
3.18 LIN Transceiver Driver 46
3.18.1 [BRF00228] LIN Transceiver to be specified 46
3.19 FlexRay Protocol Spec Issues Concept 47
3.19.1 [BRF00268] Support for FlexRay Single Slot Mode 47
3.19.2 [BRF00273] Support for Dual FlexRay Channels 48
3.20 FlexRay Spec 3.0 Concept 50
3.20.1 [BRF00272] Support for active FlexRay Stars 50
3.20.2 [BRF00277] Support of FlexRay Specifications 3.0 50
3.21 TCP/IP CommStack Externsions Concept 51
3.21.1 [BRF00283] Enable BSW to communicate via TCP/IP 51
3.21.2 [BRF00284] Enable the Applications to communicate via TCP/IP 51
3.21.3 [BRF00285] Enable the PDU Router to communicate via TCP/IP 51
3.21.4 [BRF00286] Support Ethernet as an additional communication medium .. 52
3.21.5 [BRF00287] Implementation of Diagnostic Communication over IP 52
3.22 Time Determinism Concept 55
3.22.1 [BRF00120] Provision of a synchronized time-base within a cluster 55
3.22.2 [BRF00121] Runtime timing protection and monitoring 55
3.22.3 [BRF00122] Support for timing constraints 56
3.22.4 [BRF00123] Responsiveness to external events 56
3.22.5 [BRF00125] Monitoring of local time 56
3.22.6 [BRF00126] Services for synchronization of SW-Cs 57
3.22.7 [BRF00127] Services for accessing to synchronized time-bases 57

3.22.8 [BRF00278] Sync AUTOSAR OS with Global Time from providing bus system in a well-defined way ... 58
3.23 XCP for AUTOSAR Concept ... 59
3.23.1 [BRF00279] Trigger Configuration of FR CC Buffer with non-static Configuration ... 59
3.23.2 [BRF00280] AUTOSAR BSW XCP Modules ... 59
3.24 NM Coordination Concept ... 61
3.24.1 [BRF00256] NM Coordinator should support coordination to any kind of AUTOSAR busses ... 61
3.24.2 [BRF00271] NM Coordinator should support NM Gateway to FlexRay ... 61
3.24.3 [BRF00274] FlexRay Network Management Scheduling Timing Window Relief ... 62
3.25 Functional Diagnostics of SWC Concept ... 63
3.25.1 [BRF00027] Functional Diagnostics of SWC ... 63
3.25.2 [BRF00229] Decentralized modular diagnostic configuration of SW-Cs ... 63
3.26 FlexRay Network Reliability Concept ... 65
3.26.1 [BRF00302] FlexRay Transmission Completion Confirmation ... 65
3.26.2 [BRF00303] FlexRay Transmission Timeout Handling ... 65
3.26.3 [BRF00304] FlexRay Reception Completion Confirmation ... 65
3.26.4 [BRF00305] FlexRay Payload Length Check ... 66
3.26.5 [BRF00306] FlexRay Hardware Check ... 66
3.26.6 [BRF00307] FlexRay Reset/Reinitialization ... 66
3.27 TTCAN Concept ... 68
3.27.1 [BRF00312] Introduction of TTCAN into AUTOSAR [accepted] ... 68
3.28 Debugging Concept ... 69
3.28.1 [BRF00152] BSW Variables becomes accessible by external Debuggers ... 69
3.28.2 [BRF00083] ORTI comparable XML Module Description ... 69
3.28.3 [BRF00084] Debugging-Extension of the M2 Meta Model ... 70
3.28.4 [BRF00085] Access to PduR for Debugging ... 70
3.29 DLT Concept ... 71
3.29.1 [BRF00224] Allow monitoring of DEM, RTE and COM to improve diagnostics ... 71
3.29.2 [BRF00294] Standardized Log&Trace format/protocol ... 71
3.29.3 [BRF00295] DET Trace interface for Log&Trace ... 72
3.29.4 [BRF00296] RTE/VFB Trace interface needed for Log&Trace ... 72
3.29.5 [BRF00297] DEM Trace interface needed for Log&Trace ... 73
3.29.6 [BRF00298] Log&Trace debug interface using diagnostic service ... 73
3.29.7 [BRF00300] Standardized interface/service Log&Trace for SWC ... 74
3.30 Memory related Concepts ... 76
3.30.1 [BRF00022] Modification of NVRAM Memory Access Concept ... 76
3.31 Build System Enhancement Concept ... 77
3.31.1 [BRF00057] Memory Mapping Concept ... 77
3.31.2 [BRF00077] Memory Mapping of SWCs ... 77
3.32 Support of Windowed Watchdog Concept ... 78
3.32.1 [BRF00159] Support of Windowed Wachdog ... 78
3.33 Alarm Clock Concept ... 79
3.33.1 [BRF00196] Alarm Clock ... 79
3.34 Enabling CDDs in the BSW architecture Concept ... 80
3.34.1 [BRF00225] Enabling CDDs in the BSW architecture ... 80
3.35 Concept for Libraries ... 81

3.35.1 [BRF00165] Integration of the HIS Crypto Functionality 81
3.35.2 [BRF00311] Standard AUTOSAR libraries 81
3.36 Bootloader Interaction Concept 82
3.36.1 [BRF00034] Bootloader Interaction 82
3.36.2 [BRF00262] DCM shall support the Service EcuReset 82
3.36.3 [BRF00263] DCM shall internally support the Jump to Flash-Bootloader 83
3.37 Multi Core Architectures Concept 84
3.37.1 [BRF00199] Real-time Capability & Predictability 84
3.37.2 [BRF00200] Deadlock free mutual Exclusion 84
3.37.3 [BRF00204] Multi Core System with one Core as intelligent Peripheral 84
3.37.4 [BRF00205] Multi Core System with one OS per Core 85
3.37.5 [BRF00206] Multi Core System with one OS controlling all Cores 85
3.37.6 [BRF00207] Freedom from Deadlocks 86
3.37.7 [BRF00208] Freedom from unbounded Blocking 86
3.37.8 [BRF00209] Service Compatibility to single Core Systems on one Core 86
3.37.9 [BRF00210] High Service Compatibility to single Core Systems across multiple Cores 87
3.37.10 [BRF00211] Static Assignment of Tasks to Cores 87
3.37.11 [BRF00212] Activation of Tasks across Cores 88
3.37.12 [BRF00214] Resources across Cores 88
3.37.13 [BRF00215] Static Assignment of Interrupts to Cores 89
3.37.14 [BRF00216] Disabling/Enabling Interrupt API Calls work locally 89
3.37.15 [BRF00217] Event Mechanism shall work across Cores 89
3.37.16 [BRF00218] Offline Configurability of Number of Cores 90
3.37.17 [BRF00220] Initialization and Startup 90
3.37.18 [BRF00221] Controlled Data Exchange between Cores 90
3.37.19 [BRF00222] Common Configuration 91
3.37.20 [BRF00223] Inter Core Timer Synchronization 91
3.38 Error Handling Concept 92
3.38.1 [BRF00156] Specification of the Error Handling 92
3.38.2 [BRF00193] List of Standardized Errors 92
3.38.3 [BRF00275] Error Handling Capabilities for Partitions 93
3.39 SRS Core Test 95
3.39.1 [BRF00185] Test modes 95
3.40 SRS Flash Test 96
3.40.1 [BRF00186] Test Result Processing 96
3.41 SW-C E2E Communication Protection Concept 97
3.41.1 [BRF00114] SW-C end-to-end communication protection 97
3.42 Memory Partitioning Concept 98
3.42.1 [BRF00115] SW-Cs grouped in separate user-mode memory partitions 98
3.43 Program flow monitoring Concept 100
3.43.1 [BRF00131] Logical Program Flow Monitoring 100
3.44 BSWM Defensive behavior Concept 101
3.44.1 [BRF00128] Protection against Unauthorized Use of BSW 101
3.44.2 [BRF00129] Protection of Data 101
3.45 Communication Stack Concept 102
3.45.1 [BRF00110] Protection of Communication 102
3.45.2 [BRF00111] Data Sequence Control 102
3.45.3 [BRF00112] Routing Integrity 103
3.45.4 [BRF00113] Communication Watchdog 103

3.45.5 [BRF00241] Multiple Communication Links 103
3.45.6 [BRF00242] Network Communication Monitoring 104
3.46 E-Gas Monitoring Applicability Concept 105
3.46.1 [BRF00301] Ability to make an AUTOSAR application compatible to the e-Gas monitoring Concept 105
3.46.2 [BRF00248] Testing and monitoring of I/O data and I/O HW 105
3.46.3 [BRF00251] Priority Access to SPI BUS 105
3.46.4 [BRF00243] Communication protections against corruption and loss of data 106
3.47 Configuration related features 107
3.47.1 [BRF00042] Use of XML instead of OIL for Configuration of OS 107

1 Scope of Document

This document currently describes all **new** features of the AUTOSAR Basic Software (BSW) and the RTE.

All new features in Release 4.0 will be described, which extend the Release 3.1 of the BSW and the RTE.

The features are grouped together according to “Concepts” which collect related features for a special purpose.

2 Conventions to be used

In requirements, the following specific semantics shall be used (based on the Internet Engineering Task Force IETF).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as:

- SHALL: This word means that the definition is an absolute requirement of the specification.
- SHALL NOT: This phrase means that the definition is an absolute prohibition of the specification.
- MUST: This word means that the definition is an absolute requirement of the specification due to legal issues.
- MUST NOT: This phrase means that the definition is an absolute prohibition of the specification due to legal constraints.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

3 Requirements Specification

In this chapter a structure of a requirements document is given. The structure is strongly related to Basic Software Module requirements specifications (SRS documents).

3.1 AUTOSAR Scheduler harmonization Concept

3.1.1 [BRF00020] Integration of existing BSW Scheduling into the RTE

ID:	BRF00020
Initiator:	AUTOSAR PL
Date:	03.05.2007
Short Description:	Integration of existing BSW scheduling into the RTE
Importance:	**high** | medium | low
Description:	Investigate and implement means to handle the scheduling of Application SW and Basic SW using the RTE.
Rationale:	The Release 2 of AUTOSAR (i.e. result of phase 1) includes 2 modules implementing scheduling aspects, on the basis of the mechanisms provided by the AUTOSAR OS. These modules are the RTE, for SWC scheduling, and the BSW Scheduling Module for the BSW scheduling. The judgment is that there is a large possibility for optimization in this area, by integrating the BSW scheduling into the RTE altogether, thus providing the RTE with the possibility to generate a more optimal scheduling scheme, taking into account the complete ECU SW needs. This would result in the removal of the BSW Scheduling Module from the BSW architecture.
Use Case:	Application SW calling BSW Service NvM_Read(). This call is accessing some job queue for adding the job. The NvM main function is also accessing the job queue for emptying it. If there are no precautions taken this may lead to concurrent access to the job queue and inconsistent state. If both software is scheduled from the same entity (RTE) such conflicting access can be detected and protected.
Dependencies:	--
Conflicts:	--
Supporting Material:	This task shall therefore elaborate the RTE specification to also cover the BSW scheduling (i.e. task mapping, concurrency protection, event handling). If, during this work, potential improvements of the existing solutions for SWC scheduling are detected, implementation of these improvements shall be considered. By removing the BSW scheduler, optimization can be done on scheduling and data consistency between SWc and BSW. (same as exclusive area in SWC) On another hand, the performances of the BSW could be impacted (interrupt masking time, glue code generated) whereas today it is written by and therefore optimized. The RTE might support local interrupt enabling / disabling (example CAN ISR, Lin, Etc.) The RTE can also generate empty function to be coded by ECU Integrator.
Contributes to:	--

3.1.2 [BRF00260] Support of at Runtime dynamically schedulable BSW Modules

ID:	BRF00260
Initiator:	--
Date:	31.01.2008
Short Description:	Support of at runtime dynamically schedulable BSW modules

Importance:	medium
Description:	Support of multiple BSW modes with different sets of active functionality by dynamically enabling or disabling the scheduling of BSW modules. The AUTOSAR BSW shall support multiple modes with different sets of BSW functionality that is available per mode.
Rationale:	The full functionality of the BSW might not always be required. For example, in a low power mode, the CPU of the ECU is clocked with a lower frequency while not all BSW functionality is required. To improve responsiveness of the remaining BSW functionality, a mechanism to define active BSW functionality based on modes is required. The active set of BSW functionality would then be defined by the BSW modules that are scheduled for the current mode while the functionality of the BSW modules that are not scheduled in the current mode is not available.
Use Case:	- Telematic ECU that periodically checks GPS position and environment (e.g. temperature) and if necessary communicates with telematic central computer (via GPRS/UMTS) or wakes up the bus for communication with other ECUs. - For the periodic checks, the whole BSW functionality (e.g. bus communication) is not required.
Dependencies:	Other mode concepts - VMM/AMM - Mode dependent communication - AUTOSAR Scheduler Harmonized
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.1.3 [BRF00261] Enable Integrator to optimize Startup/ Shutdown Behavior of BSW Modules

ID:	BRF00261
Initiator:	AUTOSAR WP Software Architecture and OS
Date:	31.01.2008
Short Description:	Implementation of a framework mechanism to enable the integrator to optimize the startup/shutdown behavior of the BSW according to specific needs of an integrator.
Importance:	medium
Description:	The R3.0 startup/shutdown procedure defines three initialization blocks with fixed activities between them. With the increasing functional dependency between BSW modules in R4.0, it will be more complex for the integrator to ensure correct and collision-free startup/shutdown procedures for the BSW modules. Additionally, these interdependencies might introduce additional delays (e.g. a BSW module waiting for startup data from NVRAM). AUTOSAR therefore shall support the integrator at: - Resolving startup/shutdown conflicts between BSW modules and - Optimizing startup/shutdown procedures according to specific needs (e.g. minimum time) A framework mechanism based on BSW modes that are cycled through during startup and shutdown is a possible approach.
Rationale:	Speed and correctness of startup/shutdown of the AUTOSAR BSW depends heavily on the integrator. Support by the AUTOSAR standard to fulfill this

	task is desirable.
Use Case:	- Integration of a body-comfort-ECU with many SW-Cs and full-blown BSW requiring almost instant startup
Dependencies:	Other mode concepts - VMM/AMM - Mode dependent communication
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.2 RTE API enhancement Concept

3.2.1 [BRF00023] Additional RTE Status 'Never Received'

ID:	BRF00023
Initiator:	AUTOSAR WP VFB and RTE
Date:	21.06.2007
Short Description:	Additional RTE status "never received"
Importance:	High | **medium** | low
Description:	Add an additional optional RTE-Status "never received". This is the new initial status of each data element for which it is configured. This initial status will be cleared when the first reception occurs.
Rationale:	This additional optional status establishes the possibility to check, whether a data element has been changed since system start.
Use Case:	Get the information whether involved data have been received at any time since system start.
Dependencies:	--
Conflicts:	--
Supporting Material:	It is not clear yet whether it concerns end to end communication or sender-side COM to SWC.
	--
Contributes to:	--

3.2.2 [BRF00024] Additional RTE read API using Return Value

ID:	BRF00024
Initiator:	Continental
Date:	03.05.2007
Short Description:	Additional RTE read API using return value
Importance:	high | medium | **low**
Description:	Investigate whether an additional RTE-Read API to return value as result (without rte_status) does provide any advantage from an efficiency point of view. If there is a benefit it shall be included in the specification.
Rationale:	When the RTE_Read() caller is not interested in the status of the received information it may be more efficient to use an API which returns the result as return value.
Use Case:	Many calls to RTE_Read() are expected for Application SWCs. Allow the RTE to generate efficient code which does not need SPECIAL code optimization compilers to benefit from.
Dependencies:	It has to be investigated whether current "embedded" compilers are able to generate efficient code regardless whether the value is returned directly (as proposed here) or the status result is return BUT NOT used.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.2.3 [BRF00092] Extension of the Receive Queue Behavior

ID:	BRF00092
Initiator:	Daimler
Date:	29.05.2007

Short Description:	Extension of the Receive Queue Behavior
Importance:	high | medium | **low**
Description:	The receive queue of the RTE shall be changed to become able to act as a poll able receiver buffer which indicates a reception since the last poll. Therefore the receive buffer shall be able to overwrite the oldest signal in case of an overflow. The current behavior rejects received signals in case of an overflow.
Rationale:	Up to now it is only possible to get a receive indication after receiving a signal while for some applications polling is more feasible. Therefore a receive buffer of size 1 is applicable which overwrites the content in case of further received signals. If the application gets the signal it polls, this is the indication of a reception implicitly, since in case of an empty queue this situation will be indicated.
Use Case:	SWC polls asynchronously (to the FlexRay cycle) the value of a signal and get information, if this signal has not been received since the last poll.
Dependencies:	--
Conflicts:	--
Supporting Material:	Similar the queue handling in RTE (in case of queue size 1)
Contributes to:	--

3.2.4 [BRF00259] Extend RTE API with Rte_IFeedback

ID:	BRF00259
Initiator:	Daimler
Date:	31.01.2008
Short Description:	Extend RTE API with Rte_IFeedback
Importance:	high | **medium** | low
Description:	The current R3.0 RTE SWS does only provide an Rte_Feedback API to query the state of the transmit status for explicit communication. This shall also be available for implicit communication where the runnable entity can query whether the data provided in a previous execution has actually been transmitted.
Rationale:	Allow to query the transmission state also for implicit communication.
Use Case:	A runnable entity shall be able to check whether the information provided from the last execution has actually been transmitted.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.3 Triggered Event Concept

3.3.1 [BRF00031] Triggered Event

ID:	BRF00031
Initiator:	Continental
Date:	03.05.2007
Short Description:	Triggered events
Importance:	**high** | medium | low
Description:	The SWS_RTE defines 7 kinds of events but it does not correctly support recurrent but not timing based events (e.g. crankshaft event in Power train application). The already existing TimingEvent seems to be really relevant to time based event. For a matter of fact, TimingEvent has a "period" propriety. These recurrent events shall trigger a set of category 1 runnables like timing events do, i.e. trigger a corresponding task then runnables are executed in a given order. These events could be generated by BSW (for sure) or by SWC (to be discussed). A global RecurrentEvent could be defined. Already existing TimingEvent is a specialization of RecurrentEvent.
Rationale:	Synchronous and asynchronous triggering of Runnables based on sporadic and non timing based recurrences, for time-critical application (e.g. Powertrain)
Use Case:	– Angle Synchronous Processing in a combustion engine to control a combustion engine several processes must be synchronized with the crank shaft and cam shaft position. – Angle periodic triggering of processes for instance calculation of the Mass Air Flow of a combustion engine. The specific nature of such processing is comparable to timing events but the periodicity is defined by the engine speed. – Sporadic triggering of processes at crankshaft determination related events for instance first valid detection of a tooth in the crankshaft signal. – The occurrence of such events at maximum engine speed is very high and the required delay and jitter for the triggered runnable entities is very low. Due to that a coupling with the communication system of the ECU shall be strongly avoided! – Such events are expected to be handled locally in one ECU always.
Dependencies:	--
Conflicts:	--
Supporting Material:	Camshaft-dependant events: Hardware may provide a set of events, may act like a data received event triggering a task with the assigned ordered runnables.
Contributes to:	--

