Document Title	AUTOSAR BSW & RTE Conformance Test Specification Part 1: Background
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	281
Document Classification	Auxiliary

Document Version	1.0.2
Document Status	Final
Part of Release	3.2
Revision	1

Document Change History

Date	Version	Changed by	Change Description
27.04.2011	1.0.2	AUTOSAR Administration	Legal disclaimer revised
23.06.2008	1.0.1	AUTOSAR Administration	Legal disclaimer revised
14.11.2007	1.0.0	AUTOSAR Administration	Initial Release

Disclaimer

This specification and the material contained in it, as released by AUTOSAR is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the specification.

The material contained in this specification is protected by copyright and other types of Intellectual Property Rights. The commercial exploitation of the material contained in this specification requires a license to such Intellectual Property Rights.

This specification may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only.
For any other purpose, no part of the specification may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The AUTOSAR specifications have been developed for automotive applications only. They have neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Advice for users

AUTOSAR Specification Documents 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 Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, 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 Introduction

1.1 Purpose, Scope and Target Audience

1.1.1 Purpose and Scope

This document is part of the process specifications for AUTOSAR BSW conformance testing. These process specifications contain process and methodology related descriptions that enable the specification of conformance tests for parts of the BSW in a uniform way. The focus of the present specifications lies on BSW modules (ICC3).

The process specifications for BSW conformance test specification follow a top-down decomposition approach (see Figure 1) and are made up of the following documents:

AUTOSAR BSW & RTE Conformance Test Specification Part 1: Background This document. It contains generic background information and explains testing principles that are necessary to understand the other three specification documents. It describes the meaning and context of BSW conformance tests within the AUTOSAR standard and, in particular, the scope of BSW conformance testing in general, e.g., what is tested, what not, which test methods are applied and what are the implications of AUTOSAR conformance tests for the test objects.

AUTOSAR BSW & RTE Conformance Test Specification Part 2: Process Overview [13] The Conformance Test Specifications (CTSpecs) contain a combination of executable and manual test cases that (if successfully passed) attest conformance of the BSW module under test to its specifications. The Process Overview document describes the overall creation and validation process for test cases that are suitable for attesting conformance. The starting point of the creation process is the Software Specification (SWS) document of a BSW module.

AUTOSAR BSW & RTE Conformance Test Specification Part 3: Creation & Validation [14] The Creation & Validation document deals with the realization of the CTSpec creation process. It describes in detail how the activities of the CTSpec creation process are carried out and defines the structure of the artifacts of the creation process.

AUTOSAR BSW & RTE Conformance Test Specification Part 4: Execution Constraints [15] The Execution Constraints document specifies the constraints on the execution environment for executing the CTSpecs against BSW module implementations or simulations of them (for validation purposes). A main part of this document is the functional specification of Test Adapters.

Figure 1 depicts principally the relationships between the four process specification documents, and puts this in the context of the CTSpec and the involved activities.

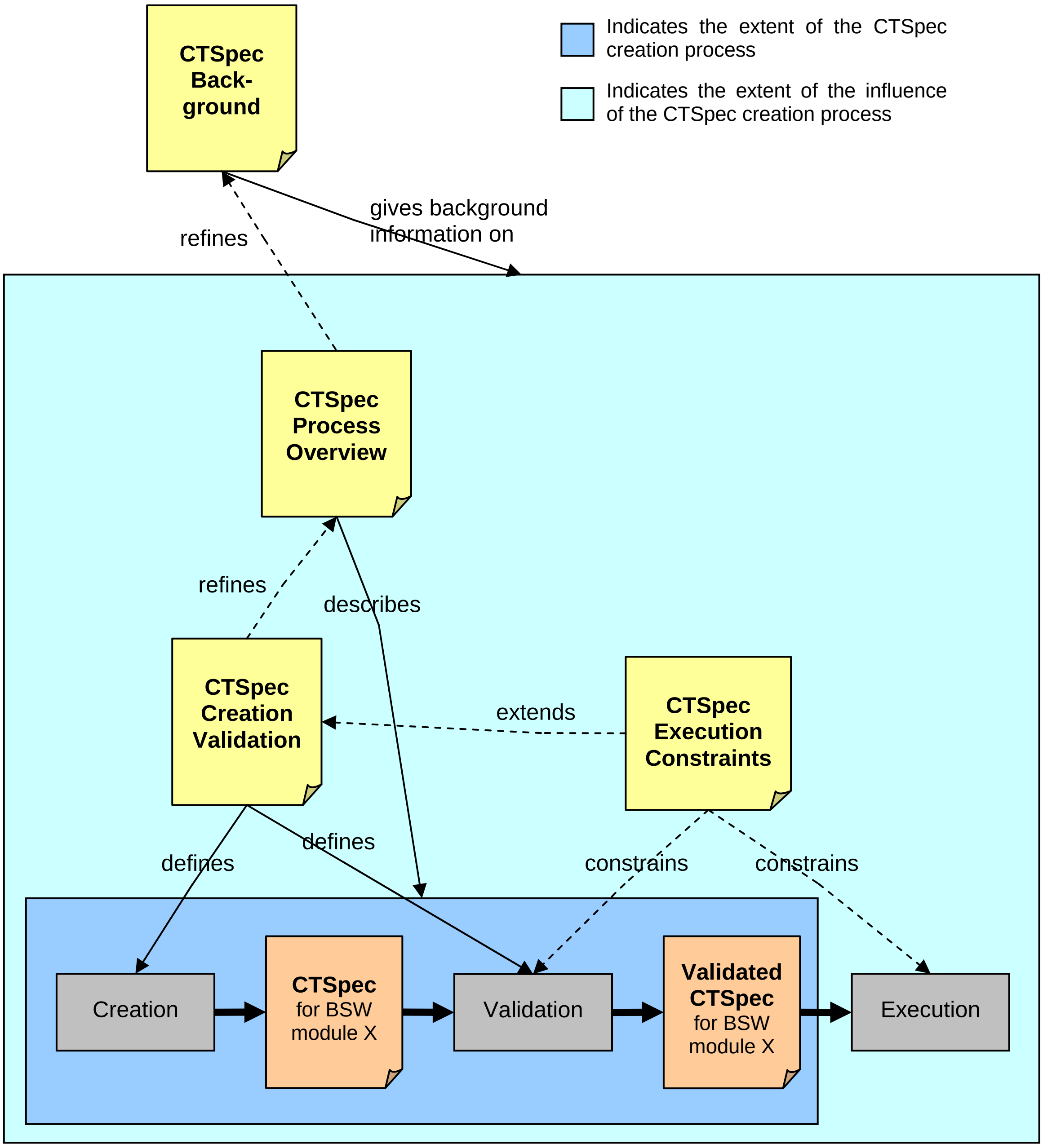
CTSpec Back-ground
Indicates the extent of the CTSpec creation process
Indicates the extent of the influence of the CTSpec creation process
refines
gives background information on
CTSpec Process Overview
refines
describes
CTSpec Creation Validation
extends
CTSpec Execution Constraints
defines
defines
constrains
constrains
Creation
CTSpec for BSW module X
Validation
Validated CTSpec for BSW module X
Execution

Figure 1 – Relationships between the Process Documents

1.1.2 Target Audience

This document set is intended to be used by any and all companies, groups and individuals engaged in the creation of Conformance Test Specifications for AUTOSAR BSW modules and clusters of modules, including the RTE. Different parts of the document set will be relevant at different stages of the creation of such specifications, but all parts need to be read, along with the documents listed in chapter 1.3.1, ‘Referenced Documents’, for a sufficient understanding of the needs of the conformance test process.

It is assumed that any and all companies, groups and individuals making use of this document set have a basic knowledge and understanding of “Best Practice” in the field of software unit test and conformance test. Where this document set differs from “Best Practice”, the document set shall prevail.
It is required by AUTOSAR that all parts of the document set are adhered to in the creation of Conformance Test Specifications, except where specific parts are marked as optional.

1.2 Definitions, Acronyms, Abbreviations

1.2.1 Definitions

The following definitions are given because they reflect the specific use of the terms within this document set.

Module As described by an SWS, the module is what is specified (as opposed to what is implemented). It is the superset of all possible variants.

Cluster Clusters are integrated units like modules. Compared with modules, clusters provide functionality that covers a larger scope and usually offer more highly integrated (less granular) services.

Variant[1] A single, integratable (or testable) instantiation of the implementation of a module or cluster. A variant is a concrete variation of one ore more features of a module that can vary according to the SWS.

Family A set of variants, as embodied by an implementation of an SWS. A family of a module or cluster has many common features (functional and non-functional properties) and varies in some other features. A family cannot be integrated or tested directly, but needs to be processed to generate a variant to permit integration or testing.

Dynamic Tests Tests that exercise an object under test by executing the code of that object. Dynamic tests generally involve calling a function of the object under test, with test data (passed parameters, global data…), and checking the result of that function call (returned values, affected global data…).

Static Tests Static tests, as a counterpart to dynamic tests, do not execute the object under test, but ensure that static parameters of the object (its configuration, its interface signature…) match its specification.

1.2.2 Acronyms and Abbreviations

Abbr.	**Description**
API	Application Program Interface (function signatures)
BSW	Basic Software
BSWMD	Basic Software Module Description
CB	Call Back
CC	Conformance Class
CD	Coder/Decoder (TTCN-3 – see Part 4)
CH	Component Handling (TTCN-3 – see Part 4)
CoDec	Coder/Decoder (TTCN-3 – see Part 4)
CTA	Conformance Test Agency

[1] Note that the term ‘Variant’ is also used, with a different meaning, within the SWS documents.

CTS	Conformance Test Suite
CTSpec	Conformance Test Specification
DEM	Diagnostic Event Manager
ECU	Electronic Control Unit
FCC	Functional Conformance Class
HLL	High Level Language, e.g. C, Java, C++
ICC	Implementation Cluster Conformance Class
ICS	Implementation Conformance Statement
IF	Interface
IP	Intellectual Property
PA	Platform Adapter (TTCN-3 – see Part 4)
PC	Personal Computer
PS	Product Supplier
RfC	Request for Change
RTE	Run Time Environment
SA	System Adapter (TTCN-3 – see Part 4)
SM	Service Management
SUT	System Under Test, e.g., an ICC3 module
SW-C	Software Component (an Application)
SWS	Software Specification
TCI	TTCN-3 Control Interface (TTCN-3 – see Part 4)
TE	TTCN-3 Executable (TTCN-3 – see Part 4)
TL	Test Logger (TTCN-3 – see Part 4)
TM	Test Manager (TTCN-3 – see Part 4)
TO	Test Object
TRI	TTCN-3 Runtime Interface (TTCN-3 – see Part 4)
TTCN-3	Testing and Test Control Notation, version 3 (see http://www.ttcn-3.org/)
UML	Unified Modeling Language (see http://www.uml.org/)
USB	Universal Serial Bus
XML	Extensible Markup Language (see http://www.w3.org/TR/xml11)

1.3 References

1.3.1 Referenced Documents

[1] AUTOSAR Glossary

AUTOSAR_Glossary.pdf

[2] Specification of Graphical Notation

AUTOSAR_GraphicalNotation.pdf

[3] Layered Software Architecture

AUTOSAR_LayeredSoftwareArchitecture

[4] Specification of ECU Configuration

AUTOSAR_ECU_Configuration.pdf

[5] Specification of ECU Configuration Parameters

AUTOSAR_EcucParamDef.xml

[6] List of Basic Software Modules

AUTOSAR_BasicSoftwareModules.pdf

[7] Methodology

AUTOSAR_Methodology.pdf

[8] Conformance Test Requirements

AUTOSAR_RS_ConformanceTest

[9] Requirements on BSW Module Description Template

AUTOSAR_RS_BSW_ModuleDescription

[10] Conformance Test Process Definition Path A-C

AUTOSAR_DS_CT Path A-C

[11] Specification of C Implementation Rules

AUTOSAR_SWS_C_ImplementationRules

[12] AUTOSAR BSW UML Model

AUTOSAR_BSW_UML_Model

[13] AUTOSAR BSW & RTE Conformance Test Specification Part 2: Process Overview,

AUTOSAR_CTSpec_Process_Overview

[14] AUTOSAR BSW & RTE Conformance Test Specification Part 3: Creation & Validation,

AUTOSAR_CTSpec_Creation_Validation

[15] AUTOSAR BSW & RTE Conformance Test Specification Part 4: Execution Constraints,

AUTOSAR_CTSpec_Execution_Constraint

[16] AUTOSAR Conformance Test Specification Template

AUTOSAR_CTSpec_Template

[17] ETSI. The Testing and Test Control Notation version 3; Part 1: TTCN 3 Core Language. ETSI, 2003
http://www.ttcn-3.org/

[18] IEEE Standard Glossary of Software Engineering Glossary. IEEE Std 610.12-1990

[19] IEEE Standard for Software Unit Testing. IEEE Std 1008-1987

[20] IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998

[21] IEEE Standard for Software Test Documentation. IEEE Std 829-1998

1.3.2 Traceability Matrix

Document: Conformance Test Requirements [8]

Ref	Requirement	Satisfied by
CTR0010	Testing Scope should cover BSW, RTE, API configuration and software components behavior	Part 3 [14]
CTR0020	Conformance Test coverage shall be comprehensive	Section 3.3 & 3.6 Part 3 [14]
CTR0030	AUTOSAR shall provide the test specification	Section 1.1.1 Part 2 [13]
CTR0060	Comprehensive identification of required test set up	Section 3.2 Part 2 [13] Part 4 [15]
CTR0070	Unambiguous Identification of test cases	Part 3 [14]
CTR0080	Test specifications shall be formal	Section 3.2
CTR0090	Tests specifications must be available for all defined Conformance Classes configurations	Section 3.2
CTR0100	The test setup shall be documented	Sections 3.2 & 5
CTR0101	Validity of Attestation	Section 3.1 & 3.9
CTR0102	Extension of Attestation	Sections 2.2.5, 3.7 & 3.9
CTR0110	Validity of conformance approval is permanent	Out of scope for Conformance Test Specification
CTR0130	Formal conformance procedure	Out of scope for Conformance Test Specification
CTR0170	Update Test implementations (CTS) in synchronization to standard releases	Out of scope for Conformance Test Specification
CTR0180	Worldwide representation of CTAs in major automotive areas	Out of scope for Conformance Test Specification
CTR0191	A CTA shall be impartial with respect to testing and attestation services	Out of scope for Conformance Test Specification
CTR0200	AUTOSAR maintains an internal list of all accredited CTAs	Out of scope for Conformance Test Specification
CTR0230	Only a CTA can issue a valid conformance attestation	Out of scope for Conformance Test Specification

Ref	Requirement	Satisfied by
CTR0240	Time frame for attestation (obligation of CTA)	Out of scope for Conformance Test Specification
CTR0260	Worldwide acceptance of test results	Out of scope for Conformance Test Specification
CTR0270	A product that is conform to the AUTOSAR specifications can carry the AUTOSAR trademark	Out of scope for Conformance Test Specification
CTR0280	CTAs will be assessed regularly	Out of scope for Conformance Test Specification
CTR0331	No limitation to the number of CTAs	Out of scope for Conformance Test Specification
CTR0351	A third party CTA is expected to offer a conformance test suite (CTS) and attestation services	Out of scope for Conformance Test Specification
CTR0360	The test specifications should be organized in a hierarchy	Not covered. The rationale does not apply to the methods used.
CTR0370	Atomicity of test cases	Part 3 [14]
CTR0380	Modification of test cases is not allowed in CTS	Section 3.2
CTR0390	AUTOSAR license for a CTA and sign the CTA agreement	Out of scope for Conformance Test Specification
CTR0410	Test specifications shall enable automatic generation of test implementations	Not covered. This requirement is the subject of an RfC.
CTR0420	For self conformance declaration a product supplier shall become first party CTA.	Out of scope for Conformance Test Specification
CTR0430	Conformance testing and attestation processes shall involve an accredited party	Out of scope for Conformance Test Specification
CTR0440	CTA surveillance shall include CTS continuous improvement check	Out of scope for Conformance Test Specification
CTR0450	Quarterly report by a CTA	Out of scope for Conformance Test Specification

1.4 Overview

This document provides the background to the process of specifying conformance tests. The various chapters achieve this by first setting the context within which the Conformance Test Specifications are created (chapter 2), then by defining the scope of the conformance tests that will be specified (chapter 3). Chapter 4 gives a broad overview of the steps involved in creation and validation of a conformance test specification, and chapter 5 finally describes the mechanisms that are to be used to specify and (to a certain extent) execute the conformance tests.

This document does not attempt to define the execution of conformance tests – that is beyond the scope of the specification of the conformance tests. Nevertheless, some aspects of the test specification process restrict or constrain the range of options available for conformance test execution. This theme is covered in more detail in Part 4 of this document set, but the background to these restrictions is embodied in this document.

2 Domain Context

AUTOSAR is a large project, covering software architecture & processes for use in automotive control systems in general. This chapter gives the user some guidance as to where to find more information on the topics relevant to conformance test of the basic software and run time environment, and goes into a little more detail on the architecture defined by AUTOSAR for basic software.