3.4 Enhance Measurement and Calibration Enabling per-Instance Memory as measurable Concept

3.4.1 [BRF00076] Enhance Measurement and Calibration Enabling per-Instance Memory as measurable

ID:	BRF00076
Initiator:	Continental
Date:	14.06.2007
Short Description:	Enhance Measurement and calibration enabling Per-instance Memory as measurable
Importance:	high | **medium** | low
Description:	Provide RTE support for measurement of per-instance memory. PIM is provided by RTE, therefore this support can only be provided by RTE.
Rationale:	Providing access to per instance memory by measurement tooling.
Use Case:	PIM can be used for storing internal intermediate computations and an access to PIM is required for measurement. Due to the usage of PIM as RAMBuffer of NVRam Memory, the PIM can contain data which is modified during the execution of functionality (adjustment of relative positions counters or sensor deterioration).
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.5 Avoidance of duplicated Type Definitions in RTE Types Header File Concept

3.5.1 [BRF00078] Avoidance of duplicated Type Definitions in RTE Types Header File

ID:	BRF00078
Initiator:	Continental
Date:	14.06.2007
Short Description:	Avoidance of duplicated type definitions in RTE types header file
Importance:	high | medium | **low**
Description:	The current RTE specification states no rules in the case of types defined with the same name contained in RTE generator's input. A) But the AUTOSAR Software Component Template allows the usage of compatible data types for AUTOSAR Interfaces without restriction of type naming. In one case the types are compatible and named identical. B) There are two types with same name but different semantic defined which is possible by AUTOSAR Software Component Template and difficulty to avoid, if several companies working on same vehicle system.
Rationale:	Achieve higher code quality by avoidance of unnecessary compiler warnings. Avoidance of name space conflicts in C implementations.
Use Case:	Functional structuring of system description in fully-defined Sub-Packages under separate Version Management. Definition of a data converter component recalculating data from one range to an other with identical named types.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.6 Integrity and Scaling at ports Concept

3.6.1 [BRF00101] Self Scaling Signals at Port Interfaces

ID:	BRF00101
Initiator:	Valeo
Date:	25.06.2007
Short Description:	Self Scaling Port Interfaces
Importance:	high | **medium** | low
Description:	Allow the connection of ports with incompatible interfaces : – Incompatible because the data type and or the data semantics are not compatible according to the compatibility rules of the SWC-T. – Data semantics is important to mention because the type might be equal (e.g. UInt16), but the offset and scaling can be different. Therefore RTE shall allow automatic re-scaling of signals. The RTE shall allow specifying the scaling of signals for ports on the sender/server side and the receiver/client side to allow automatic re-scaling in the RTE. The following signal attributes shall be specified – Resolution – physical units – provided/required update rate (periodic / on event) → consider network/transmission update rates in between Connector needs to be modeled on VFB level, and based on that RTE generator has to create the adapter. The RTE generator is not allowed to do it completely on its own.
Rationale:	Avoid writing SWC glue to interface two different SWCs conversion code provided by the integrator / sub-system designer (e.g. using a COMPU-METHOD). Hence, no recalculation of signal resolution in the affected SWCs is necessary.
Use Case:	Integration of SWCs – RTE generate scaling changes (C cast) to interface different SWCs – Avoid to write SWC glue to interface 2 different SWCs In diagnostics the resolution of signals has to be provided as specified in the ISO document for OBDII (e.g. engine speed). If the RTE could provide these signals in the correct resolution the SWC is not required to do this re-calculation.
Dependencies:	--
Conflicts:	--
Supporting Material:	system template has to be modified
Contributes to:	--

3.6.2 [BRF00097] Conversion between internal and Network Data Types for InterECU Communication

ID:	BRF00097
Initiator:	AUTOSAR WP Methodology and Configuration
Date:	21.06.2007 (FMC)
Short Description:	Conversion between internal and network data types for InterECU Communication

Importance:	high | medium | **low**
Description:	Be able to specify, and generate conversion routines for, different data types for data elements used in sender-receiver communication within one ECU (high resolution) and data elements used between ECUs (low resolution) to save serial data link bandwidth and non-productive development work. Specify one data type for data elements with high resolution, e.g. real32, int32, int16, used in sender-receiver communication between SWC's within one ECU. Another data type with lower resolution specified in inter-ECU sender-receiver communication between SWCs mainly to save data link bandwidth. The relationship between the two different data types can be described using data type semantics defined in the Software Component Template. This relationship can be an attribute e.g. in the communication specification of the assembly connection (also defined in the Software Component Template) that connects the two SWC on two different ECUs. An alternative is to specify the "fallback" data type to be used in inter-ECU communication already in the interface type definition. This way the information is always present even if you chose to move one SWC into the same ECU so that the communication becomes intra-ECU. The conversion routines, from high-to-low resolution and low-to-high resolution can be generated and used in RTE or COM interface functions.
Rationale:	Today you must use the same data type for a data element regardless if the data element is used in communication within one ECU or between two ECUs. This restricts the resolution you can use for the data element. One work around today is to create a special communication SWC that takes care of these conversions. This is a non-productive work and you can already describe the relation between the data types of high and low resolution in the SWC Template using data type semantics. The conversion routines between the data types of different resolution can be generated by the RTE or COM configuration generator.
Use Case:	Within one ECU the SWC's want to use data types with high resolutions in computations to get a small epsilon (computational deviation). The developer specifies which data type is to be used if/when the data element is sent over the serial communication bus. On the receiving SWC on the other ECU the data type of the data element is converted to the data type of high resolution. This way we do not need to introduce special communication SWC on each ECU that handles the necessary conversions. The use case that this derives from is used a LOT in the ECUs at Volvo Powertrain and good support in RTE or COM would enhance the efficiency in the development of ECU SW. Today we have are using a special communication component that converts between the data types.
Dependencies:	Software Component Template, RTE and/or COM configuration generator.
Conflicts:	The SWC does not see explicit that the data type resolution becomes lower when the data is sent between two ECUs instead of within one ECU. This should be marked in the software composition diagram (part of the Software Component Template). You must take a look into the communication specification for the assembly connection that connects the two SWC residing on two different ECUs. The data type resolution for data sent within one ECU is always the one chosen in the interface type definition.
Supporting Material:	--
Contributes to:	--

3.6.3 [BRF00074] DataSemantics Ranges Check during Runtime

ID:	BRF00074

Initiator:	Daimler
Date:	11.06.2007
Short Description:	DataSemantics Ranges Check during runtime
Importance:	high | **medium** | low
Description:	Provide functionality in the RTE to enable checking the ranges of data elements values for both S/R and C/S. This checking shall be configurable. Default is OFF.
Rationale:	For debugging it is useful to check whether SWCs sending data to provide the data within the specified ranges. In case there is a violation a Development Error should be raised.
Use Case:	– Check the range of send data during development phase. Disable this feature for production code. – Ease SWC integration.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.7 Implicit Communication Enhancement Concept

3.7.1 [BRF00079] Optimization of Semantic of implicit Communication due to Resource Need

ID:	BRF00079
Initiator:	Continental
Date:	14.06.2007
Short Description:	Optimization of semantic of implicit communication due to resource need
Importance:	**High** | medium | low
Description:	Optimize buffers usage for S/R implicit communication when cooperative runnable placement strategy with non pre-emptive tasks can be applied. Currently the semantic of implicit communication is limited to Copy strategy which needs at least 1 + (Number of task priorities in which the data is used) buffers per data. In non preemptive environments the copy of data can be avoided. In addition a higher flexibility shall be defined in which point of time the data are received or sent. The data consistency is also guaranteed, if data is received before it is consumed first time or if it is sent after it is produced. Therefore a hard limitation to task start or task termination is not required!
Rationale:	Optimize the implicit communication to respect the limited resource and hard real time constrains of automotive embedded systems.
Use Case:	Runnable placement strategy can reduce task specific buffer usage for S/R implicit communication.
Dependencies:	--
Conflicts:	--
Supporting Material:	The RTE SWS (R2.1) describes in general the mechanisms for data consistencies. – Sequential scheduling strategy – Interrupt blocking strategy – Usage of OS resources – Task blocking strategy – Cooperative runnable placement strategy – Copy strategy Rejected because of RTE subgroup decision ()
Contributes to:	--

3.8 A2L Generation Support Concept

3.8.1 [BRF00021] A2L Generation Support

ID:	BRF00021
Initiator:	AUTOSAR WP Software Architecture and OS
Date:	03.05.2007
Short Description:	A2L generation support
Importance:	**high** | medium | low
Description:	The RTE generator has to generate a XML file defining mapping between dataelements / Calprm / CalprmElementGroup and RTE variables to be used by A2L generators
Rationale:	Support of A2L file generation
Use Case:	Support of A2L file generation
Dependencies:	Might need a SWC-T update / upgrade
Conflicts:	--
Supporting Material:	This XML file is one input for subsequent tools generating the A2L file
Contributes to:	--

3.9 DEM Behavior Concept

3.9.1 [BRF00230] DEM behavior requirement

ID:	BRF00230
Initiator:	Daimler
Date:	25.01.2008
Short Description:	DEM behavior requirement
Importance:	high | **medium** | low
Description:	Harmonize DEM behavior requirements to allow for common fault manager (DEM) implementation Harmonize requirements for – storage of DTCs, – status handling and supported statuses – environmental data handling (event log vs. data update) – DTC prioritization – DTC self-healing – overwriting – etc. These requirements (and additional which have to be identified) must be common amongst all AUTOSAR parties to have a common and testable DEM-BSW.
Rationale:	As of today the DEM primarily specifies interfaces to implement the content-wise requirements regarding the behavior of a fault manager which can be derived from ISO14229-1 and ISO15031-5/SAEJ1979. However the actual behavior of the fault manager when a fault is indicated by a SW-C (i.e. Dem_SetEventStatus) is currently undefined. Each OEM has different requirements regarding storage of faults and associated environmental data, e.g. some require to store a record for the first occurrence and the most recent occurrence of the same fault while others store one record per each occurrence of the same fault. This behavior varies from simple DTC storage and DTC status implementation to environmental data storage and self-healing, prioritization and overwriting criteria.
Use Case:	A common implementation of a DEM which can be used unaltered by each AUTOSAR-participating OEM can only be implemented when the aforementioned harmonization is achieved. Until then, only a DEM hull can be provided by AUTOSAR software suppliers which then have to be “populated” by an OEM-specific fault manager (DEM) version.
Dependencies:	--
Conflicts:	Fault management is usually defined in OEM-specific requirement specifications and may be considered as a competitive advantage if a certain storage concept is used. Harmonization of content wise requirements would require participating OEMs to agree on common implementations and may result in several OEMs having to change their internal specifications.
Supporting Material:	--
Contributes to:	--

3.9.2 [BRF00231] Forwarding of notifications about new freeze frame data

ID:	BRF00231
Initiator:	BMW and Elektrobit
Date:	18.01.2008
Short Description:	Forwarding of notifications about new freeze frame data
Importance:	high | **medium** | low
Description:	The DEM shall be enabled to notify other SW-C (or BSW modules) about new freeze frame data (e.g. time stamp). If this functionality is configured for an event, it shall be executed on each entry of a new freeze frame of this event into the event memory.
Rationale:	In the current version of the DEM SWS, there is no possibility to provide freeze frame data (like time stamp) to another SW-C / BSW module beside the DCM. Additionally this functionality provides a simple way for supporting this data to other components (at every time, where new data are available), so that no cyclic polling is needed.
Use Case:	The information provided by this functionality is needed by modules like a special 'Diagnostic active response handler'.
Dependencies:	None
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.9.3 [BRF00232] Control of event handling

ID:	BRF00232
Initiator:	BMW and Elektrobit
Date:	18.01.2008
Short Description:	Control of event handling
Importance:	**high** | medium | low
Description:	The DEM module shall provide locking functionality for DTC-deletion. This functionality is configurable per event.
Rationale:	It shall be possible to disable the clearance of some dedicated events e.g. permanent error of OBD.
Use Case:	Some dedicated events must never get cleared from event memory, while the ECU is in an special operation mode (e.g. assembly-, transport-, or flash-mode).
Dependencies:	None
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.9.4 [BRF00233] Memory Overflow indication

ID:	BRF00233
Initiator:	BMW and Elektrobit
Date:	18.01.2008
Short Description:	Memory Overflow indication
Importance:	high | **medium** | low
Description:	The DEM module shall indicate for each Event Memory if the event memory

	is full and the next event occurs to be stored in this event.
Rationale:	If there are limitations of the memory size it is necessary to indicate the memory overflow and to provide a displacement strategy.
Use Case:	Memory overflow indicates that some root causes were not described by the content of the event memory.
Dependencies:	None
Conflicts:	--
Supporting Material:	The displacement strategy is not specified in AUTOSAR. It shall be indicated via an API function. The information could be linked to a Extended Data Record.
Contributes to:	--

3.10 Support of large data types Concept

3.10.1 [BRF00004] Support for variable-length Data Types

ID:	BRF00004
Initiator:	AUTOSAR PL
Date:	26.04.2007
Short Description:	Support for variable-length data types
Importance:	high | **medium** | low
Description:	Often it is necessary to transfer data with arbitrary size (could be bigger than 8 bytes). This must be covered by all affected communication functionality.
Rationale:	Handling of strings with a fixed length is not preferable, since a lot of ECU resources will be allocated by this approach. The usage of an always fixed size would result in wasting memory and bandwidth in case of not used but reserved space. The alternative usage of a couple of interfaces and signals to serve different sizes will complicate the APIs and waste configuration freedom.
Use Case:	– Transferring message and information strings of variable length between ECU's, e.g. Central ECU and Instrument Cluster. – Transferring the content of a received SMS to display it. – Transferring (variable) strings to a display
Dependencies:	--
Conflicts:	--
Supporting Material:	Additional information must be transferred. This can be either a termination (like 0x00 terminating C-strings) or the dedicated length information. RTE, COM and perhaps the PDUR will be influenced. A mechanism to transfer fragmented application data is necessary, in case transfer shall be done by CAN or LIN. The fragmentation and reassembly must be handled somewhere. CAN and LIN restricts the frame length to 8 bytes. Current transport protocol definition (able to transfer large numbers of bytes) covers diagnostic purposes only.
Contributes to:	--

3.10.2 [BRF00005] Support of Data Type Size > 8 Bytes (Signal Size)

ID:	BRF00005
Initiator:	AUTOSAR PL
Date:	26.04.2007
Short Description:	Support of data type size > 8 bytes (signal size)
Importance:	**high** | medium | low
Description:	Sometimes it is necessary to transfer data with a bigger fixed size than 8 Bytes only. This is regular in case of complex data types containing coherent data elements. Currently it is in the responsibility of the SWC's to deal with huge amounts (or summarized sizes) of coherent data. The transfer of such big data types must be covered by all affected communication functionality.
Rationale:	It is a better solution to provide the possibility to transfer such big data types instead to leave the according solution in SWC's responsibility. A central solution avoids multiple implementation of the same problem.
Use Case:	A SWC provides a number of coherent data occupying more than 64 bits

Dependencies:	--
Conflicts:	CAN and LIN restricts the frame length to 8 bytes. Current transport protocol definition (able to transfer large numbers of bytes) coves diagnostic purposes only.
Supporting Material:	COM and perhaps the PDUR will be influenced. A mechanism to transfer fragmented application data is necessary, in case transfer shall be done by CAN or LIN. The fragmentation and reassembly must be handled somewhere.
Contributes to:	--

3.10.3 [BRF00290] Support of Array Type with dynamic Size

ID:	BRF00290
Initiator:	BMW
Date:	30.01.2008
Short Description:	Array type should provide size information.
Importance:	high | **medium** | low
Description:	The AUTOSAR specification in Release 3.0 only supports array types with a maximum size that is known at specification time (“maxNumberOfElements”). The maximum Array size might not be needed always by the SW-C using the Array. The RTE does not know how many of the ArrayElements are actually in use when called by the SW-C. Therefore the maxNumberOfElements is transported. This can lead to unnecessary usage of bandwidth on the physical transportation medium. To avoid this unnecessary traffic the SW-C needs a mechanism to inform the RTE about the number of used ArrayElements. It suffices to provide this size information for byte and char arrays.
Rationale:	In the MM/T/HMI domain there is in some cases the need for data types with dynamic length. The MM/T/HMI (especially the HMI parts) often make use of dynamic data, for example to display a list of tracks or a list of POIs (point of interest). The size of these lists may differ from a couple of elements (CD track list) up to a huge number of complex data types (navigation destination data like [country, city, streetname, house number]). Always sending the maxNumberOfElements would lead to excessive usage of unneeded bandwidth. Byte and char arrays contain strings with dynamic information and therefore the length variation of the content can be significant.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.11 VMM AMM Concept

3.11.1 [BRF00045] Support Disable ‘normal’ Communication

ID:	BRF00045
Initiator:	AUTOSAR PL
Date:	03.05.2007
Short Description:	Support Disable ‘normal’ Communication
Importance:	high | medium | **low**
Description:	According to ISO 14229-1(Service $28) it shall be possible to disable normal communication and at the same time keep a predefined set of PDUs active, including diagnostic communication. Currently the DCM and ComM do not support this. According to BSW09083 in "AUTOSAR_SRS_Mode_Mgmt.SRS" the ComM only supports three communication modes: – full communication (send & receive operations) – silent communication (receive operations) – no communication (neither send nor receive operations)
Rationale:	It must be possible to suppress normal communication during execution of diagnostic tasks to assign maximum bandwidth.
Use Case:	– Upload of flash images to ECUs – Download of error logs – Execution of system diagnostics
Dependencies:	--
Conflicts:	--
Supporting Material:	GM require that support for disabling/enabling "networkManagementCommunicationMessages and normalCommunicationMessages" (according to Appendix B.1 in ISO 14229-1) without disabling diagnostic messages exist.
Contributes to:	--