2.1 AUTOSAR Documents

AUTOSAR documents can be generally divided into three types for testing purposes:

1. Those that are direct inputs to the creation of Conformance Test Specifications for any given AUTOSAR basic software artifact.
2. Those that are direct inputs to the definition of the process of creating such Conformance Test Specifications.
3. Those that are at best indirectly relevant to Conformance Test Specification.

The first type will be referred to most often during the creation of the Conformance Test Specifications. These include (but may not be limited to) the Software Specification (SWS) for the artifact to be tested, the SWS for each of the modules providing services to and requiring services from the artifact to be tested, the configuration parameters for the artifact under test (these are listed for all basic software in [5]), the Basic Software Module Description (BSWMD) for the artifact under test, and the AUTOSAR BSW UML model ([11]).

The second type include documents concerning the architecture ([3], [4], [5]), process (0, [8], [10]) and methodology ([7]); and those of a more general nature, for instance the background to the BSW requirements ([9]) and the Glossary ([1]).

The third type includes all those documents not referenced by the AUTOSAR BSW & RTE Conformance Test Specification process document set. Some of these are clearly not relevant to the specification of conformance tests (for instance the Simulink Style Guidelines), others have a direct bearing on the subject of conformance testing (for instance the Specification of the Virtual Functional Bus provides important information regarding the RTE). The set of indirectly relevant documents is both too large and too subject to change (being directly related to the test subjects) to be listed in this document set.

2.2 BSW Modules, Cluster and Layers

2.2.1 Architectural Overview

As described in some detail in [3], and defined with rigor in [11], the AUTOSAR architecture divides neatly into two major sections, separated by the Run Time Environment (RTE). Above the RTE are application software components, which are of little relevance to the BSW Conformance Tests as defined herein. Below the RTE is the basic software (BSW), the component parts of which (and combinations thereof) are the subject of this process document set.

A commonly used, simplified version of the BSW architecture is given in Figure 2. This figure shows how the BSW architecture is broadly divided into three layers (not including the RTE), although it also shows how some modules extend across more than one layer. It further indicates how the architecture is divided vertically into functional groups, which may be clustered to give improved performance or efficiency. An example cluster (the communications stack) is outlined in red in the figure.

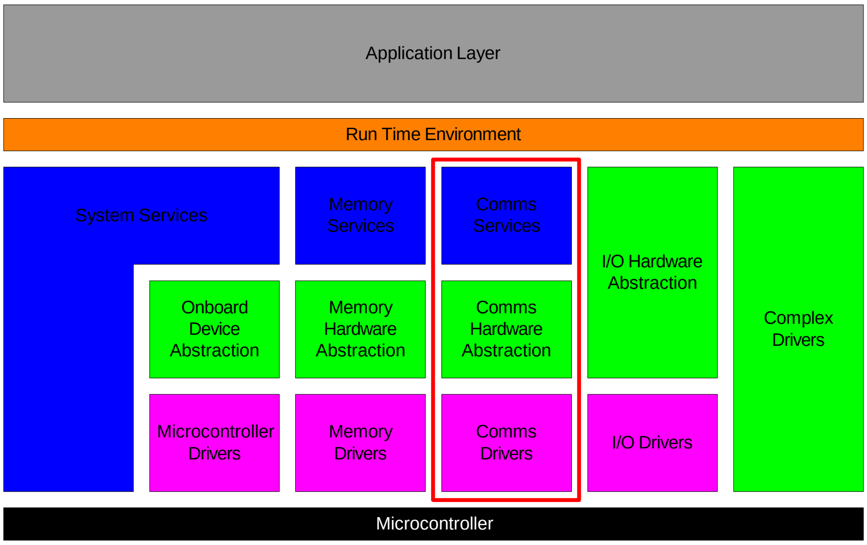

Figure 2 – Simplified AUTOSAR Architecture

As outlined above, Figure 2 is a simplified diagram. More information can be found in [3] & [11], although it can be stated here that there are approximately 60 BSW modules (providing test objects at ICC3 level), and around a dozen clusters (providing test objects at ICC2 level) have been proposed.

2.2.2 Specifications of BSW Modules

Within AUTOSAR, a Software Specification (SWS) defines the complete scope of functionality, interface and configurability of a Basic Software Module. The SWS document describes this textually, with atomic requirements on the functionality, configurability and interface description being given in uniquely numbered SWS items. Some of these items are represented in a more formal (although less descriptive) manner in the Basic Software Module Description (BSWMD), which takes the form of an XML document.

2.2.3 Optional Parts in a BSW Module

Not all of the functionality, interface and configurability specified in an SWS is mandatory. Optional parts[2] are indicated textually within the specification, and formally within the BSWMD. To be more precise, functionality within a BSW module is specified within the SWS as being present or not. Similarly interfaces (which may be tightly coupled with functionality) are also either present or not. Configurability, on the other hand, is specified as a permitted range of values within the SWS. Configurable parameters may be of any type, and in fact are also the mechanism by

[2] Here, and throughout this chapter, 'parts' is taken to mean any combination of functionality, interface and configurability.

which functionality and interfaces are declared optional (using a parameter with a fundamentally Boolean type).

Any provided implementation of a module (or cluster) may contain anywhere between none and all of the optional parts, and may cater for any permitted sub-range (from a single value to the full range) of a configurable parameter. Any concrete implementation of a module or cluster – i.e. one that can be compiled to an object file and integrated into the software for an ECU – may also contain between none and all of the optional parts, but must have its configuration limited to a single value for each configurable parameter.

2.2.4 Modules, Clusters, Families and Variants

To avoid having to repeat the description given above whenever it is unclear whether the subject of discussion is what's described in the SWS, what's provided by an implementer, or what's integrated into an ECU, the terms module, cluster, variant and family have been defined above (see section 1.2.1).

2.2.5 Implementation Conformance Statements

The idea of an Implementation Conformance Statement (ICS) is useful here. This is not a concrete object within the AUTOSAR process or methodology, rather a view on the BSWMD at specific points in that process.

At the point of specification of a BSW module, there is no ICS. There is only the BSWMD, containing the list of configurable parameters and their ranges. When a Product Supplier (PS) creates an implementation of the BSW module (a family of variants), the BSWMD for that implementation then specifies the implemented ranges of those parameters. At this point, there is a view on the BSWMD that can be considered the ICS, and this is a statement by the PS of what is implemented, and consequently what can be tested for conformance.

When conformance tests are to be run on this implementation, first a concrete instantiation has to be created (a single variant), in which each configurable parameter is provided with a single value. This variant can then be subjected to the tests defined in the conformance test specification, such as compilation and linking with the test harness for automated testing, or passing to reviewers or inspectors for manual testing. The ICS at this point represents this variant.

Since it is only specific variants that can be tested, and there may be a considerable number of unique variants possible within an implemented family, the tested variants need to be chosen such that reasonable confidence exists that all possible variants will be conformant.

Similarly, when the module is to be integrated into an ECU system, each configurable parameter must be provided with a single value so that a single variant can be created. At this point, the view on the BSWMD equivalent to the ICS contains the same values as the ECU Extract of System Configuration (see [7]).

It is important to note that there are aspects of the BSWMD (and hence the ICS) that have a bearing on the eventual attestation of conformance (see also section 3.8), but which may not require active 'configuration' as part of the creation of a variant from the family represented by the implementation of the SWS. Among these are references to hardware and registers etc. for device drivers.

3 Conformance Testing

This chapter characterizes AUTOSAR conformance tests and discusses the potential and limits of AUTOSAR conformance testing in relation to test goals, test references and test objects. It further indicates what this conformance test scope means for a test object's statement of conformance, i.e., for AUTOSAR modules or clusters.

Every conformance test requires an object to be examined for conformity and something to which this object shall conform. However there is no single accepted definition of exactly what tests a conformance test comprises. Thus this chapter starts with a short discussion of what conformance means in general in section 3.1. Section 3.2 outlines more specifically the purpose and nature of AUTOSAR conformance tests. Section 3.3 details the objects that shall conform and section 3.4 adds details on what these objects shall conform to (test references). Section 3.5 elaborates the objectives (goals) of the AUTOSAR conformance tests. Next, in section 3.6, the discussion moves to the test methods that AUTOSAR conformance tests apply to the test objects based on the test references to accomplish the test purpose and test goals. Section 3.7 extends the discussion of conformance to cover selection of variants from a family in order to deduce conformance of all variants of the family. Sections 3.8 & 3.9 close the chapter with a discussion of what kinds of results might be expected of the conformance tests and what can be read into an attestation of conformance based on these results.

Part 2 of the document set [13] develops this chapter further, by describing the process resulting from the need to specify conformance tests within the scope presented here. Neither this chapter nor that document deal with the conformance attestation process, including the implementation of a conformance test suite, the execution of the conformance tests and the verification of the test results. The various paths of the conformance attestation process together with a definition of the tasks of the participating parties are defined in [10].

3.1 Standards and Conformance Statements

A conformance statement in general says something about the extent to which an object (or a process) compares to a standard, which may be an international standard, some guidelines or a specification. Conformity to a standard implies a certain, fundamental quality of an object regarding the standard. The scope (extent) of the standard does not affect what must be done to specify and run conformance tests, but it does define the scope (extent) of the consequent conformance statement, which matches that of the standard. For instance, conformance statements regarding vehicle emissions are restricted to a geographical area corresponding exactly to the scope of the standards against which the tests are written. US homologation does not cover a vehicle to be sold in Europe, and vice versa – despite there being no difference in the vehicle. For AUTOSAR, the scope of the conformance statement is the membership of AUTOSAR.

A conformance statement can only be provided for an object when that object is deemed to have passed such tests as have been defined to show its conformity with the relevant standard. Thereby, conformity tests are usually distinguished from more comprehensive tests of functional properties, also called functional tests, and tests of non-functional properties, also called quality tests. The extent of these distinctions depends on the purpose and goals of each of these tests. Each test of a functional or a non-functional property of an object against a specification document or against a standard regulating more fundamental principles is a conformance test. However, conformance tests against a general standard cannot by their very nature replace evaluations that test for bugs aggressively in every critical part of specific object's implementation. That is, conformance tests are not all-embracing functional tests or quality tests.

3.2 Purpose and Nature of Conformance Tests

AUTOSAR conformance tests check whether a variant of a BSW module or cluster (the test object) implemented by one or another product supplier (PS), complies with relevant AUTOSAR specifications (SWS documents). AUTOSAR conformance tests verify whether the object under test satisfies selected conditions imposed in the relevant specification documents (SWS).

AUTOSAR Conformance Test Specifications are a part of the formally released AUTOSAR document set, and have a strict relationship to the SWS documents the conformance to which they are designed to attest. As such, the CTSpecs shall not be altered in any way outside the normal AUTOSAR change mechanisms, and any Conformance Test Suite generated from a CTSpec shall be directly and uniquely traceable to that CTSpec.

Primarily, AUTOSAR conformance tests are functional tests. They intend to show the correct functionality of test objects in terms of their public input/output behavior in typical, valid applications. To ensure the reproducibility of tests, and the comparability of test results, the tests check only deterministic, implementation-independent behavior with deterministic test methods. Because tests and the test results depend on many factors not directly contained in test specifications, AUTOSAR conformance tests require the traceability of

- The test environment (test system),
- The tools and tool parameters used for compilation of executable binaries (test objects, test systems),
- The variants of the test object actually tested, and
- The target platform (hardware) and its configuration

In general AUTOSAR conformance tests

- Are unambiguous due to a formally defined test specification language
- Abstract from the implementation specifics of the test object
- Can be adapted to test objects that are deployed either on a development machine (in case of hardware-independent test objects) or on a target platform
- Can automatically be translated into executable test drivers and test stubs
- Enable iterative and incremental test executions throughout and at the end of the development process
- Define test inputs and expected test outputs in such a way that conformance tests are reproducible and provide consistent test results

- Follow a Testing-by-Contract philosophy in that tests are only specified for situations that match the pre-conditions; if there is no claim for an operation to work under a specific condition, there is no test for that condition[3]

AUTOSAR conformance tests can be applied to test objects on several levels of integration of BSW modules. As the size of the test objects increases from modules (ICC3) to subsystems (ICC2) to complete system including the RTE (ICC1) the inputs, outputs and preconditions of the conformance tests get more complex and the aspects to be tested change, but the basic intention remains the same.

Currently, AUTOSAR conformance tests focus on ICC3 modules. Conformance tests do not consider specifications of nonfunctional properties at the module level, i.e., specifications that restrict the resource usage of test objects, like limits on response and execution times or memory usage. This is based on the assumption and best practice that the implementation of such specification items should be engineered and tested on higher abstraction levels[4] allowing for system-wide optimizations or at lower abstraction levels hiding hardware and event-driven constraints.

3.3 Test Objects

Any attestation of conformance must have at its core something that conforms. In the case of an AUTOSAR BSW module or cluster this is a two-part entity:

- The implementation of a software specification. This will be, in general, a representation of a family of variants, which is in turn a subset of the family of variants specified by the SWS.
- The formal description of the implemented family (which can be considered the Implementation Conformance Statement), provided by the associated BSWMD[5], and its specific set of values.

As alluded to above (see chapter 2.2.5), in order to perform any dynamic test on the family of variants representing a module, a specific variant must be generated. This generated variant then becomes the test object for a specific set of applicable test cases. The family of variants from which each variant for test is chosen is described formally by the BSWMD for the family, and each variant is generated from the implementation[6]. The object of a dynamic test is always a valid and executable variant.

AUTOSAR conformance tests shall take into account all possible production variants of the test objects, but shall only actually test a selected subset of these production variants. The tests do not consider development variants.

Some of the BSW modules that are addressed by AUTOSAR conformance tests implement AUTOSAR Services ([1], 3.17). However, AUTOSAR conformance tests handle these AUTOSAR Services strictly as the functionality behind an API, subject

[3] The Testing-by-Contract philosophy is based on the Design-by-Contract philosophy. A different test approach is Defensive Testing which tests under normal and abnormal pre-conditions. This is applicable to defensive implementations in which an object under test would accept any input.

[4] Future releases of AUTOSAR conformance test specifications of modules at ICC3 level or conformance tests at higher aggregated levels, i.e. at ICC2 or ICC1 level, might consider selected non-functional properties.

[5] Here it is assumed that every test object has an associated BSWMD (see [9]) regardless of the aggregation level (ICC3, ICC2 or ICC1).

[6] The mechanism of generation is beyond the scope of the subject of conformance test specification. However, Part 3 of the document set contains information on the generation and selection of module variants to be tested.

to configuration, within the BSW module (the combination of function, interface and configuration is outlined beginning in 2.2.2 above), and not as AUTOSAR Software-Components ([1], 3.18). Thus the conformance tests interface with AUTOSAR Services exclusively at the Standardized Interface ([1], 3.182) of BSW modules that implement services, and they do so according to the test objectives and by means of the test specification methods explained in the following sections.

3.4 Test References

In addition to an attestation of conformance having at its core the thing that conforms, the attestation must be clearly based on something to which that conforming object conforms, that is the test reference. For an AUTOSAR BSW module or cluster this test reference comprises two parts:

- The Software Specification (SWS) that defines, through a set of uniquely identified SWS items, the totality of functionality, interfaces and configurability that are permitted in this module.
- The corresponding Basic Software Module Description (BSWMD) that formalizes the permitted configurability (as well as other aspects of the specification).

Specification documents (SWS documents) are the primary test references for the specification of conformance tests. A specification (SWS) defines both functional properties and non-functional properties of families of objects to be implemented and tested. The BSWMD which includes the ICS is used for adapting a conformance test suite during test execution.

Specifications of functional properties require the existence of operations with a defined behavior that transform some input data into some output data. Specifications of non-functional properties (e.g. efficiency, robustness, reentrancy and safety) constrain functional properties and thus the implementation of operations. Configurability is a special non-functional property defining valid variants of the module or cluster under test. All specification items are uniquely identified across all BSW specification documents for reasons of requirements traceability. This unique identification assists in the determination of implementation coverage and test coverage.

Many, but not all, specification items are relevant, because AUTOSAR conformance tests are fundamentally black-box functional tests that check the behavior of the test object on the basis of typical and correct applications. Relevant SWS-items (test items according to [21]) thus concern:

- Operation signatures and interface files[7]
- Behavior of single operations
- Behavior of operations affected by conditions from previous operations (call sequences covered in object states)
- Structure and contents of data values flowing across the object's interfaces
- Configurations that affect the behavior or resource usage

Additionally, various other AUTOSAR documents are referenced during testing, including such as the layered software architecture [3] and the specification of ECU configuration parameters [5].

[7] Interface files are called header files in the programming language C.

BSW modules (or clusters) that provide functionality to the application layer do so through the provision of Services. These services are described in a chapter of the appropriate SWS documents, but are not identified as unique SWS-items within those documents. For this reason, and for the reasons given in the section on Test Objects (3.3), BSW Services are not the direct subject of ICC3 conformance tests[8].

3.5 Test Objectives

AUTOSAR conformance tests shall basically show that the test object conforms to the relevant specification documents. Consequently, AUTOSAR conformance tests basically assume that the test object is correct[9].