3.11.2 [BRF00060] Control Runtime Changeable LIN Schedule Tables

ID:	BRF00060
Initiator:	BMW
Date:	03.05.2007
Short Description:	Control runtime changeable LIN schedule tables
Importance:	**high** | medium | low
Description:	Whereas the LIN interface already provides functionality to execute the change of schedule tables, an ‘authority’ is needed to initiate the change of schedule tables during runtime, if necessary, possible and allowed.
Rationale:	The LIN interface is not able to decide whether and when to switch the schedule tables. An upper ‘authority’ to decide and cause the switch is needed.
Use Case:	Changed control mode of exterior mirrors from switch controlled to key controlled
Dependencies:	--
Conflicts:	--
Supporting Material:	--

Contributes to:	--

3.11.3 [BRF00073] Port Groups

ID:	BRF00073
Initiator:	Denso
Date:	25.05.2007
Short Description:	Port Groups
Importance:	high | **medium** | low
Description:	PortGroups shall represent the concept of users of the communication management in the VFB (internal behavior) and shall be used to group ports of a SWC that are required by the same functionality. These ports will logically be one COMM-user. A PortGroup shall communicate with the COMM as one user.
Rationale:	In AUTOSAR release 2.1 it is not possible to represent the COMM users using the software component template. Correspondingly, the semantics of a user to have full - silent - or no communication is unclear. Which ports of a software component are affected by the modes of this user? As an added value, a 'COMM configurator' could automatically set up the matrix of users to required communication infrastructure by evaluating the ModeGroups and the communication hardware that is connected to the corresponding ports.
Use Case:	It is common use cases that a software component has some ports that have to be highly available and some ports that are required only for some special functionality. This could be realized by grouping these ports into two PortGroups
Dependencies:	ComM configurator, Software Component Template, evtl. RTE
Conflicts:	none
Supporting Material:	--
Contributes to:	--

3.11.4 [BRF00103] Control of Mode dependent IPDU Groups

ID:	BRF00103
Initiator:	Denso
Date:	29.06.2007
Short Description:	Control of mode dependent IPDU groups
Importance:	high | **medium** | low
Description:	It shall be possible to enable IPDU groups in selected vehicle modes only.
Rationale:	In release 2.1, it is only possible to limit the communication of a whole channel. In many cases, depending on the mode, it is necessary to keep full communication for the channel but limit the IPDU groups that are allowed to transmit.
Use Case:	This can be used for signals that are available in one power mode of the vehicle and are unavailable in another power mode. Note that it is often not sufficient to limit the whole channel to no communication. This feature can reduce the bus load in critical modes.
Dependencies:	VMM/AMM; COM; ComM
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.11.5 [BRF00104] Mode Dependent Reset of Initial Values

ID:	BRF00104
Initiator:	Denso
Date:	29.06.2007
Short Description:	Mode dependent reset of initial values
Importance:	high | **medium** | low
Description:	It shall be possible to reset each signals initial value to a configurable initial value when a vehicle mode is entered.
Rationale:	In release 2.1, it is only possible to set the initial values of a signal once. But, in certain power modes, a signal will not be available. To recover quickly, when the signal is available again, it shall be possible to restore an initial value.
Use Case:	This can be used for signals that are available in one power mode of the vehicle and are unavailable in another power mode. Note that it is often not sufficient to limit the whole channel to no communication.
Dependencies:	VMM/AMM; COM; ComM; BRF00103
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.11.6 [BRF00189] Enable SWCs to request dedicated Modes

ID:	BRF00189
Initiator:	AUTOSAR WP Vehicle and Application Mode Management
Date:	03.12.2007
Short Description:	Enable SWCs to request dedicated Modes
Importance:	**high** | medium | low
Description:	If configured, on each ECU one or several ECUs SWCs can request a mode. The mode request is propagated to a specific functionality, which is responsible to control the affected BSW according to the mode requests received.
Rationale:	Dependent on its inner states a SWC must be able to initiate, or to request at least, the change of operational modes. This functionality can be used to adapt power consumption (and emissions) to states of applications or the whole car. SWCs must be able to initiate mode changes due to the impossibility to define mode dependent behavior which is generic and user independent. The user specific behavior will be realized in user specific SWCs.
Use Case:	Use cases which require a control of the basic software and runnable execution depending on mode information..
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.11.7 [BRF00190] Configurable BSW internal Evaluation of Mode Requests

ID:	BRF00190
Initiator:	AUTOSAR WP Vehicle and Application Mode Management
Date:	03.12.2007

Short Description:	Configurable BSW internal evaluation of mode requests
Importance:	**high** | medium | low
Description:	The behavior of the BSW internal mode controlling functionality will be configurable due to a parameter driven behavior. While the principles of the mode controlling functionality are generic, the concrete behavior can be configured user specific.
Rationale:	This allows user specific, mode dependent behavior of SWCs by using the generic mechanism of mode control but without the necessity of a user specific implementation of the mode control in any SWCs.
Use Case:	Use cases which require a control of the basic software and runnable execution depending on mode information.
Dependencies:	BRF00189
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.11.8 [BRF00191] Propagation of Mode Information

ID:	BRF00191
Initiator:	AUTOSAR WP Vehicle and Application Mode Management
Date:	03.12.2007
Short Description:	A mode can affect SWCs located on several ECUs. Therefore by configuration the mode information must be propagated to several ECUs.
Importance:	**high** | medium | low
Description:	A mode can affect SWCs located on several ECUs. Therefore by configuration the mode information must be propagated to several ECUs.
Rationale:	A system wide synchronization of modes is necessary to keep the whole system consistent.
Use Case:	All use cases which require controlling the basic software and runnable control.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.12 Fixed Data Exchange Concept

3.12.1 [BRF00105] System Parameter

ID:	BRF00105
Initiator:	Continental
Date:	02.07.2007
Short Description:	system parameters that are set during compilation time (implemented by RTE with macro) or during Runtime (implemented with calibrations)
Importance:	high | **medium** | low
Description:	The standard sender-receiver mechanism is used to exchange data in a dynamic way. The exchange of #define macros do not require an explicit and dynamic read since the value is fixed. The value should be set in all consumers at compile time. This should be formalized with port to show the dependencies. This feature could be implemented by an extension of the Calibration parameter concept: for a CalPrm, a parameter SettingType could indicate whether it is pre-fixed (SettingType =CompileTime i.e. a #define macro) or post-fixed (SettingType = RunTime i.e. a read of calibration in flash). This new parameter SettingType helps the RTE to choose the right implementation. The implementation of a SWC consuming such CalPrm shall not rely on whether it is CompileTime or RunTime fixed. So a fixed data (i.e. a CalPrm) should not be used in pre-compilation directives (e.g. #IF) even if SettingType =CompileTime. Because the integrator can decide to use calibration in flash (SettingType =RunTime) instead. In this case the RTE API implementation will bring to a compilation error since #IF can only be used with macro. The use of such #IF directives are linked the concept "Variant handling".
Rationale:	Some SWC can hold some fixed data (#define macro) that can be consumed by other SWCs. The fixed data should be set in all consumers at compile time and not dynamically with current sender-receiver communication mechanism.
Use Case:	• A SWC can hold some general constant values, e.g. PI, Avogadro constant, that are used in calculation by many SWC. • A SWC can hold some application fixed data that can be used in calculation by other SWC.
Dependencies:	BRF00157: this feature proposes to add a new element to set an "unconnected" R-port to a fixed value. The calibration concept BRF00029: Variant handling BRF00167: macro value to set variant
Conflicts:	Issue: a Calprm could either be used as data in an algorithm, either as an input for variant. Example: in a power train application, a parameter NumberOfCylinder could be used in a for loop or in a pre-compilation directive to set a variant
Supporting Material:	--
Contributes to:	--

3.12.2 [BRF00157] Fixed Values for R-Ports

ID:	BRF00157
Initiator:	Continental

Date:	04.10.2007
Short Description:	Fixed values for R-ports
Importance:	high | medium | **low**
Description:	Provide Initial values for unconnected Sender-Receiver R-Ports
Rationale:	If same SWC is used in different AUTOSAR Systems the SWC has usually not needed Ports in the System with lower functionality. In this case for the Sender-Receiver R-Ports initial values have to be provided.
Use Case:	Easy reuse of SWC in Systems with different overall functionality. Enable lean design of AUTOSAR Systems. Avoid the overhead of creation of inartificial Software Components providing initial values.
Dependencies:	--
Conflicts:	--
Supporting Material:	Currently the RTE rejects configurations with unconnected R-Ports due to System integrity check reasons. Proposal: Define a dedicated element on VFB level which is able to provide static initial values for required sender receiver ports.
Contributes to:	--

3.13 Variant Handling Concept

3.13.1 [BRF00029] Variant Handling on VFB Level

ID:	BRF00029
Initiator:	GM
Date:	03.05.2007
Short Description:	--
Importance:	**high** | medium | low
Description:	Specify a mechanism how variability can be described on VFB level and how this variability is further broken down into the methodology and templates.
Rationale:	Electronic systems in the car are designed to support several variants in order to save effort and money and increase the quality. Concepts to support variants need to be supported on VFB level.
Use Case:	1. Using the same ECU (hardware and software) for several vehicle lines. This leads to different communication matrices, which need to be used depending in which vehicle line the ECU is build in. Also the functionality may be slightly different in each vehicle line. 2. A project can choose out of different existing implementations of Autosar SWCs respectively SW-compositions with same or compatible interfaces to implement the sum of the system functionality. This addresses pre-built variant handling.
Dependencies:	The main work will be performed in WP General Methodology and Configuration for inclusion of the concepts in the AUTOSAR Templates.
Conflicts:	--
Supporting Material:	Supporting Material: The AUTOSAR BSW already supports several configurations for some BSW modules. What is missing is the proper methodology and input from higher level templates (SWC-T, Sys-T, EcuC-T) and a concept on VFB level.
Contributes to:	--

3.13.2 [BRF00155] Support for different Interfaces of a SW-C

ID:	BRF00155
Initiator:	Renault
Date:	07.09.2007
Short Description:	Support for different interfaces of a SW-C
Importance:	high | **medium** | low
Description:	Implement diversity by enabling or disabling PortInterfaces based on parameters and allow pre-compile optimizations with #define.
Rationale:	Diversity management: regroup all variants of a software component in a single ComponentType with different interfaces depending on a parameter. This feature can allow reducing the number of objects to be tracked in version management systems since the same component reference can be used in different configurations.
Use Case:	The same component can have different interfaces if used in a Diesel or gasoline engine, however part of the functionality is unchanged. For example both variants have a lot of common sensors, and a few specific sensors which require different interfaces.
Dependencies:	Software component template RTE SWS
Conflicts:	This could be an extension to BRF00029, to use the VFB mechanism for

	variant handling in the actual SW-C implementation.
Supporting Material:	--
Contributes to:	--

3.13.3 [BRF00167] Macro Value to set Variant

ID:	BRF00167
Initiator:	Continental
Date:	12.10.2007
Short Description:	Macro value to set variant
Importance:	high | **medium** | low
Description:	Macro can be used to inhibit/activate a code variant. A macro is produced by a SWC (BRF00105, CalPrm with SetWhile=CompileTime) then consumed by other SWC. The macro is used in compilation directives (e.g. #IF) to activate or inhibit some part of algorithm.
Rationale:	Code Variants are often handled with macro values.
Use Case:	see BRF00029
Dependencies:	BRF00105 BRF00029
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.14 Bus Monitoring Issues Concept

3.14.1 [BRF00087] API to query the FlexRay Rate and Offset Correction

ID:	BRF00087
Initiator:	Daimler
Date:	29.05.2007
Short Description:	API to query the FlexRay rate and offset correction
Importance:	high | medium | **low**
Description:	1) It shall be possible to query rate and offset correction of a specific FlexRay Cluster calculated by the clock synchronization algorithm 2) It shall be possible to query rate and offset correction of a specific FlexRay Communication Controller calculated by the clock synchronization algorithm 3) It shall be possible to send rate and offset correction on the bus
Rationale:	It could be useful to know the current drift rate and the offset of every FlexRay controller. The new feature is needed to fulfill the safety requirement OSR107 AUTOSAR shall provide ECU hardware monitoring and testing to detect faults in the following components: – CPU – memory – peripheral devices – communication components – address, data and control buses
Use Case:	The monitoring of ‘drift’ of dedicated FlexRay controllers could shorten the error analysis. The monitoring of ‘drift’ of dedicated FlexRay controllers could allow the prediction of hardware errors.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.14.2 [BRF00093] Detection of missed FlexRay Startup Frames

ID:	BRF00093
Initiator:	Daimler
Date:	29.05.2007
Short Description:	Detection of missed FlexRay startup frames
Importance:	high | medium | **low**
Description:	It shall be possible to detect the situation that no startup frames are present but the FlexRay is still operating because of still available sync frames.
Rationale:	Every network consists of at least 2 Coldstart nodes and these coldstart nodes send startup frames. All Coldstart nodes are shut down if no startup frame is present thus it is impossible for any node to integrate himself into the network without a previous shutdown of all nodes. The indication could be used to "force a shutdown of all nodes" to allow for a restart with all nodes.

Use Case:	A network with 3 coldstart nodes and 2 sync nodes. If the 3 coldstart nodes fall asleep/perform a restart because of an error (e.g. reset or low voltage) it is necessary to restart the FlexRay to allow the reintegration of all FlexRay nodes.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.14.3 [BRF00264] FlexRay Transceiver Errors reported for Diagnostics

ID:	BRF00264
Initiator:	GM
Date:	31.01.2008
Short Description:	FlexRay transceiver errors reported for diagnostics
Importance:	**high** | medium | low
Description:	The intention of this feature is to provide error and status information available by the transceiver to the host. While different transceivers provide different status information, and make them available in different ways, this feature attempts to standardized the least common error and status information while leaving room for transceiver proprietary information.
Rationale:	AUTOSAR Transceiver driver hides the details of transceiver implementation. Upper interface to physical transceiver driver provides a standardized error interface, perhaps 0-127 reflects the mandatory errors that are supported by ALL transceivers, and 128-255 could reflect the transceiver specific error information.
Use Case:	Diagnostics would need this to detect this to understand the state of the characteristics of the system. Useful during development as well as continued operation.
Dependencies:	Error Handling, DET, DCM, FlexRay Driver, FlexRay Interface.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.14.4 [BRF00265] Reporting Applied FlexRay Clock Correction Terms

ID:	BRF00265
Initiator:	GM
Date:	31.01.2008
Short Description:	Reporting applied FlexRay clock correction terms
Importance:	**high** | medium | low
Description:	Norm
Rationale:	There are no functions to provide applied Clock correction terms to the upper layers.
Use Case:	Diagnostics would need this to detect this to understand the state of the characteristics of the system. Useful during development as well as continued operation.
Dependencies:	Error Handling, DET, DCM, FlexRay Driver, FlexRay Interface.
Conflicts:	--
Supporting Material:	--

Contributes to:	--

3.14.5 [BRF00266] Report of Aggregated FlexRay Channel Status Error

ID:	BRF00266
Initiator:	GM
Date:	31.01.2008
Short Description:	Report of aggregated FlexRay channel status error
Importance:	**high** | medium | low
Description:	Norm
Rationale:	Aggregated channel status error is required for correct diagnosis of the bus.
Use Case:	Diagnostics would need this to detect this to understand the state of the characteristics of the system. To detect and differentiate between different types of errors during development and operation of the system.
Dependencies:	Error Handling, DET, DCM, FlexRay Driver, FlexRay Interface.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.14.6 [BRF00267] Report of FlexRay Status Data for number of Sync Frames

ID:	BRF00267
Initiator:	GM
Date:	31.01.2008
Short Description:	Report of status data number of Sync Frames
Importance:	**high** | medium | low
Description:	Norm
Rationale:	Status data contains number of sync frames received or transmitted for each channel.
Use Case:	Necessary for Diagnostic purposes to detect persistent asymmetry between applied clock correction terms of different nodes. Useful to set DTC. Allows to service the nodes that are missing.
Dependencies:	Error Handling, DET, DCM, FlexRay Driver, FlexRay Interface.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.14.7 [BRF00299] Report of FlexRay Status Data for list of IDs

ID:	BRF00299
Initiator:	GM
Date:	31.01.2008
Short Description:	Report of status data for list of IDs
Importance:	**high** | medium | low
Description:	Norm
Rationale:	Status data contains list of IDs received or transmitted for each channel.
Use Case:	Necessary for Diagnostic purposes to detect persistent asymmetry between

	applied clock correction terms of different nodes. Useful to set DTC. Allows to service the nodes that are missing.
Dependencies:	Error Handling, DET, DCM, FlexRay Driver, FlexRay Interface.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.15 Support of SAE J1939 Protocol Features Concept

3.15.1 [BRF00168] Support of SAE J1939 Protocol Features

ID:	BRF00168
Initiator:	Volvo, Daimler and Vector
Date:	22.10.2007
Short Description:	Support subset of SAE J1939 protocol.
Importance:	**high** | medium | low
Description:	Subset of selected SAE J1939 protocol features to allow proper communication with SAE J1939 components. **CAN Identifier Handling:** A function needs to know the source address related to a PGN. The same PGN can be sent by multiple senders in the same vehicle. The combination of source addresses with the same PGN is limited (5 – 10 cases) and known during configuration. Dynamic addressing is not needed; Priority bits in CAN-Ids are fixing. CAN-Ids are completely known at configuration time. Re-Configuration shall be possible by changing EOL data (parameter) without re-compiling the CAN driver software. This is needed for own TX-messages, too. **Message Request:** Requesting of messages from other devices is not needed. **Transport Layer:** Both SAE J1939 transport protocols CMDT (Connection Mode Data Transfer) and BAM (Broadcast Announce Messaging) shall be supported. **Diagnostic messages:** Will be sent out periodically and on error conditions. **Signal lengths:** Parameters with variable length are needed. Parameters longer than 64 bits are needed. **J1939 Network Management:** Only a subset of J1939 network management is needed: On 'Address claim' and address conflict the lower priority ECU will go silent.
Rationale:	The requested subset of SAE J1939 functionality will allow integration and communication with J1939 CAs. Many Truck OEMs needs to maintain systems with multiple networks, including SAE J1939.
Use Case:	Use Case A: Existing SAE J1939 off-the-shelf components can be integrated or reused. Use Case B: SAE J1939 protocol is an industry standard in many markets. Therefore support of J1939 is mandatory for Truck OEMs.
Dependencies:	RTE specification COM
Conflicts:	--
Supporting Material:	Some explanations: – PGN stands Parameter Group Number: The 29 bit CAN identifier is organized into several fields. One of the fields is the PGN which is standardized by the SAE for all information, e.g. engine temperature has an own PGN. – EOL stands for End Of Line. It would be better to talk here about Post Build Configuration. – Priority Bits are the upper 3 bits in a J1939 CAN-ID. These bits

	do not identify the content of the message but are used to change the priority for the CAN message on the bus at runtime, what results in a changed CAN-identifier. This feature is not needed for the defined subset. CAN-Identifiers are considered to be fix at runtime. – 'This is needed for own TX-messages, too.': In full J1939 receive and transmit CAN-Identifier are dynamic. The mechanism to determine the CAN-Identifier is slightly different for receive and transmit messages. The intended subset works with fixed CAN-IDs but needs post build configuration for RX and TX identifiers.
Contributes to:	--