Tests of objects implemented by different product suppliers for conformance to a public standard like AUTOSAR can only check the behavior that is externally controllable and observable. Being functional tests, AUTOSAR conformance tests are not concerned with how internal processing occurs, but rather, with the results of the processing for given inputs. In particular AUTOSAR conformance tests intend to show that

- The test object provides the assets (configurable parts and operations) as specified in the test reference and declared in the ICS
- The provided operations of the test object behave as specified
- The test object uses the operations of collaborating objects as specified
- The test object provides correct data values for correct data inputs
- Public operations of test objects work correctly when used correctly

AUTOSAR conformance tests must be constructed in a modular way and be automatically executable because the tests shall address several variants of the object under test from various product suppliers. Additionally, AUTOSAR conformance tests shall be capable of testing hardware-independent objects in different hardware-independent environments.

3.6 Test Specification Methods

AUTOSAR conformance tests are engineered. They apply a diversified, balanced set of black-box test methods that suffice to fulfill the intention of the conformance tests, reach the test objectives and are suited to checking the test items (SWS-items relevant to conformance tests) with relatively low effort. AUTOSAR conformance tests lack any knowledge of internal execution paths, structure or implementation of the object under test.

In general there are several test methods supporting the black-box test strategy, such as the following.

Equivalence Class Testing	Select test data out of classes of test data that are supposed to have equal effects
Boundary-value Testing	Select test data on or nearby the boundaries of equivalence classes

[8] BSW Services may be the subject of conformance tests at higher integration classes (ICC2 & ICC1).

[9] Tests with the intention to show that object under test works is not an effective way to find deeply hidden defects. But such tests fit their purpose which is to show that the object does conform to a public standard whose specifications allow some implementation flexibility.

Syntax Checking Select a spectrum of differently structured test data for operations that must parse its input or provide structured output data

Combinatorial Testing Model the behavior of an operation, e.g. as a decision table, classification tree, or cause-effect graphs, classify and combine parameters, test data and conditions and choose a minimal number of test cases

Pair-wise Testing A special combinatorial testing method that selects test values for all pairs of variables (parameters) instead of for all possible variable combinations

Random-value Testing Subject the test object to test data and call sequences that statistically model the operating environment

State-Transition Testing Apply operations provided by the test object depending on the operation call history (conditions resulting from previous applications of the test object)

Use Case Testing Choose sequences of operations provided by the test object (application scenarios) that different clients of the test object might call to accomplish something worthwhile within different contexts

All test methods mentioned above can potentially be used for specifying AUTOSAR conformance tests. However, some of these methods are not appropriate for, and are therefore excluded from, AUTOSAR conformance tests, for the following reasons.

Boundary-value Testing Not used because the conformance tests concentrate on correct behavior, and thus do not focus on testing around the boundaries of equivalence classes.

Random-value Testing Not used because conformance tests shall test behavior deterministically and provide unambiguously reproducible results. Statistical testing can only find errors with a certain probability.

Use Case Testing Not used because such tests, like statistical testing, are not defined from the perspective of the object under test but from the perspective of different clients and different application contexts, and are therefore potentially limited to a specific scenario.

Selecting the appropriate test methods from the list of possibilities is a complex task that requires the assessment of many issues, including the goals of conformance tests, the nature of the test items, the overall purpose of the test object and the nature of the test system (test environment). No single test method is sufficient. The complementary benefits of the various test methods have to be balanced.

As few as necessary test cases shall be designed to reach the test goals and cover all test items at least once. Or in other words, a set of test techniques, test cases and test data shall be designed, where this set is sufficient to test every conformance-testable, atomic SWS-item relevant for a production variant of the object under test, such that basic confidence in the operational reliability of that object under test is gained.

Factors that need to be taken into consideration when designing the set of test techniques, cases and data include:

- How the test strategy and test coverage are affected by the characteristics and typical risks (failure modes, known or suspected weak points etc) of the test object.

- The quality of the specifications, including their degree of formalization, the depth and width of the specification items, atomicity, clarity etc.

3.7 Variant Testing

AUTOSAR specifications (SWS) define the commonalities and the variability of a functionally coherent part of the BSW, either a module or a cluster. Dynamic AUTOSAR conformance tests are always applied to valid and executable variants. The specification documents define valid variants via configuration parameters leading to a combinatorial complexity in the number of possible test objects.

The limitations imposed by combinatorial complexity in general and especially for product families[10] are most prominent in the area of testing and quality assurance. Without constraining the combinations, testing and validating of all possible property combinations is computationally intractable except for trivial cases. Thus, in the case where a family is tested for conformance, only some variants can be tested. These variants are selected basically with the pair-wise testing method (see section 3.6). Constraints between variation points (configuration parameters defined in the SWS) will restrict the number of variants out of which one is selected for dynamic testing. The number and kinds of variants to be tested can be determined more exactly in the case where domain specific requirements and typical applications and scenarios are known. According to the intention of AUTOSAR conformance tests to show the correct behavior of test objects, only valid variants are configured and subsequently tested. The configuration of variants of objects to be tested and the verification of whether these variants are valid can be seen as a static test, i.e., as a test that does not execute the test object. The variants to be tested as well as the applicable test cases are selected, according to the described procedure, during test execution.

An additional consideration when selecting variants for test is that AUTOSAR specifies several (three, at the time of writing) classes of configuration parameter. These classes are related to the stage of the build process at which the configuration is applied. Any given configuration parameter may be a member of one or more of these classes – that is, it may be applied at one or more of the specified points in the build process. The correct application of configurations must be tested in terms of both the end result of the configuration (a correctly configured variant), and that the configuration is applied at the correct stage of the process, where a particular class is specified.

3.8 A Summary of Conformance Tests

Tests can be specified from a functional point of view – at the boundary between implementer and clients – or from a structural point of view – taking an implementer's position. Functional tests target different kinds of bugs than structural tests. Both tests are useful, both have limitations. In a systematic test process both forms of unit tests precede integration tests but cannot replace them.

AUTOSAR conformance tests check the behavior of an individual object according to its specification. AUTOSAR conformance tests apply an object correctly according to what is defined in the specification, and check whether the object correctly uses collaborating objects (simulated by test stubs).

[10] Also known as *product lines*.

3.8.1 Bugs Detected during Conformance Testing

AUTOSAR conformance tests are deterministic functional black-box unit tests derived from the specification (SWS) of the object under test. These tests exploit the deterministic input-output behavior of the implementation of an object to test whether the configuration, interface and input-output behavior are as specified in the SWS. Thus, AUTOSAR conformance tests test for the following bugs at the test object's interface in the detail and scope required by the SWS:

- Invalid configurations and interfaces[11] of an object
- Incorrect configurations resulting from correct configuration parameters
- Missing functionality
- Incorrect output data for provided input data (covering incorrect operation, incorrect data format, unexpected internal state)
- Incorrect use of the specified operations of collaborating objects

3.8.2 Bugs Not Detected during Conformance Testing

Being black-box unit tests, AUTOSAR conformance tests cannot assure that every possible path or condition of the test object's implementation is executed. This is the objective of tests that apply a white-box view on the test object.

AUTOSAR conformance tests also do not check for errors resulting from wrong interactions between test objects. Verifying that several individually tested objects cooperate correctly is the objective of integration tests. Note that integration tests should not be confused with testing of integrated objects, which is just higher level black-box testing.

AUTOSAR conformance tests do not check timing[12], performance or other 'quality of service' requirements, including (but not limited to) interruptability, reentrancy, and execution in contexts dependent on hardware settings (such as interrupt contexts, or exclusive or protected memory regions).

In particular, AUTOSAR conformance tests do not cover the following, even if they are specified in the SWS:

- Internal control flow bugs (i.e. evaluations of variables at decision points)
- Internal data flow bugs (i.e. sequences of read/write operations with module-internal variables)
- Precision faults resulting from, e.g., improper use of floating-point data, unexpected truncations
- Integration bugs like wrong operation sequences and data flows among several objects and objects handling shared (exchanged) data items inconsistently

[11] Interfaces regarding the source code are described by C header files. Interfaces regarding the binary code are the function entry points in the code segment and memory cells in the data segment depending on the realization of configuration classes [11]: pre-compile time, link time, post-build time (loadable or selectable).

[12] Time-outs (mechanisms supporting time-triggered behavior in the object under test) shall not be confused with constraints on execution or response time. For time-outs, the behavior shall be tested for conformance; the timing constraint shall not.

- Misbehavior regarding insufficient performance and instability under high load resulting from high call frequencies and huge amounts of data
- Misbehavior of non-reentrant objects or operations when called in several non-deterministically interleaving operation sequences
- Misbehavior due to invalid timing (hard or soft real-time) or due to wrong sequence assumptions related to context-dependent external signals
- Extra features that exceed AUTOSAR specifications and possibly interplay with properties specified by AUTOSAR

3.8.3 Robustness

In the current version, AUTOSAR conformance tests do not check the robustness of the test objects as indicated by the Testing-by-Contract philosophy. Robustness tests[13] analyze the test object's response to any kind of operating errors including wrong applications that violate pre-conditions of operations under test. Such approaches are useful for operations that apply a defensive implementation style. AUTOSAR conformance tests address robustness issues only as far as there are explicit specification items regarding the robustness of production variants of the object under test.

From an architecture point of view, robustness is more of an issue at external interfaces (interfaces of an ICC1 cluster) than it is an issue of internal interfaces between parts of the system (ICC3 modules, ICC2 clusters). When considering ICC3 interfaces, BSW conformance tests are available for both sides of the interface (provided and required), so robustness is less of an issue. When considering ICC1 interfaces, BSW conformance tests are only available for one side of the interface, so robustness of the interface is more important. Thus, usually, robustness tests are addressed at the system level (ICC1) rather than at subsystem (ICC2) or module (ICC3) level[14]. Nevertheless, if there is an SWS-item stating that an operation is to be implemented so that it accepts any inputs then the conformance tests will include selected defensive tests, i.e., tests under abnormal conditions as well as normal conditions. But again, these tests too intend to show that the test object works as specified under normal and abnormal conditions and thus omit diabolically constructed test cases that evaluate the test object in “all“ its nooks, boundaries, and near the edges.

3.8.4 Conformance Tests for Configuration

The basic strategy for conformance tests to show the correct realization of test objects holds also for testing configurable features. AUTOSAR conformance tests check for the omission of required features (configured parts and operations) of a test object simply by executing the necessary conformance tests. If the test object lacks

[13] Tests for robustness or reliability must not be confused with tests for safety. A system is safe if it is free from accident or loss expressing the high criticality of safety bugs. Safety issues require special provisions regarding the design and verification of the test objects and thus change the test approach.

[14] An open architecture for embedded systems consisting of exchangeable parts (ICC3 modules, ICC2 clusters) must of course find the right balance between robust and efficient implementations

some feature the test will fail. However, conformance tests do not test for accidental inclusions or undesired features[15].

3.8.5 Other Considerations

Other restrictions regarding the scope of the conformance tests are related to the content of the test reference, i.e., the specification documents. Tests cannot cover specification items that leave the behavior of the object under test explicitly up to the implementation or to the way objects are used without specifying how the objects are user. Other points that restrict the test coverage come about as a result of specification items that are too vague, or are hardware-dependent – as is especially the case for hardware drivers. Some specification items may allow objects to be coupled on the basis of knowledge of implementation details, which raises the degree of coupling between objects and thus compromises exchangeability and testability of the objects[16]. Despite every effort being made within AUTOSAR to create ‘ideal’ SWS documents, it is inevitable that such points will remain, and the process of conformance test specification must be created such that it can deal with these issues in a controlled manner.

Successful AUTOSAR conformance tests indicating that a test object conforms to the relevant AUTOSAR specifications are a necessary but not sufficient prerequisite for tests that check the interoperability of the object with various other objects tested for conformance (modules and clusters). Thus AUTOSAR conformance tests cannot guarantee that test objects are free of defects – indeed there is no test that can guarantee such a thing. Moreover, conformance tests of individual objects cannot, by their nature, exclude inconsistencies and incompatibilities between objects that communicate with each other either directly or indirectly.

3.9 Implications of Conformance Statements

A test object which is successfully tested to be conformant to the AUTOSAR specifications gains considerable confidence of implementing all the essential functional properties of the relevant specification (SWS), as restricted by the ICS. The test object is integratable in an AUTOSAR BSW system from a purely functional point of view with the assumption that collaborating objects interpret and follow agreed interface contracts in the same way as the object under test does.

However, the vendor of a BSW module/cluster shall not rely exclusively on AUTOSAR conformance tests to verify the correctness of the implementation.

[15] Undesired features (permitted by an SWS) may incorrectly affect the behavior of other parts of a test object or force the test object to use more operating resources than permitted by an application.

[16] Characteristics of good specification documents can be found in [20].

4 Process of Conformance Test Specification

This chapter gives an overview on the design of tests that satisfy the purpose and goals of AUTOSAR conformance tests as stated in chapter 3.

The goal of the test specification process is the creation of a conformance test specification (see CTSpec Template [16]) which defines the test cases to be executed for conformance attestation. Section 4.1 describes the basic contents of CTSpec documents. There are several activities necessary to obtain a CTSpec. Section 4.2 deals with the test preparation and the specification activities in the narrower sense. Specifications of conformance tests must undergo a strict validation and verification because they are used to attest conformance against a public standard. Section 4.3 therefore deals with assessing the quality of the conformance test specification. Section 4.4 states criteria for ending the test specification process.

The further development of this section is to be found in Part 3 of the document set [14], where the process of developing (specifying) conformance test is discussed in more detail; in Part 4 [15], where detailed instructions concerning the impact of the methods described on the execution of the specification process are contained; and in the template for a conformance test specification [16].

4.1 Conformance Test Specification Documents

The conformance test specification document (the CTSpec) facilitates communication among several stakeholders[17,18]:

- The *standardization body* (AUTOSAR) responsible for developing and maintaining design specifications and conformance test specifications.
- *Conformance test agencies* (CTAs) for developing and maintaining Conformance Test Suites (CTSs), checking test results and issuing conformance attestations.
- *Product suppliers* (PSs) for implementing the products according to the design specification, executing CTSs and reporting test results.

For this purpose, a conformance test specification comprises

- Test items
- Test procedures
- Test case specifications
- Test architecture specification

Test items are the SWS-items covered by the test cases. *Test procedures* describe sequences of actions for the execution of the test (either automatically or manually) in a general way. Each *Test case specification* references the covered test items and defines inputs, the predicted outputs (the pass criteria), a set of execution conditions, and the concrete test procedures to be applied. Furthermore the test case specifica-

[17] Details regarding the various variants of the conformance attestation processes and the distribution of tasks among the key players (actors playing the key roles) are described elsewhere [10].

[18] Additionally, and beyond the scope of Conformance Test Specification as defined by AUTOSAR, *system integrators* putting all tested system parts together and commissioning further tests for integrated systems have an interest in the content of the CTSpec.

tions can state the purpose and rationale of the test cases. Test case specifications rely on a test object specific architecture of the test system. This architecture of test drivers, test stubs, system adapters, platform adapters and other service components are described in the *Test architecture specification*.

4.2 Preparation and Specification of Conformance Tests

In principle, test specifications orient towards the specific purpose and goals of AUTOSAR conformance tests. In the end the process provides specifications of test cases that follow a black-box test strategy and apply test methods most appropriate to check the characteristics of the test-items for conformance to AUTOSAR design specifications. Before the test specification in the narrower sense can start, there are several tasks to carry out in preparation:

1. Identify test object and the relevant test references
2. Determine all public interfaces of the test object
3. Analyze test references for items that must be tested, i.e., determine test items
4. If necessary call for improvements of the test reference
5. Group test items around operations of the test object resulting in several, minimal sets of related test items

Test preparation tasks can be grouped into the activities *Refinement of the SWS* and *Test Case Identification* (see Part 2 [13], section 3.4.1, and Part 3 [14], section 2). After having finished the test preparation, the test specification starts:

1. Select a test item set and choose test methods and techniques capable to reach the test goals
2. Choose valid inputs, expected outputs and a test procedure comprising steps for test drivers and test stubs
3. Sketch the purpose, rationale and idea of the test and the rough steps of the test cases leading to a preliminary test case design
4. Transform the test case design into formally specified, modular test cases, i.e. test cases that are self-contained and free of side-effects
5. Repeat steps 1 to 4 until all test items are covered at least once

The tasks of the specification phase can be grouped into the activities *Design of CTSpec* and *Test Case Implementation* (see Part 3 [14], section 2). Thereby, to implement a test case in the context of test specification essentially means to write a test case in terms of a test script language with a formally defined semantics (see section 5).

Finally, test cases are allocated to valid variants of the test object in order to simplify the test case selection later on, during actual conformance tests. The result of this process is an unverified conformance specification.

4.3 Validation and Verification of Conformance Tests

Sources of errors are manifold, as are reasons for and types of errors. Not only the test object but also the test reference (design specification, SWS) may contain errors. Test cases, test data, the test system, the runtime-environment (operating system), function libraries and tools used for compiling the test object and the test system may contain errors too. Thus the testware must be checked as well as the test object. This holds especially for obligatory conformance tests that have a large operating range.