3.16 LIN 2.1 Std Concept

3.16.1 [BRF00184] Adaptation of the LIN Stack at LIN 2.1 Specification

ID:	BRF00184
Initiator:	AUTOSAR WP COM Stack
Date:	03.05.2007
Short Description:	Adaptation of the LIN stack at LIN 2.1 specification
Importance:	high | medium | **low**
Description:	Adaptation of the LIN Stack at LIN 2.1 Specification
Rationale:	--
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.17 FlexRay ISO TP Concept

3.17.1 [BRF00192] Support of FlexRay Message IDs

ID:	BRF00192
Initiator:	BMW
Date:	09.12.2007
Short Description:	Support of FlexRay message IDs
Importance:	high | **medium** | low
Description:	The FlexRay specification describes a communication mechanism which allows the identification of data frames by an assigned 'message id' which is contained in the data frame as part of the data load. This identification of data frames is an alternative possibility
Rationale:	Due to a high load of the FlexRay it becomes impossible to generate communal (covering more than one model line) schedules. A switch from a hard linked slot/cycle assignment to a quality of service approach solves this problem.
Use Case:	The movement of data a source (a SWC) in a FlexRay network from one ECU to another ECU would not lead to a recompilation of other ECUs due no change of the FlexRay schedule would be necessary.
Dependencies:	Currently this communication mechanism has been classified as 'optional'. Nevertheless each existing FlexRay Communication Controller implements this functionality and the FlexRay consortia discuss currently to make it mandatory.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.17.2 [BRF00252] ISO 10681-2 conform FlexRay Communication

ID:	BRF00252
Initiator:	Daimler
Date:	28.01.2008
Short Description:	ISO 10681-2 conform FlexRay Communication Support of ISO 10681-2 Road vehicles – Communication on FlexRay – Part 2: Communication Layer Services
Importance:	**high** | medium | low
Description:	The FlexRay Communication Stack (FrDrv, FrIf and FrTp) shall support communication as described within the ISO-10681-2 document.
Rationale:	(a) Currently the FrTp is not based on an official ISO document. (b) Currently communication within FlexRay's dynamic segment is also static configured. There is no possibility to send FR-Frames with dynamic Payload length (normal diagnostic communication vs. flash reprogramming communication). Also there is no possibility to allocate bandwidth dynamically (flash reprogramming of multiple ECUs in parallel vs. reprogramming of a single ECU in burst mode). (c) ISO 10681 supports a byte oriented retry mechanism. This is an optimization of transfer time in case of large data transfers. (d) ISO 10681 supports data transfers with unknown message length at the beginning of the transmission.
Use Case:	(a) Data Transfer for measurement, calibration and diagnostics applications - segmented/unsegmented communication

	- acknowledged/unacknowledged communication - communication with known /unknown message length (streaming communication) (b) Diagnostics and Flash reprogramming - bandwidth optimization for normal diagnostic communication - bandwidth optimization for flash reprogramming.
Dependencies:	FrTp, FrIf, FrDrv.
Conflicts:	--
Supporting Material:	Document: ISO 10681-2 Road vehicles – Communication on FlexRay – Part 2: Communication Layer Services
Contributes to:	--

3.18 LIN Transceiver Driver

3.18.1 [BRF00228] LIN Transceiver to be specified

ID:	BRF00228
Initiator:	Bosch
Date:	25.01.2008
Short Description:	LIN Transceiver to be specified
Importance:	**high** | medium | low
Description:	The LIN Transceiver specification is currently missed in AutoSar. But a specification of the LIN Transceiver behavior shall be established.
Rationale:	Currently there are several lacks in interface descriptions / concepts referring to the not existing LIN transceiver. So for all the Lin transceiver implementations solutions are needed that are not specified in AutoSar. To close this gap and have a integrated description from LIN driver to LIN interface the LIN transceiver document is needed.
Use Case:	--
Dependencies:	LIN SRS might be adapted
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.19 FlexRay Protocol Spec Issues Concept

3.19.1 [BRF00268] Support for FlexRay Single Slot Mode

ID:	BRF00268
Initiator:	GM
Date:	31.01.2008
Short Description:	Support for FlexRay Single Slot Mode
Importance:	**high** | medium | low
Description:	FlexRay includes a "single slot" mode that can be can be configured for use as an extension of the startup process. During integration all FlexRay nodes only transmit one frame in the static segment (identified as the "key slot"). They do this to minimize the number of frames they might corrupt if their timing is incorrect and they transmit in the slots of other nodes. Once they integrate there are two possibilities. Nodes that do NOT use the single slot mode allow their other frames (static and dynamic) to be automatically enabled as soon as they integrate. The underlying assumption is that the mere fact that they integrated is sufficient confirmation of their conformance to the schedule to allow these transmissions to be enabled. Nodes that use the single slot mode must enable the transmission of their other frames with an explicit host command. The underlying assumption in this case is that the host will first perform some sort of explicit confirmation that its transmitted frames are properly timed before it executes the command. This covers the scenario where only the transmission timing is flawed - in a way that does not impact the ability to integrate. The likely way to perform this confirmation is to configure another node to explicitly confirm the timing of the single slot frame with a flag in a frame that it transmits. It simply sends a specified frame with the confirmation flag set if it receives the single frame from the node whose timing it is supervising. If the host sees receives this confirmation flag, it issues the command to the controller to disable single slot mode and this causes all frames to be enabled.
Rationale:	FlexRay provides support for Single Slot mode.
Use Case:	Active NM only after controller transitions from Single Slot mode to normal communication (i.e., all slots are activated). Single Slot mode is used for Diagnostics to limit transmission of all slots until node is sure that it is not interfering with transmission of normal communication messages by other nodes, i.e., the node transitions from Single Slot mode when it is sure that it is in a compatible system mode.
Dependencies:	Mode management, COM, Vehicle Mode Management, FlexRay State Manager
Conflicts:	
Supporting Material:	With the current FR Driver it is not possible to use Single Slot Mode. All Slot modes shall be supported. API shall be provided to switch to ALL_SLOT mode; this shall be provided by the FrIf. The FrSm will not call this API but this has to be done by the OEM specific SM Extension. Potentially the Error Handling Group shall be involved. The Fr Driver and the FrIf seem to be involved but it is not listed in the dependencies list if the feature request.
Contributes to:	--

3.19.2 [BRF00273] Support for Dual FlexRay Channels

ID:	BRF00273
Initiator:	AUTOSAR PL
Date:	03.05.2007
Short Description:	Support for Dual FlexRay channels
Importance:	**high** | medium | low
Description:	There are basically three types of clusters: a) Single channel. b) Dual channel with all ECUs on both channels c) Dual channel with some ECUs on only one channel. The limitations in AUTOSAR make it clear that (a) is supported and (c) is not since you cannot have (c) unless you can wake up all devices. There is nothing in the limitations directly about b), but without managing coldstart inhibit and providing a mechanism to coordinate wakeups on the two channels, you cannot start a dual channel cluster. This has been stated before. The point I am making is that the stated limitations give the impression that wakeup forwarding is missing, but the real limitation (although not stated) is that only single-channel clusters are supported. I would think this should be more obviously expressed. The problem is also deeper than FRSM. NM doesn't work right either unless some node(s) cascade NM votes from one channel to the other and this can be tricky.
Rationale:	FlexRay provides dual channels.
Use Case:	The mechanisms should be improved to handle the most general case consisting of a "reference configuration" with: 1. Single channel nodes connected to only channel A 2. Single channel nodes connected to only channel B 3. Dual channel nodes connected to both channels A and B Some further analysis will be necessary to identify all necessary changes, but the following two objectives are identified at this point in time: 1. Proper support for scenarios where wakeup is initiated by single channel nodes, but startup must be performed by the dual channel nodes. 2. Improvements to FlexRay NM to allow votes from one channel to be "mirrored" onto the other channel in a manner that still synchronizes shutdown of both channels. The first objective requires the following new mechanisms: 1. A wakeup forwarding mechanism is needed so that a wakeup transmitted by a single channel node will be repeated onto the other channel by one of the dual channel nodes. 2. Proper support for the coldstart inhibit capability of the FlexRay protocol so that dual channel nodes that receive a wakeup can inhibit startup until they have forwarded the wakeup to the other channel and allowed sufficient time for those nodes to awaken. 3. Sufficient configurations to ensure that when dual channel nodes are present, single channel nodes cannot perform a coldstart. The second objective requires the following new mechanisms: 1. Since NM votes broadcast by single channel nodes are not visible to

	single channel nodes on the other channel, a mechanism is needed to forward the aggregated vote observed on one channel to the other channel (and vice versa). This has some similarities to the current capability of the NM gateway for CAN, but additional modifications may be necessary since the forwarded votes are always delayed by at least one cycle and care must be taken to make sure that deadlocks are not created. 2. The NM algorithm must be improved to allow meaningful repetition cycles to be constructed around votes a. Have the same repetition cycle structure as the receiving node b. Have a cycle offset due to being relayed by the NM gateway. It may be possible to exploit the sparse schedules described in section (?) to create schedules where this becomes transparent to the receiving nodes
Dependencies:	Network Management, FlexRay State Manager
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.20 FlexRay Spec 3.0 Concept

3.20.1 [BRF00272] Support for active FlexRay Stars

ID:	BRF00272
Initiator:	GM
Date:	31.01.2008
Short Description:	Support for active FlexRay stars
Importance:	**high** | medium | low
Description:	Active Stars are part of valid FlexRay topologies.
Rationale:	
Use Case:	1. Active stars are required for electrical isolation of Branches. 2. Device drivers should read out status information to support diagnostics and selective control (and possibly enabling and disabling) of transceivers.
Dependencies:	FlexRay Transceiver, FlexRay Interface, FlexRay State Manager.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.20.2 [BRF00277] Support of FlexRay Specifications 3.0

ID:	BRF00277
Initiator:	BMW
Date:	31.01.2008
Short Description:	Support of FlexRay Specifications 3.0
Importance:	**high** | medium | low
Description:	The AUTOSAR FlexRay Stack shall support the - FlexRay Protocol Specification 3.0 - FlexRay Electrical Physical Layer Specification 3.0
Rationale:	AUTOSAR Release 4.0 shall be designed to fulfill the latest FlexRay specifications too.
Use Case:	- Usages of AUTOSAR 4.0 software together with the latest FlexRay hardware
Dependencies:	FlexRay Stack
Conflicts:	--
Supporting Material:	http://www.flexray.com/
Contributes to:	--

3.21 TCP/IP CommStack Externsions Concept

3.21.1 [BRF00283] Enable BSW to communicate via TCP/IP

ID:	BRF00283
Initiator:	BMW
Date:	21.01.2008
Short Description:	Enable BSW to communicate via TCP/IP
Importance:	**high** | medium | low
Description:	Add the TCP/IP protocol suite.
Rationale:	- Use of standard computer equipment for logging and debugging. - Not all services use the PDU Router to communicate
Use Case:	- Transport Log and Trace data e.g. from the Dlt-Module - Network Management Access to control ECU states
Dependencies:	BRF00286 (Support Ethernet as an additional communication medium)
Conflicts:	
Supporting Material:	RFC 1122, Requirements for Internet Hosts – Communication Layers, IETF (http://www.ietf.org/rfc/rfc1122.txt?number=1122) IEEE 802.3-2002 (Ethernet) (http://standards.ieee.org/getieee802/802.3.html)
Contributes to:	--

3.21.2 [BRF00284] Enable the Applications to communicate via TCP/IP

ID:	BRF00284
Initiator:	BMW
Date:	21.01.2008
Short Description:	Enable all Applications to communicate directly via TCP/IP
Importance:	**high** | medium | low
Description:	Enable applications to send and receive data via a TCP/IP protocol suite stack by using a standard interface into the RTE.
Rationale:	Increased need for generic data exchange between applications, that can not be easily mapped into signals.
Use Case:	- http based configuration and status reporting
Dependencies:	BRF00286 (Support Ethernet as an additional communication medium)
Conflicts:	-
Supporting Material:	RFC 1122, Requirements for Internet Hosts – Communication Layers, IETF (http://www.ietf.org/rfc/rfc1122.txt?number=1122) IEE 802.3-2002 (Ethernet) (http://standards.ieee.org/getieee802/802.3.html)
Contributes to:	--

3.21.3 [BRF00285] Enable the PDU Router to communicate via TCP/IP

ID:	BRF00285
Initiator:	BMW
Date:	21.01.2008
Short Description:	Enable the PDU Router to communicate via TCP/IP
Importance:	**high** | medium | low
Description:	Enable the PDU router to send and receive I-PDUs via a TCP/IP protocol

	suite stack.
Rationale:	Announced ISO Requirements for Ethernet Car Access
Use Case:	- Diagnostic access via TCP/IP - - Flash programming via TCP/IP
Dependencies:	BRF00286 (Support Ethernet as an additional communication medium) BRF00285 (Enable the PDU Router to communicate via TCP/IP)
Conflicts:	
Supporting Material:	RFC 1122, Requirements for Internet Hosts – Communication Layers, IETF (http://www.ietf.org/rfc/rfc1122.txt?number=1122) IEE 802.3-2002 (Ethernet) (http://standards.ieee.org/getieee802/802.3.html)
Contributes to:	--

3.21.4 [BRF00286] Support Ethernet as an additional communication medium

ID:	BRF00286
Initiator:	BMW
Date:	21.01.2008
Short Description:	Support Ethernet as an additional communication medium
Importance:	**high** | medium | low
Description:	Enable Ethernet to be used by the Autosar Com Stack.
Rationale:	- Announced ISO Requirement for Ethernet Car Access. - Use of standard computer equipment for logging and debugging.
Use Case:	- Diagnostic Access - Flash - Support Log and Trace functionalities
Dependencies:	--
Conflicts:	--
Supporting Material:	RFC 1122, Requirements for Internet Hosts – Communication Layers, IETF (http://www.ietf.org/rfc/rfc1122.txt?number=1122) IEE 802.3-2002 (Ethernet) (http://standards.ieee.org/getieee802/802.3.html)
Contributes to:	--

3.21.5 [BRF00287] Implementation of Diagnostic Communication over IP

ID:	BRF00287
Initiator:	Daimler
Date:	2008.01.30
Short Description:	Implementation of a communication stack (like FlexRay TP or CAN-TP) for Diagnostic communication over TCP/IP (DoIP) based on an ISO standard currently under development in ISO TC22/SC3/WG1/TF3
Importance:	**High**
Description:	This feature shall allow for a standardized implementation of the requirements defined in the ISO-standard currently under development in ISO TC22/SC3/WG1/TF3. This communication protocol stack will be similar to the current implementation of ISO15765-2 CAN transport protocol or the upcoming ISO-standard based FlexRay transport protocol and should integrate into the AUTOSAR architecture using PDU-IDs and therefore will be accessible through the PDU-Router interfaces. As the new DoIP-standard will slightly differ from the aforementioned transport protocols for CAN and FlexRay it must be ensured during definition

	that all necessary interfaces, services and configuration options are defined in order to allow for seamless integration of this new DoIP protocol.	
Rationale:	Like ISO15765-2 and FlexRayTP the new DoIP-standard will be an ISO-standard most likely to be referenced by future emissions legislation to be used for diagnostic communication between test equipment and vehicles. Like with all ISO-standards a common implementation will improve stability and software quality and should be integrateable into future ECU software architecture which will most likely by AUTOSAR-based	
Use Case:	- Fast flash re-programming - Remote vehicle diagnostic communication over computer networks and the internet - Extendability of diagnostic communication on additional future physical layers (e.g. Ethernet, WLAN, GPRS, etc.)	
Dependencies:	This feature request bases on the BMW feature request for TCP/IP and Ethernet support (BRF00284)	
Conflicts:		
Supporting Material:	IETF RFC 791	Internet Protocol - DARPA Internet Program – Protocol Specification (September 1981)
	IETF RFC 2460	Internet Protocol, Version 6 (IPv6) Specification (December 1998)
	IETF RFC 793	Transmission Control Protocol - DARPA Internet Program - Protocol Specification (September 1981)
	IETF RFC 768	User Datagram Protocol (August 1980)
	IETF RFC 2131	Dynamic Host Configuration Protocol (March 1997)
	Protocol Numbers	Protocol numbers, IANA, http://www.iana.org/assignments/protocol-numbers (last updated 28 March 2006)
	IETF RFC 826	An Ethernet Address Resolution Protocol (November 1982)
	IEEE 802.3	Collection of IEEE standards defining the physical layer and data link layer of wired Ethernet
	IETF RFC 792	Internet Control Message Protocol DARPA Internet Program - Protocol Specification (September 1981)
	IETF RFC 1122	Requirements for Internet Hosts - Communication Layers (October 1989)
	IETF RFC 1533	DHCP Options and BOOTP Vendor Extensions (October 1993)
	IETF RFC 2373	IP Version 6 Addressing Architecture (July 1998)
	IEEE EUI-64	"Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", http://standards.ieee.org/db/oui/tutorials/EUI64.html, March 1997.
	IETF RFC 2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification (December 1998)
	IETF RFC 3315	Dynamic Host Configuration Protocol for IPv6 (DHCPv6) (July 2003)
	ISO/DIS 15031-5.4	Road vehicles — Communication between vehicle and external equipment for emissions-related diagnostics — Part 5: Emissions-related diagnostic services (Date: 2004-12-10)
	ISO 3779	Road vehicles- Vehicle identification number (VIN) - Content and structure (Current stage 90.93, Stage date:

		2003-09-30)
	Port Numbers	Port Numbers , IANA, http://www.iana.org/assignments/port-numbers (last updated 06 June 2006)
	IETF RFC 3330	Special-Use IPv4 Addresses (September 2002)
	IETF RFC 3927	Dynamic Configuration of IPv4 Link-Local Addresses (May 2005)
	IETF RFC 2893	Transition Mechanisms for IPv6 Hosts and Routers (August 2000
Contributes to:	--	

3.22 Time Determinism Concept

3.22.1 [BRF00120] Provision of a synchronized time-base within a cluster

ID:	BRF00120
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Provision of a synchronized time-base within a cluster
Importance:	High
Description:	AUTOSAR shall provide a synchronized time-base for a set of ECUs within a network cluster.
Rationale:	1/ To enable distributed SW-Cs to synchronize activities 2/ To detect and compensate for the incorrect clock of one of the ECUs 3/ For deterministic behavior.
Use Case:	Four SW-Cs on four ECUs read wheel speed at the same time, for brake controlling algorithm.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1. AUTOSAR can fulfill this requirement for systems using FlexRay or TTCAN time synchronization functionality. On other networks (e.g. using CAN) it will be more difficult to fulfill this requirement. 2. It is not constrained which networks shall be used. However, if a given network is used (e.g. CAN), then there shall be a compatible synchronization mechanism. 3. In AUTOSAR R4.0 support will be limited to FlexRay and TTCAN clusters. The extensions necessary to support this feature within CAN and LIN clusters are deferred to a later phase.
Contributes to:	--

3.22.2 [BRF00121] Runtime timing protection and monitoring

ID:	BRF00121
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Runtime timing protection
Importance:	High
Description:	AUTOSAR shall provide statically configured runtime timing protection and monitoring. This includes monitoring that tasks are dispatched at the specified time, meet their execution time budgets, and do not monopolize OS resources.
Rationale:	To guarantee that safety-related functions will execute within their timing constraints. Tasks monopolizing the CPU shall be detected and handled (like heavy ECU load, many interrupt requests).
Use Case:	If deadline of a task is not fulfilled, then it may be restarted or an error is reported.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ Monitoring of task execution detects scheduler misbehavior (i.e. deviations from real-time);

	2/ As runnables are mapped to tasks, runnable monitoring can be done either in a cumulative manner or by assigning single runnables to tasks in ECU configuration.
Contributes to:	--

3.22.3 [BRF00122] Support for timing constraints

ID:	BRF00122
Initiator:	AUTOSAR Safety Team
Date:	09.05.2007
Short Description:	Support for upper bounds on timing.
Importance:	High
Description:	It shall be possible to develop implementations based on AUTOSAR with verifiable timing constraints on jitter, latency and execution time. This means that task and communication scheduling strategies shall not contradict this. The requirement relates to task scheduling, communication scheduling and responsiveness to external events.
Rationale:	--
Use Case:	--
Dependencies:	BRF00121
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.22.4 [BRF00123] Responsiveness to external events

ID:	BRF00123
Initiator:	AUTOSAR Safety Team
Date:	09.05.2007
Short Description:	Responsiveness to external events
Importance:	High
Description:	AUTOSAR shall enable the use of external events as an initiator for scheduling.
Rationale:	As certain external events require a timely response to ensure correct behavior these events must be able to initiate tasks.
Use Case:	Schedules driven by ticks calculated from angles of an engine’s crankshaft.
Dependencies:	--
Conflicts:	--
Supporting Material:	External events include IO and interrupts
Contributes to:	--

3.22.5 [BRF00125] Monitoring of local time

ID:	BRF00125
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Monitoring of local time
Importance:	High
Description:	AUTOSAR shall provide a mechanism that monitors ECU local time.

Rationale:	This is a necessary basis for deterministic execution of safety functions and for detection of failures of the system by safety integrity functions, within the guaranteed time intervals.
Use Case:	The local time is monitored to guarantee the correct timing of the safety-related runnables on the ECU.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ This measure normally require an independent clock. This may be implemented with a HW watchdog. Alternatively, a different ECU with its local time could be used as a watchdog. Yet another solution could be to use an ADC and capacitor.
Contributes to:	--

3.22.6 [BRF00126] Services for synchronization of SW-Cs

ID:	BRF00126
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Services for synchronization of SW-Cs
Importance:	High
Description:	AUTOSAR shall provide mechanisms enabling SW-Cs on the same or different ECUs to synchronize their behavior
Rationale:	To enable runnables to respect their timing constraints.
Use Case:	1/ Two runnables must read data from a sensor in the same time window so that later they can vote on them; 2/ Two distributed SW-Cs (on different ECUs) perform synchronization.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.22.7 [BRF00127] Services for accessing to synchronized time-bases

ID:	BRF00127
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Services for accessing to both local and global time
Importance:	High
Description:	AUTOSAR shall provide a service to access synchronized time bases, available to BSWMs and SWC-s.
Rationale:	To enable SWC-s to perform time-dependent actions, and in particular synchronization and monitoring.
Use Case:	A safety-related function may need to time the execution of a particular operation, or it may need to know exactly how much time has elapsed since a previous event. This timing information may also be compared or calculated with another task from another ECU and in order to achieve this both tasks must be using the same time-base.
Dependencies:	--
Conflicts:	--

Supporting Material:	Notes: 1/ Most safety related functions will be scheduled deterministically which means that they know exactly how much time has elapsed since it last started to run. However, there may be situations where more accurate timing is required within a task itself, or to help a task synchronize with another task on another ECU.
Contributes to:	--

3.22.8 [BRF00278] Sync AUTOSAR OS with Global Time from providing bus system in a well-defined way

ID:	BRF00278
Initiator:	BMW
Date:	31.01.2008
Short Description:	Sync AUTOSAR OS with Global Time from providing bus system in a well-defined way
Importance:	high | **medium** | low
Description:	It shall be possible to sync the AUTOSAR OS with the Global Time from providing bus system in a well defined and fast way
Rationale:	- For AUTOSAR Release 3.0, it is up to the implementer to write a “glue code” which is not a proper solution
Use Case:	- Enabling applications to run their tasks synchronous to the Global Time from providing bus system
Dependencies:	AUTOSAR OS
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.23 XCP for AUTOSAR Concept

3.23.1 [BRF00279] Trigger Configuration of FR CC Buffer with non-static Configuration

ID:	BRF00279
Initiator:	BMW
Date:	31.01.2008
Short Description:	Trigger Configuration of FlexRay CC Buffer with non-static Configuration
Importance:	**high** | medium | low
Description:	It shall be possible to perform a CC Buffer reconfiguration at runtime (i.e. it is not being defined during system configuration time) The set of hardware-independent configuration parameters consists of the following items: - Identifier of the CC Buffer - Direction: Transmission or Reception - FlexRay Slot Number - Cycle Counter Offset - Cycle Counter Repetition - FlexRay Channel - Payload Length of FlexRay Frame - FlexRay Header CRC
Rationale:	- Dynamic bandwidth assignment to the XCP slaves by the XCP master.
Use Case:	- Manipulation of parameters for adjustment purpose (e.g. chassis suspension, motor management,..)
Dependencies:	FrIf, FrDrv
Conflicts:	--
Supporting Material:	http://www.asam.net/doc_int/getfile/getfile.php?id=376
Contributes to:	--

3.23.2 [BRF00280] AUTOSAR BSW XCP Modules

ID:	BRF00280
Initiator:	BMW
Date:	30.01.2008
Short Description:	AUTOSAR BSW XCP Modules
Importance:	**high** | medium | low
Description:	XCP shall be an AUTOSAR BSW as it was already shown in the first AUTOSAR architecture.
Rationale:	- Dynamic bandwidth assignment to the XCP slaves by the XCP master - Possibility to include XCP into the ECU configuration process
Use Case:	- Manipulation of parameters for adjustment purpose (e.g. chassis suspension, motor management,..)
Dependencies:	- ECU configuration tool - BRF “Trigger Configuration of FlexRay CC Buffer with non-static Configuration”

Conflicts:	--
Supporting Material:	http://www.autosar.org/download/AUTOSAR_LayeredSoftwareArchitecture.pdf http://www.asam.net/doc_int/getfile/getfile.php?id=376
Contributes to:	--

3.24 NM Coordination Concept

3.24.1 [BRF00256] NM Coordinator should support coordination to any kind of AUTOSAR busses

ID:	BRF00256
Initiator:	FMC
Date:	25.01.2008
Short Description:	NM Coordinator should support coordination to any kind of AUTOSAR busses
Importance:	**high** | medium | low
Description:	The "Generic Network Management Interface" has to be as generic as possible to support coordination/synchronization between networks.
Rationale:	The "Generic Network Management Interface" has to support coordination/synchronization between networks of any kind, except of those Bus-Specific NM's that are not included in the AUTOSAR. Such shall not be in the responsibility of AUTOSAR, but performed as extending the functionality of standard components by the OEM.
Use Case:	Coordinated shutdown of multiple networks. Synchronized shutdown of a multiple network. Improvement of the shutdown algorithm.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.24.2 [BRF00271] NM Coordinator should support NM Gateway to FlexRay

ID:	BRF00271
Initiator:	GM
Date:	31.01.2008
Short Description:	NM coordinator should support NM gateway to FlexRay
Importance:	**high** | medium | low
Description:	Current NM Coordinator concept is not viable when FlexRay is one of the coordinated subnets. Two basic situations can be considered: 1. Shutdown coordination between one CAN subnet and one FlexRay subnet, and 2. Shutdown coordination between two FlexRay subnets. The rest of this paper focuses primarily on the former, but the latter is briefly addressed at the end.
Rationale:	
Use Case:	Synchronized shutdown needs occur between FlexRay clusters and CAN clusters and also between FlexRay and FlexRay clusters
Dependencies:	Network Management Interface, CAN NM, FlexRay NM
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.24.3 [BRF00274] FlexRay Network Management Scheduling Timing Window Relief

ID:	BRF00274
Initiator:	GM
Date:	31.01.2008
Short Description:	FlexRay Network Management Scheduling Timing Window Relief
Importance:	**high** | medium | low
Description:	The timing window to schedule the NM task when using the Static segment NM Vector hardware service is too constrained and a timing window relief is desired. Particularly, there is a dependence between when a NM task can be scheduled and when the NM message in transmitted in a slot in the same cycle. A timing window relief could aim at the following two windows: a) The NM Task could be scheduled anywhere in the Communication Cycle. b) The NM Task could be scheduled anywhere in the Static segment of the Communication Cycle.
Rationale:	
Use Case:	1. Network Management need not run every FlexRay communication cycle saving processing time and resources. 2. Less stringent constraints on scheduling Network Management main function and transmission slots.
Dependencies:	Error Handling, DET, DCM, FlexRay Driver, FlexRay Interface.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.25 Functional Diagnostics of SWC Concept

3.25.1 [BRF00027] Functional Diagnostics of SWC

ID:	BRF00027
Initiator:	AUTOSAR PL
Date:	03.05.2007
Short Description:	Functional diagnostics of SWC
Importance:	high | medium | **low**
Description:	Each SWC must support interfaces for the diagnostic BSWC to access diagnosis relevant data. This includes the following elements for each of which a separate port must be available in order for the DSP to get access to these elements: - signals (e.g. sensor values, internal states) which shall be read via diagnostic service 0x22 in DCM - Parameter values which shall be readable and writeable via diagnostic service 0x22 and 0x2E. - actuators the current state of which shall be readable and adjustable via the diagnostic services 0x22 and 0x2F special routines which can be started, stopped and results of the execution requested via diagnostic service 0x31
Rationale:	Diagnostics are important for engineering, manufacturing and service to troubleshoot problems, verify the assembly process and successfully identify problems and replace faulty components in the dealerships. In the past the diagnostic application accessed the necessary data directly by referring to the memory location of the relevant variables/elements which is not wanted in the AUTOSAR design. Thus all features defined in "Description" must be supported via separate interface (ports) in each SWC as necessary.
Use Case:	- Access to diagnostic data of a system in engineering environment - Manufacturing area and service dealerships to develop - Assembling - Troubleshoot and repair a vehicle with complex software functions.
Dependencies:	DCM, DEM
Conflicts:	
Supporting Material:	ISO14229-1 , ISO15031-5/SAEJ1979, ISO15031-6/SAEJ2012, CCR1968.2, EU 70/220/EWG
Contributes to:	--

3.25.2 [BRF00229] Decentralized modular diagnostic configuration of SW-Cs

ID:	BRF00229
Initiator:	Daimler
Date:	25.01.2008
Short Description:	Decentralized modular diagnostic configuration of SW-Cs
Importance:	**high** | medium | low
Description:	Each SW-C must provide additional diagnostic configuration information for the SW-C, DEM & DCM to be able to generate ports to be connected between these modules in order to allow for diagnostic data to be accessible through the DCM. The following elements need to be provided by each SW-C and interfaces

	for these elements need to be specified: 1. Current values (read, e.g. sensor readings, status information) 2. Parameters (read/write, e.g. configuration, adjustments, FTL functions) 3. I/O controls (read, temporarily controllable, e.g. actuator overriding) 4. Routines (start able, stoppable, e.g. self-calibration or lengthy tests) 5. Events (announced by SW-C, e.g. faults or occurrences) 6. Modes & state controls (read, activate, e.g. SW-C specific limp-home modes)
Rationale:	Because of decentralized configuration & interface requirements each SW-C shall provide and implement diagnostic interfaces to allow code generation and port connection in the DCM (DSP) and DEM. A diagnostic design guideline should define these interfaces for each SW-C (e.g. new document and/or update of SW-C template).
Use Case:	- Use-case example: o As of today functions and associated diagnostics are developed by several parties. Thus for each function and its diagnostic monitors (e.g. torque management in an engine controller) the diagnostic capabilities are defined separately and will not necessarily be coordinated during development. o System integration and combination of diagnostics for accessibility through DCM and DEM requires that the individual functions and diagnostic features are connected to be compiled as a complete diagnostic system (which is in case of OBD2 certification relevant.) - Use-case summary: o develop decentralized modular software and its diagnostics without permanent interaction with other SW-Cs developers o Combine modules and extract module-specific diagnostic data o link diagnostic data from SW-Cs to DCM and DEM
Dependencies:	WP General Methodology and Configuration
Conflicts:	--
Supporting Material:	ISO 14229-1
Contributes to:	--

3.26 FlexRay Network Reliability Concept

3.26.1 [BRF00302] FlexRay Transmission Completion Confirmation

ID:	BRF00302
Initiator:	Toyota and Bosch
Date:	15.06.2008
Short Description:	Transmission Completion Determination
Importance:	**high** | medium | low
Description:	Errors on a bus are determined for transmission completion based on the detection information on transmission and SW.
Rationale:	Enhancement of FlexRay network reliability
Use Case:	FlexRay Error handling
Dependencies:	Fr, FrTrcv
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.26.2 [BRF00303] FlexRay Transmission Timeout Handling

ID:	BRF00303
Initiator:	Toyota and Bosch
Date:	15.06.2008
Short Description:	Transmission Time-out Handling
Importance:	**High** | medium | low
Description:	When the transmission of dynamic frame requested by COM is not completed after the fixed time passing, the information for COM to cancel the transmission of the frame automatically is sent.
Rationale:	Enhancement of FlexRay network reliability
Use Case:	FlexRay Error handling
Dependencies:	FrIf
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.26.3 [BRF00304] FlexRay Reception Completion Confirmation

ID:	BRF00304
Initiator:	Toyota and Bosch
Date:	15.06.2008
Short Description:	Reception Completion Determination
Importance:	**high** | medium | low
Description:	Even if an error on a bus is detected on reception, the judgment of the frame as normal allows the determination that the reception has been completed.
Rationale:	Enhancement of FlexRay network reliability
Use Case:	FlexRay Error handling
Dependencies:	Fr
Conflicts:	--

Supporting Material:	--
Contributes to:	--

3.26.4 [BRF00305] FlexRay Payload Length Check

ID:	BRF00305
Initiator:	Toyota and Bosch
Date:	15.06.2008
Short Description:	Payload Length Determination
Importance:	**high** | medium | low
Description:	The handling for the case where the assumed payload length differs from the actually received payload length can be specified.
Rationale:	Enhancement of FlexRay network reliability
Use Case:	FlexRay Error handling.
Dependencies:	Fr
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.26.5 [BRF00306] FlexRay Hardware Check

ID:	BRF00306
Initiator:	Toyota and Bosch
Date:	15.06.2008
Short Description:	Hardware Checking
Importance:	**high** | medium | low
Description:	To detect the hardware failure of the local node. The following three types of hardware trouble are detected: (1) Memory abnormality related to the communication of a microcomputer. (2) Abnormality detected by BD - Low voltage detection - Bus disconnection/short detection - Error signal line disconnection/short detection - TxD disconnection - TxEN disconnection/anchoring - Over-temperature - Overcurrent - GND disconnection (3) CC register/anchoring trouble of buffering in CC
Rationale:	Enhancement of FlexRay network reliability
Use Case:	FlexRay Error handling
Dependencies:	RamTest, Fr, FrSm, FrTrcv
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.26.6 [BRF00307] FlexRay Reset/Reinitialization

ID:	BRF00307
Initiator:	Toyota and Bosch
Date:	15.06.2008

Short Description:	Resetting/reinitializing
Importance:	**high** | medium | low
Description:	To perform resetting/reinitializing by the following unit in accordance with the instruction from application. (1) CC (2) BD (by channel) (3) Memory related to the communication of microcomputer
Rationale:	Enhancement of FlexRay network reliability
Use Case:	FlexRay Error handling
Dependencies:	FrIf, FrTrcv, FrSm, ComM
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.27 TTCAN Concept

3.27.1 [BRF00312] Introduction of TTCAN into AUTOSAR [accepted]