The activities and work products can be evaluated during or at the end of the test specification process. Of special importance is the evaluation of the conformance test specification document (the CTSpec). The evaluation consists of validation and verification tasks. Validation tasks determine whether the work products satisfy the requirements imposed on AUTOSAR conformance tests as stated in chapters 3 and 4. Essentially, they answer the question "Are the test cases capable of evaluating the right test-items with proper test methods?". Verification tasks determine whether the test cases are implemented correctly and are error sensitive, i.e., can they correctly indicate:

- Correct behavior in the case that an object under test is correctly implemented with regard to the test cases, and thus avoid *false positives*
- Erroneous behavior in the case that an object under test is flawed with regard to the test cases, and thus avoid *false negatives*

Furthermore, the test case specifications are analyzed for being maintainable – with a special focus on modularity and freedom from side-effects – in order that it may be possible later to easily create and combine sub-sets of the tests, without needing to modify the individual test specifications. This facilitates testing of manifold variants (configurations) of families of test objects as defined in the according specification documents (SWS).

Proper means for validation of CTSpec testware are reviews and inspections of testware by means of checklists derived from the conformance test specification documents. A proper means to verify a conformance test suite (CTS) created from the CTSpecs is to apply it to prototypical implementations of the objects under test. These prototypical objects[19] simulate correct and erroneous behavior of various object variants.

Simulation of erroneous behavior is a form of mutation testing. It amounts to the selective injection of faults into the prototypical test object and then demonstrating that the specified tests are capable to catch the injected faults. Which faults to inject are determined according to the fault model of the applied test methods. Thus the prototypical object implementations could be seen as a means for breaking the specified test suite. A prototypical object breaks a test suite when the test suite fails to reveal faults that the test suite should catch. The kind and number of faults to be injected according to the applied fault model should be based on a risk analysis in the light of the limited test resources available.

4.4 End of the Test Specification Process

There is no single valid, rational criterion for stopping the test process. Almost the same holds for the test specification process. In general, the test specification stops for a dedicated test object when the CTSpec contains validated test specifications for all SWS-items that a conformance test shall cover according to the purpose and goals of conformance tests.

[19] Prototypical Objects will generally be HLL simulations of the test object. It is conceivable that they take the form of TTCN-3 'counter-test-cases', although experience show these to be less practical than HLL simulations.

For conformance test specification, coverage is determined in relation to the test reference and in relation to the simulating test object used for verification purposes. Independently developed simulations of the test object are recommended as a means for test case verification. Every test item must be covered by at least one conformance test case and every test case must be verified and pass test runs against the simulating test object. The tests shall run the simulating test object in operational mode (simulating correct behavior) and in such failure modes (simulating erroneous behavior) as the validation analysis (section 4.3) dictates.

5 Techniques of Conformance Testing

The effectiveness of test specifications depends on the techniques available to implement and apply the tests in an unambiguous way. This chapter gives an overview on the techniques that support or enable the AUTOSAR conformance tests as described in chapters 3 and 4.

AUTOSAR conformance testing is performed to determine the existence and correctness of the basic functionality of a test object in relation to what is specified in AUTOSAR design specifications and declared in its Implementation Conformance Statement (ICS). Testing techniques ensure that the test object satisfies the specification requirements selected for testing according to the general goals and purpose of the tests. In general, testing techniques can be classified – independently of the abstraction level of the test object (module, subsystem or system) – according to the following dimensions:

- functional or structural testing concepts;
- static or dynamic testing concepts;
- manual or automated testing concepts.

Chapter 3 points out the necessity of conformance tests in general to be functional tests treating the test objects as black boxes. Structural testing concepts are only applied for the purpose of verification of conformance test specification (see section 4.3). This chapter examines AUTOSAR conformance tests with regard to the remaining dimensions. Section 5.1 discusses automated vs. manual test concepts and dynamic vs. static test concepts in the context of AUTOSAR conformance tests. Section 5.2 presents the concepts behind abstract test case specifications used to formally define AUTOSAR conformance tests. Section 5.3 highlights consequences of abstract test cases for the architecture of test systems.

The further development of this section of the background document is to be found in Part 4 [15] of the document set, which describes the constraints on the execution of conformance tests that are imposed by the combination of this document and the processes based on it, as described in Parts 2 [13] and 3 [14]. Test techniques are specified for each test case of a conformance test suite (see CTSpec template [16]).

5.1 Basic Test Concepts

The conformance test specification process results in a test suite applying various test methods and techniques. In general, conformance tests will be either dynamic tests or static tests. The application of the test specifications (which is otherwise beyond the scope of this document) will be either automated or manual. Dynamic tests require the test object to be executed, and hence in the end involve programming test cases and test systems (a test environment). Static tests do not involve the execution of the operational test object, but can be automated as well. Automated testing techniques are performed by tools; manual techniques are performed by people.

Fundamentally, AUTOSAR conformance tests are automated, dynamic tests. The more formalized the test references are (usually the design specifications), the more comprehensive the tests and the easier it becomes to automate the tests. However, not all test activities can be automated.

5.1.1 Test Specification Phase

The specification of tests is always a creative engineering process. Thus the test case specification includes many manual activities starting with qualitative analysis of the test reference in order to determine the test-relevant and testable SWS items (test items, see section 4.2). Afterwards proper test methods must be selected and combined for specifying effective and maintainable test cases that check groups of test items with minimal effort.

Test case specifications must include a basic design of the environment (test system) necessary to execute the tests later. Manual inspections can also be used for verifying the design of the test case specifications in conjunction with automated test runs against simulating test objects in a prototypical test environment (section 4.3).

5.1.2 Test Execution Phase

During the test execution phase (during conformance attestation) the contents and structure of interface files (C header files) and configuration descriptions (described in the ICS) can be manually inspected, if no automation is available. Linking and execution of variant-specific conformance tests supports the verification of correct configurations of a test object. For instance, if the test object is missing a required functionality then it will not be possible to link the executable test suite (the test harness) to the test object (provided that the test suite is correctly configured). Errors occurring during execution could result from wrong test object configurations and parameterization.

5.2 Concepts for Specifying Conformance Tests

Dynamic AUTOSAR conformance tests are expressed in a formally defined test script language that enables abstraction from the representations of the test object, test data, clients of the test object and the system environment. The language offers constructs for modularization and parameterization of tests and thus improves the portability, maintainability and configurability of the tests. Test cases look the same independently of the platform (development machine or ECU) and the implementation of the test object (e.g. in C, Java or C++). TTCN-3 provides adequate means of expression [17]. Basic concepts are:

- Test modules with module parameters and the capability to export and import definitions
- Type system supporting test-specific needs like native types for test verdicts and test components (test drivers and test stubs)
- Test data templates for easy matching of expected test results
- Statements supporting test-specific needs, e.g., for the compact handling of stimuli and several alternative answers of the test object
- Test components with ports for communicating with the test object, with other test components or with the system environment in a unified manner
- Procedure-based interfaces or message-based interfaces bound to component ports

Figure 3 illustrates how these concepts can be combined to build a test harness comprising several test drivers and test stubs.

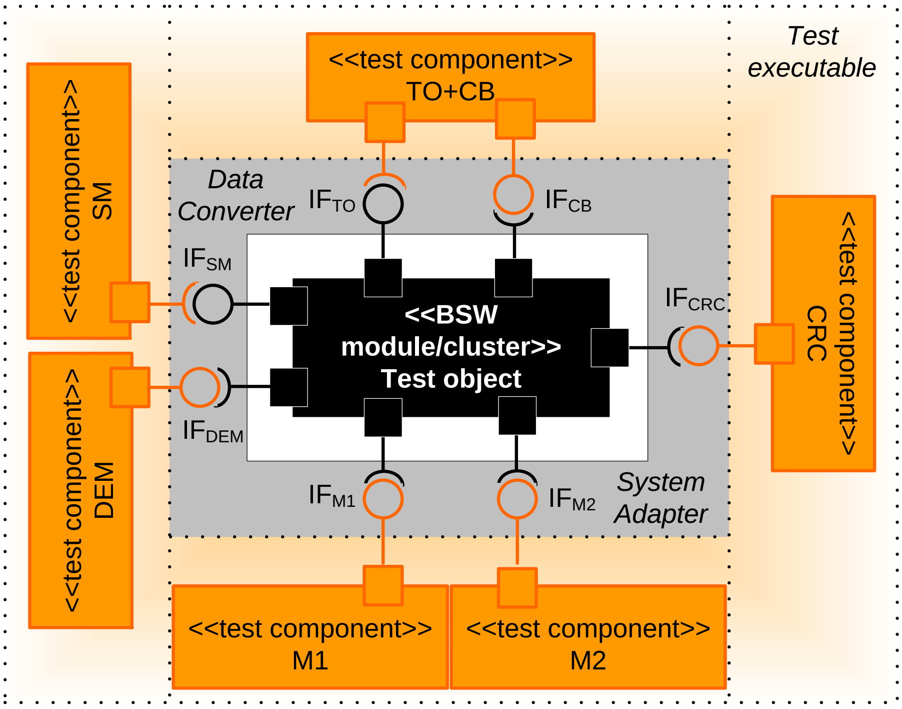

Figure 3 – Test harness for a test object with various provided and required interfaces (IF)

The test object provides interfaces to other BSW parts (IF_{TO}: interface covering the test-object-specific service operations, IF_{SM}: interface with operations for life-cycle management like *Init* and *Main*) and requires interfaces form other BSW parts (e.g., IF_{DEM}: interface with the diagnostic event manager, IF_{CB}: call back interface). *System adapter* and *data converter* completely isolate the test executable from the test object and thus enable the specification of abstract test cases. The test components execute the test cases. Test cases communicate with the test object via component ports. For testing BSW modules or BSW clusters, only procedure-based call and return interfaces are bound to the component ports. For the test object these interfaces could be:

- interfaces of Java objects – applied when verifying the test case specifications (see section 4.3);
- C functions – when testing productive implementations during test execution.