ID:	BRF00312
Initiator:	Bosch
Date:	10.09.2008
Short Description:	TTCAN Time Triggered extension to CAN
Importance:	high | **medium** | low
Description:	The CAN protocol is extended with a time triggered scheduler enabling time triggered CAN messages. TTCAN is an absolute superset to CAN without any functional changes in CanSM, CanTP and CanNM. Superset means, that TTCAN implies also all CAN functionalities. I.e. a TTCAN stack can serve both a CAN and a TTCAN bus.
Rationale:	CAN is an event driven communication system. With the introduction of time triggered extension CAN may also be used as a time triggered bus system. A coherent communication system from FlexRay over CAN to LIN with a common time base can be realized. Busloads close to 100% may be achieved increasing the bandwidth of current CAN communication. The error handling concept of the underlying CAN protocol extended with predictable communication supports safety relevant applications.
Use Case:	- Distributed control application with requirements for low latency and low jitter. - Sub-bus to other time-triggered protocols, e.g. FlexRay - X-by-wire applications - Crank-shaft synchronous communication, e.g. motor control unit
Dependencies:	CAN
Conflicts:	
Supporting Material:	The time scheduler is part of the hardware (TTCAN controller). Additional parameters are necessary to support its configuration. TTCAN is standardized in ISO 11898-4.
Contributes to:	--

3.28 Debugging Concept

3.28.1 [BRF00152] BSW Variables becomes accessible by external Debuggers

ID:	BRF00152
Initiator:	Elektrobit
Date:	03.09.2007
Short Description:	BSW variables become accessible by external debuggers
Importance:	high | medium | **low**
Description:	To be debugged variables must be defined as global and published.
Rationale:	To allow the debugging module to determine the size of a variable, the type definition of the variable needs to be accessible by the main BSW header files.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	Replacing BRF00080 and BRF00081 So it is up to the BSW module, to select which variables can be debugged in a specific implementation. But there are still some requirements: - If a variable shall be debugged in a specific implementation, the name of the variable has to be documented. Also the "meaning" of the different values shall be documented. - If the BSW SWS specifies e.g. state variables together with the encoding of states, it would be nice, if those variables could be debugged, since they are not implementation specific and are really helpful for debugging. I think this is the only point, where we are really talking about specific variables. But also here, the debug module does not specify them, this is only a recommendation for BSW implementations.
Contributes to:	--

3.28.2 [BRF00083] ORTI comparable XML Module Description

ID:	BRF00083
Initiator:	Keil
Date:	12.06.2007
Short Description:	ORTI comparable XML module description
Importance:	high | **medium** | low
Description:	Each BSW module and the RTE shall create an additional XML description of information necessary to do symbolic debugging.
Rationale:	Allow host tools to display additional information like encoding of bits, meaning of values, comments etc.
Use Case:	Human readable display of debug data during debugging sessions
Dependencies:	Functional backwards compatibility to ORTI is requested
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.28.3 [BRF00084] Debugging-Extension of the M2 Meta Model

ID:	BRF00084
Initiator:	Keil
Date:	12.06.2007
Short Description:	Debugging-Extension of the M2 meta model
Importance:	high | **medium** | low
Description:	Debugging-Extension of the M2 meta model with functional backwards compatibility to ORTI.
Rationale:	Adaptation of ORTI to AUTOSAR
Use Case:	Prerequisite for porting all ORTI functionality to AUTOSAR. ORTI specifies functionality which is currently not covered by the meta model.
Dependencies:	AUTOSAR meta model, BRF00083
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.28.4 [BRF00085] Access to PduR for Debugging

ID:	BRF00085
Initiator:	Keil
Date:	12.06.2007
Short Description:	Access to PduR for debugging
Importance:	**high** | medium | low
Description:	Currently all messages are forwarded either to Com or Dcm by the PduR. The debugging module cannot use the PduR-Com interface or the PduR-Dcm interface to send or receive debugging messages. For a clean solution, an additional communication module “Debug” should be added above the PduR.
Rationale:	Integration of debugging in communication stack to access debugging target remotely.
Use Case:	Communication between debugging host and debugging target using the PduR to access the target (an other ECU!).
Dependencies:	An additional BSW module ‘Debugging Module’ exists and will use this additional interface.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.29 DLT Concept

3.29.1 [BRF00224] Allow monitoring of DEM, RTE and COM to improve diagnostics

ID:	BRF00224
Initiator:	BMW
Date:	14.12.2007
Short Description:	Allow monitoring of DEM, RTE and COM to improve diagnostics.
Importance:	high | **medium** | low
Description:	AUTOSAR should standardize the monitoring of the BSW modules COM, DEM and RTE to supervise the activities of SWCs and their interaction with the COM and the DEM. Currently the RTE allows this (via tracing) but for the DEM and the COM hooks shall be added.
Rationale:	In the current version of AUTOSAR the DCM and DEM take care of the diagnostics and implement several features to fulfill existing diagnostic standards like ISO14229 or ISO15765. These standards define diagnostics on an ECU level. They assume that the functionality is not spilt across several ECUs. With the upcoming widespread usage of SWC and their close interaction the relationship of setting DTCs in one ECU to signalize the failure of a distributed feature will be probably not enough to understand the misbehavior. Functional based diagnostics is trying to cover this topic, but within this field there is up to now no standard which can be used. By offering monitoring capabilities in the DEM, COM and RTE different types of functional diagnostics can be implemented.
Use Case:	New diagnostic concepts are currently considered which require these interfaces for their work.
Dependencies:	none
Conflicts:	none known
Supporting Material:	(currently) none
Contributes to:	--

3.29.2 [BRF00294] Standardized Log&Trace format/protocol

ID:	BRF00294
Initiator:	BMW/Fraunhofer ESK
Date:	30.01.2008
Short Description:	All log, trace and error data reported from an AUTOSAR system should have a standardized format. This should be concerned by a standardized high level protocol for transmitting, and a format for data storage.
Importance:	**high** | medium | low
Description:	A specified binary format should be defined, witch covers all requirements of log and trace mechanisms. So some information about the source, and the context (like timestamp, number of runnable …) should be transmitted to cover some requirements about the filtering and traceability. Also the transmitted data (payload) should have a standardized format, to interpret them later on in the correct way.
Rationale:	Since logging and tracing is an important mechanism for testability and proofing product quality, it is necessary to standardize the stored and transmitted data. This is important for archiving, comparing and analyzing of log or trace data. Also it is possible, to build common tools to interpret the

	incoming data.
Use Case:	- SWC sends a log - DLT convert it to the DLT-Packet-Format - DLT sends the packet over an interface to a data storing client (PC) - The stored data of different ECU's are interpreted by the client - Logs from different ECU's can be merged to understand relationship of behavior from distributed applications.
Dependencies:	DLT
Conflicts:	
Supporting Material:	
Contributes to:	--

3.29.3 [BRF00295] DET Trace interface for Log&Trace

ID:	BRF00295
Initiator:	BMW/Fraunhofer ESK
Date:	30.01.2008
Short Description:	DET should forward its trace events to the DLT.
Importance:	high | **medium** | low
Description:	The DET receives trace events from errors from the BSW and SWC during debugging time. If a DLT module exists, these events should be forwarded to the DLT to collect logs and traces only in one instance.
Rationale:	To have an overview of all log, trace and error messages and to set all of them in the correct context, it is important to have all these messages and events in one list (context). Also it is not practicable to use more than one mechanism to report errors, logs and traces to a debugging interface. So all these sources should be routed to the DLT
Use Case:	- in a debugging scenario, an SWC or BSW Module uses the DET interface to trace an error - this error is forwarded by the DET to the DLT - the DLT turns these events in the DLT format and sends it over the debugging interface, together with all the other logs and traces
Dependencies:	DET, DLT
Conflicts:	BRF00224 includes the same requirement
Supporting Material:	
Contributes to:	--

3.29.4 [BRF00296] RTE/VFB Trace interface needed for Log&Trace

ID:	BRF00296
Initiator:	BMW
Date:	30.01.2008
Short Description:	RTE should provide an interface for trace of RTE/VFB calls for DLT. It should also be possible to trace the VFB during the production phase.
Importance:	high | **medium** | low
Description:	RTE provides at the moment the possibility to trace the VFB. No mechanism is specified, how the RTE can be configured, that a defined BSW module can receive the trace calls. It should be possible, that one or more BSW modules can receive VFB trace information. If possible, RTE should provide a mechanism to enable, disable traces during runtime of groups of RTE interfaces. RTE ports/interfaces should be

	grouped by SWC or other grouping mechanisms.
Rationale:	Debugging and DLT requires an RTE trace interface in parallel during debugging phase. In the production phase the DLT needs also the RTE trace. In the future more and more applications will be integrated in one ECU. As a consequence the communication between SWC is done locally and not over an external traceable bus like CAN or Flexray. It is important to trace the internal communication over VFB. To prevent heavy CPU load the traces should be grouped and enable, disabled individually during runtime.
Use Case:	- Trace of VFB interface - Access to VFB for advanced diagnostic services, see BRF00224
Dependencies:	RTE, Debugging, DLT
Conflicts:	BRF00224 combines several requirements
Supporting Material:	--
Contributes to:	--

3.29.5 [BRF00297] DEM Trace interface needed for Log&Trace

ID:	BRF00297
Initiator:	BMW
Date:	30.01.2008
Short Description:	The DEM should forward incoming events to the DLT interface.
Importance:	high | **medium** | low
Description:	The DEM should forward error events to the DLT.
Rationale:	To have an overview of all log, trace error messages and to set all of them in the correct context with the error events reported to the DEM, it is important to have all this messages and events in one list (context). This makes an analysis of the reported errors more efficient and gives a correct picture of the ongoing sequences, which report an error.
Use Case:	- a SWC or BSW module sets an DTC in the DEM - the DEM forwards this event to the DLT - the DLT turns these events in the DLT format and sends it over the a network interface to a DLT client (PC)
Dependencies:	DLT, DEM
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.29.6 [BRF00298] Log&Trace debug interface using diagnostic service

ID:	BRF00298
Initiator:	BMW
Date:	30.01.2008
Short Description:	The DCM should provide an interface for DLT to transport log and trace data over a diagnostic service.
Importance:	**high** | medium | low
Description:	DCM should provide an interface for DLT to send and receive data over the diagnostic service. Logging and tracing data are sent over this service and control requests for DLT are received. For this purpose the DCM should implement the ResponseOnEvent service

	(see UDS spec.). DCM should provide an interface for DLT to send data and receive control requests.
Rationale:	Log&Trace needs an interface to send Log&Trace data out of the ECU. DCM provides a bus independent access to the ECU over standardized diagnostic. This is available during production phase and provides a secured session control. Because log and trace messages are event triggered and the storage on the ECU is limited, these messages must be sent when they occur.
Use Case:	- Transmitting log and trace data during a diagnostic session - Advanced Diagnostic Tracing, optional over telematic services
Dependencies:	DCM, DLT
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.29.7 [BRF00300] Standardized interface/service Log&Trace for SWC

ID:	BRF00300
Initiator:	BMW
Date:	30.01.2008
Short Description:	AUTOSAR should provide a standardized service for Log&Trace of SWC
Importance:	**high** | medium | low
Description:	Logging and tracing is a debugging mechanism needed by a lot of ECU's. A Log&Trace service provides a mechanism to write Log&Trace entries from several SWC's into a centralized Log&Trace BSW module. The Log&Trace service buffers the Log&Trace entries, if necessary and provides a connection to a test client over a testing interface and/or a diagnostic service. Each Log&Trace entry will provide attributes, like timestamp, priority, context, the log message id and optional parameters. The format, how Log&Trace entries are stored and transported, should be standardized. The Log&Trace SW should be available during debugging and production phase. Individual traces should be enabled or disabled by trace levels, for example individually per SWC.
Rationale:	Each Tier1 uses its own mechanisms to provide such a logging interface, using some internal or external debugging interfaces. The format of the logging content also differs from ECU to ECU. When testing several ECU's, many different tools and parsers are needed to get the right information out of the logs. A standard Diagnostic Logging Component with standardized logging content should help to reduce the testing efforts and enable new automated testing mechanisms. Also the number of tools could be reduced by a standard logging content and protocol.
Use Case:	- Development support - Functional Testing - Test Automation - Test against models - Driver intensive tests - Advanced Diagnostic Tracing, optional over telematic services
Dependencies:	DLT
Conflicts:	--
Supporting Material:	--

Contributes to:	--

3.30 Memory related Concepts

3.30.1 [BRF00022] Modification of NVRAM Memory Access Concept

ID:	BRF00022
Initiator:	AUTOSAR PL
Date:	03.05.2007
Short Description:	Modification of NVRAM memory access concept
Importance:	high | medium | **low**
Description:	Modify the NVRAM access concept from block based access to data element access.
Rationale:	Currently the SWCs do access the NVRAM on a block base, however the assignment of data elements to blocks shall not be predetermined by the SWC. Prevent wasting of NVRAM space in case of small used sections of blocks
Use Case:	Allow the integrator to partition NVRAM data into blocks. Enables the portability to other HW platforms with small NVRAM-sections.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.31 Build System Enhancement Concept

3.31.1 [BRF00057] Memory Mapping Concept

ID:	BRF00057
Initiator:	Bosch
Date:	09.10.2007
Short Description:	Memory mapping concept
Importance:	**high** | medium | low
Description:	AUTOSAR shall define a memory mapping mechanism based on the SWS_Memory_Mapping from AUTOSAR Phase I. The mechanism shall support the definition of the abstract memory sections used by BSW modules and software-components in XML and the generation of memory mapping header files based on these descriptions.
Rationale:	With the currently defined mechanism, there is a high effort for integration of Software Components and BSW modules. Define a memory mapping concept which simplifies the integration.
Use Case:	Easy integration of MemMap files from different implementers.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.31.2 [BRF00077] Memory Mapping of SWCs

ID:	BRF00077
Initiator:	Continental
Date:	14.06.2007
Short Description:	Memory Mapping of SWCs
Importance:	high | medium | **low**
Description:	Memory Mapping of R2.1 is usable for BSW only because the used prefix <MSN> (the module abbreviation) is without meaning for Software Components.
Rationale:	Easy integration of Application Software
Use Case:	Integration of Software Components delivered as Source Code in common build environment.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.32 Support of Windowed Watchdog Concept

3.32.1 [BRF00159] Support of Windowed Wachdog

ID:	BRF00159
Initiator:	Valeo
Date:	03.05.2007
Short Description:	Support of Windowed Watchdog
Importance:	high | medium | **low**
Description:	The AUTOSAR BSW shall support hardware windows watchdogs and possible synchronization to refresh in the right time the hardware watchdog. A windows watchdog is a more restrictive watchdog, which requires the its trigger in a well defined time span (trigger not before t_1 and not later than t_2).
Rationale:	Windows watchdogs are well known in embedded systems and state of the art - AUTOSAR shall support this. Windows watchdogs are more restrictive than normal watchdogs and leads to safer systems.
Use Case:	Detection of a situation, where a task has lost its synchronization
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.33 Alarm Clock Concept

3.33.1 [BRF00196] Alarm Clock

ID:	BRF00196
Initiator:	GM
Date:	06.12.2007
Short Description:	Alarm clock
Importance:	**high** | medium | low
Description:	Provide a real-time clock for wakeup purposes. SWC can set or cancel Wakeup Alarms. SWC's must be able to request a controller to leave a Sleep state when they require functionality in the future. With future we mean several hours to weeks in the future. SWC's must be able to set and cancel alarms either relative or absolute to the current time and request the current time. When two alarm services are requested the earliest expiring alarm will take precedence. Besides the wakeup functionality the feature will also provide timing capability during the RUN mode
Rationale:	Allow internal wakeup requests based on time to leave a sleep state instead of only external I/O events.
Use Case:	- Engine Off Time - Battery Charge Monitoring - HVAC Auxiliary Engine heater. - Security/Theft
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.34 Enabling CDDs in the BSW architecture Concept

3.34.1 [BRF00225] Enabling CDDs in the BSW architecture

ID:	BRF00225
Initiator:	MBTech
Date:	10.01.2007
Short Description:	Enabling CDDs in the BSW architecture
Importance:	**high** | medium | low
Description:	Enabling implementation of CDDs in the BSW architecture. AUTOSAR stack allows CDDs to interface to all BSW modules. However, the BSW specifications often do not allow for this. This has either to be fixed, or the ability of CDDs to access BSW interfaces has to be restricted, otherwise it is not possible to write CDDs
Rationale:	Allow writing of CDDs as defined in the AUTOSAR architecture
Use Case:	Use cases for CDDs need not to be given. However, to state some current problems: - CDDs accessing MCAL (e.g. PWM, but call back routines of MCAL only call IOHWA, not a CDD) - CDDs accessing PduR (e.g. Debugging; but PduR only interfaces to Com or Dcm), CDDs accessing CanIf (e.g. OSEK NM or XCP, but there exists a parameter to only select PduR, CanTp or CanNm in CanIf) - new I/O busses besides SPI like USB etc.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.35 Concept for Libraries

3.35.1 [BRF00165] Integration of the HIS Crypto Functionality

ID:	BRF00165
Initiator:	BMW
Date:	18.10.2007
Short Description:	Integration of the HIS crypto functionality
Importance:	high | medium | **low**
Description:	It is necessary (and has been already planned) to provide some crypto functionality to the BSW and the SWCs.
Rationale:	For the realization of sophisticated and/or to be protected functionality crypto functionality is necessary. To avoid the costumer specific adaptation an integration of convenient functionality into the BSW is indicated.
Use Case:	- It shall be impossible to change some special parameters unauthorized - It shall be impossible to activate some special functionality unauthorized
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.35.2 [BRF00311] Standard AUTOSAR libraries

ID:	BRF00311
Initiator:	Continental
Date:	01.02.2008
Short Description:	Standard AUTOSAR libraries
Importance:	High
Description:	AUTOSAR consortium should define libraries for commonly used functions: fixed point math, interpolation, floating point, Bit handling, special functions. Only interfaces shall be defined. These functions may be used either in BSW modules and applicative SWC as pure function calls.
Rationale:	avoid multiplication of non standardized libraries
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.36 Bootloader Interaction Concept

3.36.1 [BRF00034] Bootloader Interaction

ID:	BRF00034
Initiator:	Freescale
Date:	06.09.2007
Short Description:	Bootloader interaction
Importance:	high | **medium** | low
Description:	AUTOSAR software may switch to bootloader mode during runtime in order to allow update software or parameters in flash memory. The bootloader may ramp up AUTOSAR software after finishing. It might be very helpful if AUTOSAR software and the bootloader could exchange certain data (reset reason, update counter, update occurred, etc.). For free running windowed watchdogs as present in certain SBCs, it is even necessary to synchronize the watchdog drivers in AUTOSAR software and the bootloader.
Rationale:	It is necessary to exchange data between AUTOSAR software and bootloader and to synchronize watchdog drivers in AUTOSAR software and bootloader.
Use Case:	Switching from AUTOSAR software to Bootloader on requests for software or parameter updates or diagnostics purposes
Dependencies:	--
Conflicts:	Potential conflicts with wake-up concept, MCU and WDG drivers
Supporting Material:	--
Contributes to:	--

3.36.2 [BRF00262] DCM shall support the Service EcuReset

ID:	BRF00262
Initiator:	Vector Informatik
Date:	2007.01.28
Short Description:	DCM shall support service EcuReset
Importance:	**high** | medium | low

Description:	The DCM has to force a shutdown on external tester request.
	There are different options to do that:
	`1 Hard Reset`
	`2 Key off/on Reset`
	`3 Soft-Reset`
	`4 Enable Rapid Power Shutdown`
	`5 Disable Rapid Power Shutdown`
	
	Hard Reset: is already supported in ASR 3.0. An additional requirement to WdM is, not to report an error due to an unexpected reset.
	Key off/on Reset: could be handled by the planned VehicleManager by simulating an ignition key off/on
	Soft Reset: shall be handled by EcuManager After the shutdown is forced and all runables are stopped the NvM has to write the Nv data. After writing the Nv data a positive response has to be sent before the ECU will be reset.
	En-/Disable Rapid Power Shutdown: The ECU has to enter a mode with constant power consumption. Can be achieved by going to sleep mode without any after run of the ECU.

Rationale:	Usually parts of this diagnostic service are required for mass production.

Use Case:	ECU reset to achieve a defined status of the ECU. ECU reset to load new calibration values. Forcing immediate shutdown to sleep mode to measure the power consumption of the vehicle to detect loose contacts.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.36.3 [BRF00263] DCM shall internally support the Jump to Flash-Bootloader

ID:	BRF00263
Initiator:	Vector Informatik
Date:	2007.01.28
Short Description:	DCM shall internally support the jump to flash-bootloader
Importance:	**high** | medium | low
Description:	Usually the diagnostic communication protocol (e.g. UDS) is used to jump to the bootloader. According HIS the data exchange between the bootlader and DCM has to be done in Nv-Memory. Therefore the Memory-Stack shall be extended to support at least two different configurations. One subset for the bootloader and at least one for the application. That the bootloader (subset configuration) can write also data into the flash, the FEE / EA have to copy the unknown blocks while flash sector-switch. This implies that the FEE / EA have to detect an ‘updated’ configuration that the unused or incompatible blocks can be deleted to free memory.
Rationale:	Not support by the current AUTOSAR architecture.
Use Case:	Necessary for the jump to bootloader
Dependencies:	WP Maintenance of Specifications
Conflicts:	FEE / EA do not support multiple configurations or accessing the same data from two applications.
Supporting Material:	--
Contributes to:	--

3.37 Multi Core Architectures Concept

3.37.1 [BRF00199] Real-time Capability & Predictability

ID:	BRF00199
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Real-time capability & Predictability
Importance:	**high** | medium | low
Description:	Each multi core operating system solution shall meet the requirements of automotive real-time systems. Moreover, the real-time behaviour shall be predictable. The real-time capability and predictability of the AUTOSAR single core OS concepts shall be taken as reference.
Rationale:	The real-time capability and predictability is state of the art for single core systems in the automotive domain. AUTOSAR OS and OSEK are examples for this.
Use Case:	--
Dependencies:	OS Specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.2 [BRF00200] Deadlock free mutual Exclusion

ID:	BRF00200
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Deadlock free mutual exclusion
Importance:	**high** | medium | low
Description:	Each multi core operating system solution shall support a deadlock free mutual exclusion mechanism between cores.
Rationale:	In a multi core system a mutual exclusion mechanism is needed to synchronise different cores. This mutual exclusion mechanism shall be deadlock free.
Use Case:	--
Dependencies:	OS Specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.3 [BRF00204] Multi Core System with one Core as intelligent Peripheral

ID:	BRF00204
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Multi core system with one core as intelligent peripheral
Importance:	**high** | medium | low
Description:	It shall be possible to use cores as intelligent peripheral. It should e.g. be able to perform specialized tasks in conjunction with I/O without an operating system.

Rationale:	This offers the possibility to reduce the interrupt load on the "main" core and to keep it free from high frequency I/O tasks.
Use Case:	--
Dependencies:	Potentially OS Specification, potentially on MCAL driver
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.4 [BRF00205] Multi Core System with one OS per Core

ID:	BRF00205
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Multi core system with one OS per core
Importance:	**high** | medium | low
Description:	It shall be possible to use multiple closely coupled CPU Cores for functionally independent subsystems (one subsystem per core), with independent operating systems for each core.
Rationale:	Provide means to make use of multiple closely coupled CPU cores for functionally independent subsystems with one OS per core.
Use Case:	Integration of independent applications on a multi-core system.
Dependencies:	OS Specification, potentially on MCAL driver
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.5 [BRF00206] Multi Core System with one OS controlling all Cores

ID:	BRF00206
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Multi core system with one OS controlling all cores
Importance:	**high** | medium | low
Description:	It shall be possible to use one common OS to manage multiple closely coupled CPU Cores.
Rationale:	Reasons to provide a solution with one common OS are: - Enables efficient parallelization of functions. - Supports sharing of peripherals - Upward and downward scalability in number of cores - Supports migration of strongly integrated single applications from single to - multi core
Use Case:	Applications (e.g. signal processing applications) with the need to achieve high performance computing via algorithm parallelization Multi core systems with common BSW Applications that grow beyond the boundary of the given number of cores (e.g. one) can easily utilize a higher number of cores (upward scalability). Applications designed for multiple cores can be stripped down (e.g. for low cost systems) to fewer (e.g. one) cores (downward scalability).

	Migration of engine control systems to multi core Integration of formerly separated applications into one multi core ECU
Dependencies:	OS Specification, potentially on MCAL driver
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.6 [BRF00207] Freedom from Deadlocks

ID:	BRF00207
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Freedom from deadlocks
Importance:	**high** | medium | low
Description:	All operating system mechanisms shall be designed in a way that their usage cannot cause deadlocks. Subfeature of BRF00206.
Rationale:	Prevent the application from deadlocks. Ensure real-time capability and predictability of the operating system.
Use Case:	Concept of resource sharing mechanism shall guarantee freedom from deadlocks.
Dependencies:	OS Specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.7 [BRF00208] Freedom from unbounded Blocking

ID:	BRF00208
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Freedom from unbounded blocking
Importance:	**high** | medium | low
Description:	All operating system mechanisms shall be designed in a way that there usage cannot cause unbounded blocking.
Rationale:	Ensure real-time capability and predictability of the operating system.
Use Case:	--
Dependencies:	OS Specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.8 [BRF00209] Service Compatibility to single Core Systems on one Core

ID:	BRF00209

Initiator:	Bosch
Date:	26.09.2007
Short Description:	Service compatibility to single core systems on one core
Importance:	**high** | medium | low
Description:	The behaviour of OS services (e.g. task activation) should be identical to single core systems when the originating and the manipulated object (e.g. a task) reside on the same core.
Rationale:	Known OS services for single core systems should behave identically in a multi-core system when used locally, i.e. without crossing core boundaries.
Use Case:	--
Dependencies:	OS Specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.9 [BRF00210] High Service Compatibility to single Core Systems across multiple Cores

ID:	BRF00210
Initiator:	Bosch
Date:	26.09.2007
Short Description:	High service compatibility to single core systems across multiple cores
Importance:	**high** | medium | low
Description:	OS services used across cores, i.e. where originating and manipulated object reside on different cores, should behave identical to their single core versions. Note that it is not possible to have exactly the same functionality for all services on a multi core system in all cases (example: Interrupts disabling), and that it is not useful to have exactly the same behavior on a multi core in all cases, (example: resource locking).
Rationale:	The user of OS services should not be confronted with different behavior in comparison to single core systems, where this is avoidable.
Use Case:	--
Dependencies:	OS Specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.10 [BRF00211] Static Assignment of Tasks to Cores

ID:	BRF00211
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Static assignment of tasks to cores
Importance:	**high** | medium | low
Description:	It shall be possible to assign tasks statically to cores.
Rationale:	The system integrator should be able to configure the system such that a given task is always executed on a core of his/her choice.
Use Case:	--
Dependencies:	OS Specification, Subfeature of BRF00206.

Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.11 [BRF00212] Activation of Tasks across Cores

ID:	BRF00212
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Activation of tasks across cores
Importance:	**high** | medium | low
Description:	It shall be possible that software running on a given core activates tasks on another core. This shall apply to all possibilities to activate tasks, i.e. not only by using the ActivateTask() API call.
Rationale:	The offline relocation of tasks between cores in different projects (e.g. low cost and high-end vehicles) shall be possible without reprogramming all task activations (especially in BSW components). Moreover, it shall be possible for the system integrator to assign sub-functionality to a core with free processing power.
Use Case:	Usage of third-party SW delivered as object code.
Dependencies:	OS Specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.12 [BRF00214] Resources across Cores

ID:	BRF00214
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Resources across cores
Importance:	**high** | medium | low
Description:	The single core mechanism for resources shall be extended to a multi core version that allows to form a critical section between tasks on different cores. This extended resource mechanism shall handle deadlocks appropriately and avoid priority inversion. If it is not possible to achieve the required behaviour by extending the existing mechanism an alternative mutual exclusion mechanism shall be provided.
Rationale:	The offline relocation of tasks between cores in different projects (e.g. low cost and high-end vehicles) shall be possible without introducing a completely new mechanism for mutual exclusion . Moreover, it shall be possible for the system integrator to assign sub-functionality to a core with free processing power. However it is unclear whether the required behaviour can be efficiently modeled and therefore at least a mutual exclusion mechanism that works across cores is needed.
Use Case:	Sharing peripherals between tasks on different cores. Sharing data between tasks on different cores.
Dependencies:	OS specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.13 [BRF00215] Static Assignment of Interrupts to Cores

ID:	BRF00215
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Static assignment of interrupts to cores
Importance:	**high** | medium | low
Description:	It shall be possible to assign IRQs statically to cores.
Rationale:	The system behavior is not predictable, if the IRQs are not statically assigned to cores.
Use Case:	--
Dependencies:	OS specification, potentially MCAL driver, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.14 [BRF00216] Disabling/Enabling Interrupt API Calls work locally

ID:	BRF00216
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Disabling/enabling interrupt API calls work locally
Importance:	**high** | medium | low
Description:	All disabling and enabling interrupt API calls shall take effect on the local core only.
Rationale:	In general disabling interrupts is not sufficient to build a critical section on a multi-core system. Moreover, making the disable/enable interrupt API calls effective on all cores causes unnecessary performance degradation of the system.
Use Case:	--
Dependencies:	OS specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.15 [BRF00217] Event Mechanism shall work across Cores

ID:	BRF00217
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Event mechanism shall work across cores
Importance:	**high** | medium | low
Description:	It shall be possible to send an event to a task on a different core The single core event mechanism shall be extended to a multi core version such that it is possible to send events from one core to another.
Rationale:	The offline relocating of tasks between cores in different projects (e.g. low cost and high-end vehicles) shall be possible without introducing a new event mechanism.
Use Case:	--
Dependencies:	OS specification, Subfeature of BRF00206.

Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.16 [BRF00218] Offline Configurability of Number of Cores

ID:	BRF00218
Initiator:	Bosch
Date:	26.09.2007
Short Description:	Offline configurability of number of cores
Importance:	**high** | medium | low
Description:	The number of cores that the operating system manages shall be configurable.
Rationale:	The operating system specification shall not be limited to a certain number of cores.
Use Case:	Use of the operating system in projects with different numbers of cores.
Dependencies:	OS specification, Subfeature of BRF00206.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.17 [BRF00220] Initialization and Startup

ID:	BRF00220
Initiator:	Vector
Date:	29.11.2007
Short Description:	Defined Initialization/Startup of the OS
Importance:	**high** | medium | low
Description:	Initialization/Startup of the OS has to be synchronized
Rationale:	In the case that cores run independent Oss it is necessary to define the Startup. E.g. Tasks can not be started across cores if one OS has not yet finished StartOS. Either a synchronization of StartOS is required or a special error handling has to be defined.
Use Case:	Startup of all multicore systems
Dependencies:	OS specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.18 [BRF00221] Controlled Data Exchange between Cores

ID:	BRF00221
Initiator:	Vector
Date:	29.11.2007
Short Description:	Controlled data exchange between cores
Importance:	**high** | medium | low
Description:	A controlled mechanism to exchange data between cores has to guarantee data consistency independent of cache validation/invalidation strategies
Rationale:	Microcontrollers with cache invalidation in software have to be considered. Cache invalidation in software makes it necessary to define a data exchange

	API which covers cache invalidation. Possibly the API has to be made atomic
Use Case:	Every data exchange via shared memory
Dependencies:	OS specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.19 [BRF00222] Common Configuration

ID:	BRF00222
Initiator:	Vector
Date:	29.11.2007
Short Description:	Common configuration across several cores
Importance:	**high** | medium | low
Description:	Disjunctive Object Ids have to be generated if objects are to be addressed across cores
Rationale:	If e.g. Tasks are activated across cores Ids have to be unique across cores. This results in a common configuration and affects flashing/programming strategies.
Use Case:	Activating tasks or setting events across cores
Dependencies:	OS specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.37.20 [BRF00223] Inter Core Timer Synchronization

ID:	BRF00223
Initiator:	Vector
Date:	29.11.2007
Short Description:	Inter core timer synchronization
Importance:	high | **medium** | low
Description:	To run applications synchronized a synchronized time base is needed.
Rationale:	Necessity to synchronize tasks across cores in time. This can be done by several means. E.g. Alarms activating tasks across cores, by synchronizing counters or by using shared hardware timers. A standardized way has to be defined.
Use Case:	Timely synchronized applications
Dependencies:	OS specification
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.38 Error Handling Concept

3.38.1 [BRF00156] Specification of the Error Handling

ID:	BRF00156
Initiator:	Renault
Date:	07.09.2007
Short Description:	Specification of the error handling
Importance:	high | **medium** | low
Description:	Describe the dysfunctional behavior of BSW (detection of error, reaction to errors, requirements on other BSW to handle error conditions, etc.). The product of this concept is - a consistency between the error handling strategy of different modules - a description of the error handling strategies in the BSW
Rationale:	The AUTOSAR specifications define the functional behavior of BSW modules. This allows switching from an implementation to another of a BSW module (or a cluster of modules). However, some details of the dysfunctional behavior (detections, reactions, etc.) are missing to allow exchange of modules leading to the same error typology, error behavior.
Use Case:	- Description of the behavior of a failure during <MSN>_Init() - Exchange of BSW which support the same behavior to error conditions
Dependencies:	Support from WP Functional Safety and Processes Support from expert in the various BSW
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.38.2 [BRF00193] List of Standardized Errors

ID:	BRF00193
Initiator:	AUTOSAR WP Error Handling
Date:	12.11.2007
Short Description:	List of standardized errors
Importance:	high | **medium** | low
Description:	List existing errors and new errors that shall be handled by AUTOSAR architecture. (implementation specific errors are still allowed; errors detection can depend on HW) Describe/Define the behavior of AUTOSAR architecture when one of these errors is detected.
Rationale:	- Warranty that any BSW implementation will exhibit the same behavior in case of defined standard errors. - Make sure the AUTOSAR infrastructure provides the services for application SW to handle errors.
Use Case:	- The list of errors will exhibit new errors that need to be handled; inconsistency in the handling of different, but similar, errors. - This will add a more detailed specification of the modules behavior

	and permit a safer exchange of module implementation.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.38.3 [BRF00275] Error Handling Capabilities for Partitions

ID:	BRF00275
Initiator:	AUTOSAR WP Error Handling
Date:	31.01.2008
Short Description:	Add the capability to perform error handling for partitions, from the BSW as well as from the application level, i.e., issue requests to the OS to terminate and/or restart a partition.
Importance:	**high** | medium | low
Description:	Error recovery sometimes requires restart of SWCs in order to remove erroneous states in or shutdown of faulty SWCs. Currently, this can only be done at ECU level (ECU reset). To enable termination and restart of SWCs from the BSW and the application level, services are needed to gain access to OS-level services to perform terminations and restarts of SWCs. As SWCs are contained in partitions as error containment regions, the operations shall be based on partitions.
Rationale:	Currently the default recovery action for a failing SWC is to reset the ECU (for instance by a watchdog timer). For certain applications this might be too coarse grained and the possibility to restart individual SWCs or groups thereof is desired. Since an “application” may consist of several (possibly distributed) SWCs, recovery becomes application dependent and must be controlled above RTE. To stop error propagations, SWCs can be located in partitions, which then act as error containment regions. The partitions are subject to error handling mechanisms, including memory and timing protection. If an error is detected, the partition can be terminated or restarted before the error propagates outside the partition. Partitions allow logical grouping of SWCs that can be terminated/restarted together. Furthermore, application-level error handling is often application specific and can also be OEM specific (different strategies may be used). In order to implement different strategies, one may implement a special “Error Handling Manager” (at BSW level or at application level) in charge of monitoring applications in the vehicle and implementing the error handling strategies. Such a monitor could gather error information from e.g. the watchdog, DEM, etc. on the local ECU, and even information from other ECUs, and use this to perform proper error recovery actions. To be able to perform error handling in the form of termination and/or restart of partitions, OS-level support is needed. Restrictions in this access can be imagined. For example, one might only allow access to these capabilities to some special (“privileged”) types of SWCs (likely a configuration issue) that deal with application/system monitoring.
Use Case:	- Termination/restart of individual application/SWC contained in partitions - Reconfiguration of application (shutting down one or more partitions) - Application/OEM specific error handling through a dedicated “Error Handling Manager” implemented as a SWC

Dependencies:	The “Timing Determinism” concept depends on this feature (forcibly terminating a SWC) being available. This feature depends on the introduction of partitions.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.39 SRS Core Test

3.39.1 [BRF00185] Test modes

ID:	BRF00185
Initiator:	AUTOSAR WP Microcontroller Abstraction
Date:	05.12.2007
Short Description:	Test modes
Importance:	**high** | medium | low
Description:	The test modules (e.g RAM test, FLASH test, Core test…) shall provide foreground and background mode: - In background mode, the test is called periodically by a scheduler. - In foreground mode, the test is run on request of a higher level module.
Rationale:	Provide a flexible way of testing.
Use Case:	E.g. use foreground mode for executing a test at startup, use background mode for executing a test during normal operation. Test in foreground mode is accessible to external device (safety microcontroller) via e.g. RTE or complex device driver, depending on the overall ECU safety concept.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.40 SRS Flash Test