5.3 Architecture of Test Systems

5.3.1 Overview

The concepts of the test script language are reflected in the architecture of typical test systems. Because of the lack of any system-specific information, the abstract test case specifications constitute the center of the architecture (*test executable* in Figure 4) completely isolated from the test object (depicted as a black-box in Figure 4).

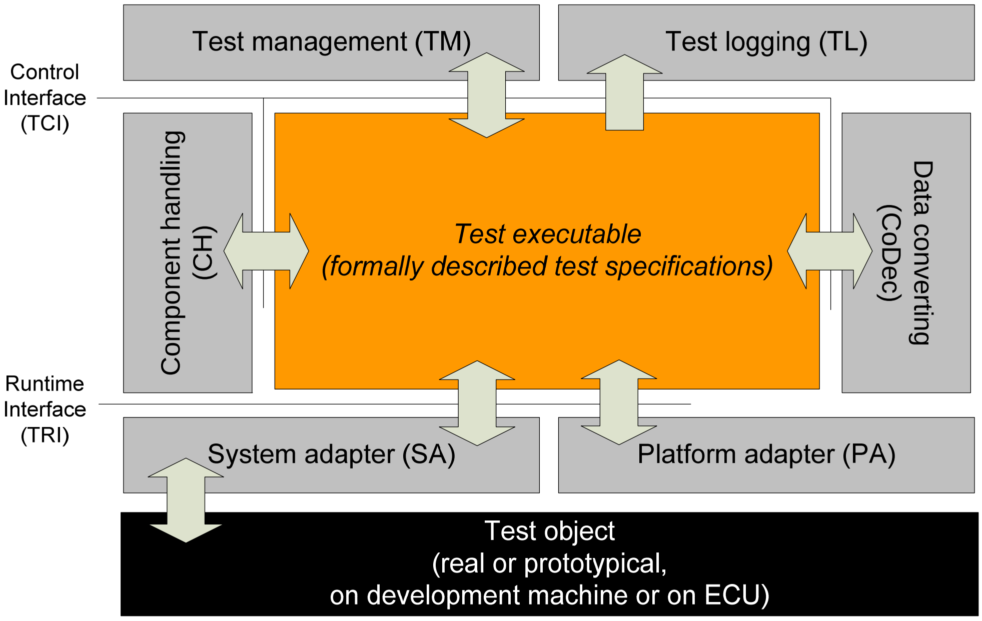

Figure 4 – Schematic overview of test system [17]

The parts of the test system that customize and coordinate test execution (test management), report on the test execution (test logging), handle components (create, connect, start, communicate, stop), code and decode data values (data converting), realize the communication with the test object (system adapter), or provide other test utilities (platform adapter) do not belong to the CTSpec (the result of the test specification process). CTAs must implement these parts in order to create an executable test system (the Conformance Test Suite, or CTS).
Based on its relation to the target hardware, an executable test object falls into one of two classes:

Class A This class is independent from any hardware interfaces and can be accessed using software interfaces (APIs) alone. Therefore, the test object can be tested on the development machine.

Class B This class requires access to hardware (for execution or to perform its function) or is available only as binary code. Therefore, the test object must be tested on specified target hardware.

Because of this classification, two dedicated test architectures can be distinguished as discussed in further details below. Note that the compiled test specifications (compiled TTCN-3 code) shall always be executed on development machines, which is hardware different from the target hardware hosting the test object in production mode.

5.3.2 Test Systems on Development Machines

In the case of Class A testing, the *test executable* accesses the test object via the locally running system adapter. The implication is that the *test executable* and test object run almost side by side on the same hardware. Because of the difficulties of executing TTCN-3 based test code on a microcontroller based target, this hardware is a development machine (a PC). This has a further implication: the test object shall be executable on a PC. In exchange there are some advantages, and these can lead to reduced test effort and shorter feedback cycles:

- No imperative to specify (and later implement) a huge number of target specific tester proxies (see below)
- Usage of established PC development tools for interfacing to the SUT
- No host/target communication

Access to software interfaces is best specified in TTCN-3 using the procedure-based communication principle.

5.3.3 Test Systems on Target Machines

Class B test objects are tested on native microcontroller based hardware. Since the *test executable* still runs on a development machine (a PC), a communication link must be established between the PC and the target platform (see Figure 5, communication link 2) to get access to the interfaces of the test object. To realize this additional communication link, one of the target's native communication links (like communication link 1 in Figure 5) can be reused, or alternatively an additional communication link (e.g. based on USB or Ethernet) can be used. The necessary functionality must be implemented in the system adapter on the PC and an associated *tester proxy* that runs on top of the test object on the target. A *tester proxy* functions as a broker: it accepts an incoming function call request from the PC and performs the actual function call at the test object; it then forwards any results from the function invocation back to the PC. The tester proxy can implement additional, test specific functionality enabling better controllability, observability or dynamic configurability of the test object, e.g., with functions for writing and reading memory cells.

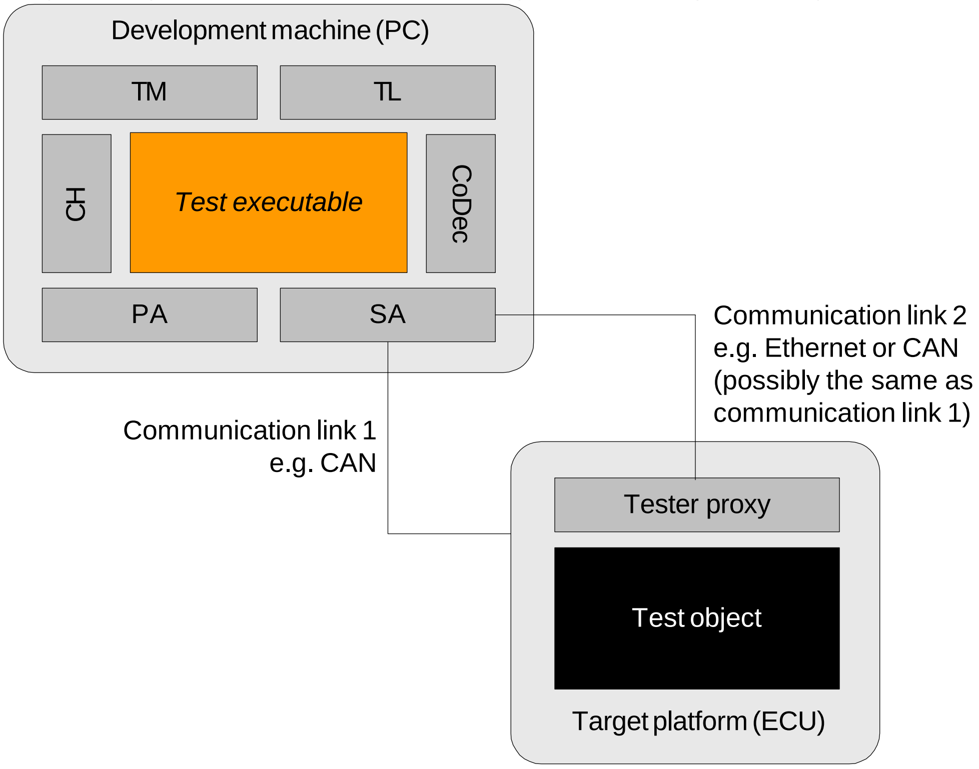

Figure 5 – Test Architecture for Class B Tests

The test cases shall access the interfaces of the test object running on the target platform in the same way as in Class A, in order to support interchangeability of test cases between classes.