3.40.1 [BRF00186] Test Result Processing

ID:	BRF00186
Initiator:	AUTOSAR WP Microcontroller Abstraction
Date:	05.12.2007
Short Description:	Test result processing
Importance:	**high** | medium | low
Description:	For the test result processing, the following concepts shall apply: - result and reference value are compared within the test module - calling entity is responsible for processing the returned test metric
Rationale:	Support a caller, which can be an external microcontroller, holding a “golden reference value”.
Use Case:	Usage of an external safety microcontroller, which holds the reference value and thus is able to verify the test correctness.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.41 SW-C E2E Communication Protection Concept

3.41.1 [BRF00114] SW-C end-to-end communication protection

ID:	BRF00114
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	SW-C end-to-end communication protection
Importance:	High
Description:	Within this concept (feature), we define the extensions to RTE and configuration to support end-to-end safe communication between SW-Cs located on remote ECUs. End-to-end communication protection is a state-of-art in a big group of safety-related systems in different industries, including automotive. Currently, some existing network stacks provide a subset of mechanisms used by safety protocol (e.g. checksum). However, the purpose of these mechanisms are availability and fault tolerance, but not safety (FlexRay is partially an exception). Logically, the concept creates a layer between VFB and SW-Cs. This is realized by means of: 1/ safety protocol library – a set of stateless library functions that verify the communication (e.g. if a CRC of a message is correct or is it on time), and which are invoked by RTE or SW-Cs, 2/ introduction of additional configurable attributes (fields) for SW-C ports (e.g. port address), used by safety protocol library. The port attributes keep the state information of the communication, whereas the stateless library function does the checks. Thanks to these extensions, any inter-ECU communication can be possibly used to transmit safety-related data. The safety protocol will work on any network/bus that is supported by AUTOSAR, including CAN, LIN, SPI and FlexRay. Depending on: (1) reliability and type of a used network, (2) size and criticality of the transmitted data, and (3) fault tolerance of application; the protocol needs to be appropriately configured. The configuration involves selection of used mechanisms and mechanism strength (e.g. CRC8 vs CRC16). This is left to the integrator to choose. Moreover, depending on: (1) Communication model (client-server vs. sender-receiver), (2) Communication multiplicity (1:n vs 1:1 vs n:1); some mechanisms are or aren't present (e.g. there is no destination address in 1:n sender-receiver communication). There are no dependencies to any other concepts. In particular, we do not depend on "Communication Stack" concept.
Rationale:	1/ To detect and tolerate faults in RTE, communication software and other BSWMs, as well as in communication hardware.
Use Case:	SW-Cs located on remote ECUs, exchanging safety-related data.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.42 Memory Partitioning Concept

3.42.1 [BRF00115] SW-Cs grouped in separate user-mode memory partitions

ID:	BRF00115
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	SW-Cs grouped in separate user-mode memory partitions
Importance:	High
Description:	The feature defines the extensions of the OS and the RTE functionality that are necessary to support groups of SW-Cs running in separate user-mode memory partitions. The most important resulting AUTOSAR extension is the inter OS-Application communication (across boundaries of memory partitions). Further (smaller) extensions are in the configuration and error handling. Partitioning of BSW is not in the scope of the concept/feature – only SW-C is covered. With these extensions, it will be possible to setup protection boundaries prohibiting a propagation of some kinds of hardware and software faults. This is especially interesting when there are several SW-Cs on one ECU, and when SW-Cs have different ASIL or they come from different parties. This is also useful for debugging and testing of SW-Cs. Memory partitioning provides protection by means of restricting access to memory and memory-mapped hardware. Memory partitioning means that OS-Applications reside in different memory areas (partitions) that are protected from each other. In particular, code executing in one partition cannot modify memory of a different partition in an uncontrolled fashion, even by indirect means. Moreover, memory partitioning enables to protect read-only memory segments, as well as to protect memory-mapped hardware. Supervisor/user modes provide the protection by means of restricting the access to CPU. Currently, OS makes the notion of a partition being identified with the notion of the associated OS-Application. In other words, each OS-Application has its own memory partition, with separate stack, data and code. OS assumes (requires) an MPU for providing memory protection (by segmentation). Support for MMU (by paging) is not specified. However, there is no communication mechanism between OS-Applications offered. OS itself does not provide the communication between OS-Applications - instead, OS clearly delegates the communication between partitions (i.e. basic techniques for transferring data between protected memory regions) to RTE. RTE assumes its role, but does not provide these mechanisms yet. Therefore, inter OS-Application communication (i.e. communication between different OS-Applications within the same ECU) is the major missing functionality. Possible extensions of RTE communication modes (client-server, sender-receiver) will be sketched by this concept, so that they work not only intra-OS-Application, inter-ECU, but also inter OS-Application.
Rationale:	This prevents the following failure modes from propagating: 1. systematic software faults in SW-Cs (i.e. bugs in software, like buffer overflows, incorrect pointer arithmetic) 2. random hardware faults in SW-Cs (e.g. faults of address unit, faults in memory cells storing pointers)
Use Case:	The concept/feature enables the following combinations of SW-Cs on one ECU: SW-Cs of different ASIL

	SW-Cs from different vendors, SW-Cs under debugging/testing.
Dependencies:	There is a hardware dependency, which is already explicit in AUTOSAR OS. “SW-Cs grouped in separate user-mode memory partitions” is only possible on processors that provide hardware support for memory protection (MPU, MMU). Another feature ([BRF00275] Capability for Application Level SW-C Management (stop, start, restart)) is very useful for this feature, but not strictly required.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.43 Program flow monitoring Concept

3.43.1 [BRF00131] Logical Program Flow Monitoring

ID:	BRF00131
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Logical program flow monitoring
Importance:	High
Description:	Add logical program flow monitoring of SW-Cs and BSW modules by means of extension of Watchdog Manager. Logical monitoring of the execution sequence of a program enables the detection of errors that cause a divergence from the valid program sequence during the error-free execution of the application. An incorrect program flow occurs if one or more program instructions are processed either in an incorrect sequence or not even processed at all. During design phase the valid program sequences are identified and modeled. During runtime the component for Logical Monitoring of Program Sequence uses this model to supervise or monitor the proper execution of program sequences. In case a divergence is detected usually the system is reset. To reduce the overhead caused by logical monitoring of program sequence, in AUTOSAR it is possible to restrict the definition of Supervised Entities (SE) to safety-related tasks/runnables. At least those have to be monitored but non safety-related tasks can be monitored as well.
Rationale:	This enables to detect to the following faults: 1. Systematic software faults 2. Random hardware faults 3. Systematic hardware faults. Faults in execution of program sequences (i.e. invalid execution of program sequences) can lead to data corruption, process crashes, or fail-silence violations. Logical program flow monitoring is required/recommended/proposed by ISO 26262, IEC 61508, MISRA.
Use Case:	Example safety-related Software Modules: - Monitoring that important steps in SW-C's computation algorithm are executed.
Dependencies:	Other concepts depend on this feature, e.g. "Multi-microcontroller support", "Defensive behavior", "Time determinism"
Conflicts:	
Supporting Material:	It is important that the checking points are placed in the program correctly. This is done by the developer or by an application-level generator (both not in the scope of ATUOSAR). Logical monitoring of program flow can be defined in various ways, both using hardware and software resources. This concept proposes a method using both software and hardware: most of the work is done by Watchdog Manager BSW-M, and part of error handling (ECU reset) is done by a HW watchdog.
Contributes to:	--

3.44 BSWM Defensive behavior Concept

3.44.1 [BRF00128] Protection against Unauthorized Use of BSW

ID:	BRF00128
Initiator:	AUTOSAR Safety Team
Date:	02.11.2006
Short Description:	Protection against unauthorized use of BSW
Importance:	High
Description:	AUTOSAR shall protect BSWMs from unauthorized use by SW-Cs and other BSWMs.
Rationale:	To avoid SW-Cs from corrupting or interfering with AUTOSAR service calls of safety-relevant SW-Cs but also to prevent non-safety related BSWMs from corrupting or interfering with BSW use by safety-related BSWMs.
Use Case:	Only selected software modules are allowed to request the shutdown of ECU.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ AUTOSAR specifies access constraints to BSWMs. For example, interrupt routines have limited rights to call OS services; 2/ The access rules that are set at configuration time must be enforced at runtime.
Contributes to:	--

3.44.2 [BRF00129] Protection of Data

ID:	BRF00129
Initiator:	Safety Team
Date:	02.11.2006
Short Description:	Protection of data
Importance:	High
Description:	AUTOSAR shall protect its safety-related data in RAM and non-volatile memory against corruption.
Rationale:	To enable the AUTOSAR to handle its internal data in a safe manner.
Use Case:	Requestors of fault-tolerant data protection such as: 1/ ECU state manager: ECU state data; 2/ DEM, FIM: current errors detected.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ This requirement has impact on AUTOSAR architecture: by this requirement it is asked that it shall be defined how the checksum services shall be used by BSWMs (and not just asking for freely usable services for data protection); 2/ Applicable both for RAM and non-volatile memory: data protection with checksums can also be used for data in RAM (for long-term safety-related data that is not stored in non-volatile memory).
Contributes to:	--

3.45 Communication Stack Concept

3.45.1 [BRF00110] Protection of Communication

ID:	BRF00110
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Protection of communication
Importance:	High
Description:	AUTOSAR shall protect safety related data communication against corruption.
Rationale:	To detect when data exchanged between SW-Cs on the same or different ECUs is corrupted.
Use Case:	Two SW-Cs on two ECUs exchange safety-related data
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ This can be realized by a safety protocol which protects sender/receiver address (or message id), control information and data by checksum with time monitoring; 2/ All currently supported communication stacks (CAN, LIN and FlexRay) shall have a communication protection; 3/ All currently supported internal-ECU-communication stacks (like SPI) shall have a communication protection.
Contributes to:	--

3.45.2 [BRF00111] Data Sequence Control

ID:	BRF00111
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Data flow control
Importance:	High
Description:	AUTOSAR shall provide mechanisms for data sequence control.
Rationale:	Receivers must have the possibility to check whether a signal is received in sequence.
Use Case:	A distributed safety related powertrain control system receives a torque request signal via CAN with a sequence counter with a value higher than expected. This error is interpreted as several messages have been lost and there might be an inconsistent state within the powertrain system. This is handled with a reinitialization of the powertrain system.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ This can be achieved by adding sequence numbers (like PDU counter) to signals or frames. 2/ If the receiver detects a wrong sequence, it may decide for example to discard the message or reinitialize communication.
Contributes to:	--

3.45.3 [BRF00112] Routing Integrity

ID:	BRF00112
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Routing integrity
Importance:	High
Description:	AUTOSAR shall provide a mechanism that detects wrong routing and supports the correct routing of signals between SW-Cs.
Rationale:	To detect when a signal is coming from an unexpected SW-C sender, or when the received signal is providing the information that it is not supposed to provide.
Use Case:	If the receiver (e.g. COM BSWM) receives an unexpected message, it reports an error and discards it.
Dependencies:	--
Conflicts:	--
Supporting Material:	Note: 1/ To identify the routing, message id can be used, or alternatively sender and receiver ids.
Contributes to:	--

3.45.4 [BRF00113] Communication Watchdog

ID:	BRF00113
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Communication watchdog
Importance:	High
Description:	AUTOSAR shall provide a mechanism that detects if periodic signals are not exchanged within defined time interval (timeout).
Rationale:	This can be used for detecting errors in the communication system (loss of messages). Timeouts are commonly used to determine if a communication system is functioning or if an individual ECU is communicating. Failure to receive a message from a particular ECU means loss of information or functionality and may therefore lead to a change in SW-C or system operation.
Use Case:	The behavior of an anti-skid system might become erroneous should it base its operation on too old sensor values. The continuous updates of the sensor values can be monitored using a communication watchdog.
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ This can be achieved by having a timer at the receiver. If a message is coming too late, even if it is correct (correct sequence number, checksum etc), it shall be considered as an error (what happens afterwards is another issue: a resend request to sender or error reporting etc).
Contributes to:	--

3.45.5 [BRF00241] Multiple Communication Links

ID:	BRF00241

Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Multiple communication links
Importance:	High
Description:	AUTOSAR shall support multiple communication links.
Rationale:	To tolerate faults on one of the channels.
Use Case:	1/ If in a given system there is redundant communication HW (like two independent CAN buses, or one CAN and one FlexRay buses), then to provide fault tolerance, one can use a safety protocol on each channel (with data protected with checksum, address id, counter and timeout for example). Then, the receiver can do 1oo2 voting (i.e. take one of two correct received messages); 2/ If one channel completely fails the second channel may be used for reduced functionality communications.
Dependencies:	BRF00206
Conflicts:	--
Supporting Material:	Notes: 1/ This assumes that at configuration time, it is possible to statically configure which communication links are used.
Contributes to:	--

3.45.6 [BRF00242] Network Communication Monitoring

ID:	BRF00242
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Network communication monitoring
Importance:	High
Description:	AUTOSAR shall monitor network communication.
Rationale:	To detect high-level issues with network communication, to increase the fault detection capabilities of the system.
Use Case:	1/ If problems are detected with one network channel, the second channel can be used instead. 2/ If problems are detected with one network channel, the error reaction can be reconfiguration of SW-C functionality (e.g. cruise control shutdown or degradation).
Dependencies:	--
Conflicts:	--
Supporting Material:	Notes: 1/ Possible monitoring mechanisms: - Bandwidth monitoring: detect monopolization of communication medium by faulty (possibly non-safety-related) communication peer; - Monitoring with periodic sign-of-life signals and a timeout: ensure that the communication links is available.
Contributes to:	--

3.46 E-Gas Monitoring Applicability Concept

3.46.1 [BRF00301] Ability to make an AUTOSAR application compatible to the e-Gas monitoring Concept

ID:	BRF00301
Initiator:	AUTOSAR Safety Team
Date:	25 Jan 2008
Short Description:	Ability to make an AUTOSAR application compatible to the e-Gas monitoring concept
Importance:	High
Description:	It must be possible for an application to respect the safety concept known as e-GAS monitoring concept and to use the AUTOSAR standard. Note: The E-Gas Monitoring Concept is standardized by the AKEGAS working group and not part of the AUTOSAR standard. It is used as an exemplary item here because it is a standardized automotive safety concept. The feature requires that AUTOSAR standard must not make the use of the E-Gas Monitoring Concept impossible.
Rationale:	A complete analysis has been done; the result is a small set of requirements which cover the two main hypothesis considered by the e-Gas experts in the AUTOSAR safety team.
Use Case:	The e-Gas monitoring concept is a standardized automotive safety concept.
Conflicts:	--
Supporting Material:	Standardized e-Gas monitoring concept for engine management systems of gasoline and diesel engines, V 2.0, 29.04.2004
Contributes to:	--

3.46.2 [BRF00248] Testing and monitoring of I/O data and I/O HW

ID:	BRF00248
Initiator:	AUTOSAR Safety Team
Date:	27.02.2006
Short Description:	Testing and monitoring of I/O data and I/O HW
Importance:	High
Description:	AUTOSAR shall allow the use of mechanisms for the testing and monitoring of I/O HW elements as well as the safety-related values received/transmitted using the I/O HW elements.
Rationale:	To detect errors in measured sensor data or output actuator data, and to detect failures in I/O HW.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.46.3 [BRF00251] Priority Access to SPI BUS

ID:	BRF00251
Initiator:	AUTOSAR Safety Team
Date:	23.11.2007
Short Description:	Priority access to SPI Bus

Importance:	
Description:	Exclusive / Priority access to SPI bus should be granted to software modules that carry out timing-critical monitoring protocols between the main controller and a monitoring unit connected via SPI bus. This should be possible for both these software modules being included in an AUTOSAR software component, and these modules being included in a complex device driver.
Rationale:	We expect that there will be systems executing monitoring protocols (for example as described by the standardized E-Gas Monitoring Concept) as well as other communication via a single SPI bus. The other communication is expected to be driven by AUTOSAR components or BSW modules using the standard AUTOSAR interfaces. The monitoring protocol shall be executed as needed (with priority) otherwise an availability penalty would be imposed. Note: The E-Gas Monitoring Concept is standardized by the AKEGAS working group and not part of the AUTOSAR standard. It is used as an exemplary item here because it is a standardized automotive safety concept.
Use Case:	Carrying out a monitoring protocol in parallel with other communication on an SPI bus.
Conflicts:	--
Supporting Material:	Standardized e-Gas monitoring concept for engine management systems of gasoline and diesel engines, V 2.0, 29.04.2004
Contributes to:	--

3.46.4 [BRF00243] Communication protections against corruption and loss of data

ID:	BRF00243
Initiator:	AUTOSAR Safety Team
Date:	23.11.2007
Short Description:	Communication protections against corruption and loss of data
Importance:	High
Description:	If the responsibility of detection is placed in application, AUTOSAR BSW must provide a mechanism to transmit the communication protections against a corruption or a loss of data to the application (end to end protection protocol). If the responsibility of detection is placed in Complex Device Drivers, AUTOSAR BSW must provide a mechanism to transmit the communication protections against a corruption or a loss of data to the Complex Device Drivers.
Rationale:	If the Basic Software is responsible of the transmitted or the received secure data, AUTOSAR BSW must provide such mechanisms.
Use Case:	Applicable for bus system that carries Safety related data.
Conflicts:	--
Supporting Material:	--
Contributes to:	--

3.47 Configuration related features

3.47.1 [BRF00042] Use of XML instead of OIL for Configuration of OS

ID:	BRF00042
Initiator:	AUTOSAR WP Software Architecture and OS
Date:	03.05.2007
Short Description:	Use XML instead of OIL for configuration of OS
Importance:	**high** | medium | low
Description:	AUTOSAR OS uses currently the OIL format of the OSEK/VDX for configuration. It would be better to use only one format for the configuration.
Rationale:	Many people complained about the fact that the OS is using a different format, which also mean that tool have to handle the XML and the OIL language. There are existing also some bugzilla bugs which can only be solved if the OS uses XML.
Use Case:	Configuration of a AUTOSAR ECU
Dependencies:	--
Conflicts:	It would be beneficial if AUTOSAR can give the (OS) XML configuration back to the OSEK/VDX organization. It has to be decided if we drop the OIL or keep it as a second configuration option. I personally prefer to drop the OIL support.
Supporting Material:	--
Contributes to:	--