

<b>Document Title</b>	Requirements on Interaction with Behavioral Models
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	102
<b>Document Classification</b>	Auxiliary

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Classic Platform
<b>Part of Standard Release</b>	4.3.0

Document Change History			
Date	Release	Changed by	Change Description
2016-11-30	4.3.0	AUTOSAR Release Management	<ul style="list-style-type: none"><li>Editorial changes</li></ul>
2014-10-31	4.2.1	AUTOSAR Release Management	<ul style="list-style-type: none"><li>Editorial changes</li></ul>
2013-10-31	4.1.2	AUTOSAR Release Management	<ul style="list-style-type: none"><li>Editorial changes</li></ul>
2013-03-15	4.1.1	AUTOSAR Administration	<ul style="list-style-type: none"><li>Finalized for Release 4.1</li></ul>
2010-02-02	3.1.4	AUTOSAR Administration	<ul style="list-style-type: none"><li>Legal disclaimer revised</li></ul>
2008-08-13	3.1.1	AUTOSAR Administration	<ul style="list-style-type: none"><li>Legal disclaimer revised</li></ul>
2007-12-21	3.0.1	AUTOSAR Administration	<ul style="list-style-type: none"><li>Document meta information extended</li><li>Small layout adaptations made</li></ul>
2007-01-24	2.1.15	AUTOSAR Administration	<ul style="list-style-type: none"><li>“Advice for users” revised</li><li>“Revision Information” added</li></ul>
2006-11-28	2.1.1	AUTOSAR Administration	<ul style="list-style-type: none"><li>Legal disclaimer revised</li></ul>
2006-05-16	2.0	AUTOSAR Administration	<ul style="list-style-type: none"><li>Initial release</li></ul>

## **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 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	About this document.....	4
1.1	Introduction .....	4
1.2	Terminology .....	4
1.3	About Requirements.....	5
1.3.1	Structure .....	5
1.3.2	Conventions used .....	5
1.3.3	Guidelines .....	6
2	Requirements .....	8
2.1	[RS_ATBM_00015] Define interaction.....	8

# 1 About this document

## 1.1 Introduction

This document defines requirements on the deliverable “Specification of Interaction with Behavioral Models”.

## 1.2 Terminology

In this section the terminology as used throughout this document is defined. The definitions are to some extent specific for the scope of AUTOSAR and especially for this deliverable. Common use of these terms, however, is taken into account as far as possible.

- An **Authoring Tool** is an AUTOSAR tool operating on any form of AUTOSAR models describing systems (software component, ECU hardware, network topology and system constraint descriptions). It is regarded to be a design entry tool for AUTOSAR descriptions according to the respective templates. Typical functions may include creating, retrieving, modifying, validating and storing such descriptions. An authoring tool may provide a tool specific language or notations for the design entry, typically used as the expressive language at the tool's user interface. These languages might differ from those used for AUTOSAR standard description formats. E.g. a graphical behavior modeling tool could be used to edit a software component description using a behavior modeling language, stored as an XML file according to the software component template. Being an authoring tool is thus more a dedicated role of a tool than a classification of a tool itself.
- **AUTOSAR Model** is a generic expression for any kind of representation of instances of the AUTOSAR meta model. It might be a set of files in a file system, an XML stream, a database or memory used by some running software, etc.
- **AUTOSAR Tools** are software tools that may occur within the AUTOSAR methodology and support interpreting, processing and/or creating of AUTOSAR models
- **AUTOSAR Authoring Tools** are AUTOSAR tools operating on any form of AUTOSAR models describing systems (software component, ECU hardware, network topology and system constraint descriptions).
- **Behavior** is used in two major variants. On the one hand, behavior is used as an abbreviation for `InternalBehavior` – as part of a description of a software component template. On the other hand, behavior is a common control engineering term used to identify the functional input/output relation of a control design over time. Throughout this document the term behavior and combinations of this term like behavior models should be understood in this control engineering interpretation. To avoid any misunderstandings the term functional behavior will be used – opposed to `InternalBehavior`.

- A **Behavior Model (BM)** is a specification or design model expressed in a functional behavior modeling language.
- A **Behavior Modeling Language (BML)** is a (often graphical) notation primarily used to capture a functional behavior specification or design of a function or system. Usually, a functional behavior modeling language is regarded to be executable, i.e. its semantics is sufficiently precise to execute functional behavior models by means of a simulation engine. Furthermore, the precision in its semantics then allows the transformation of a functional behavior model into a source code of in a programming language like C. Many functional behavior modeling languages are based on finite state machine or data flow semantics.
- A **Behavior Modeling Tool (BMT)** is used to edit a functional behavior model in a functional behavior modeling language.
- A **Model Frame** is a container for a functional behavior model consisting of structural building blocks, usually called subsystems or modules. A model frame is a contract of the functional behavior model with its environment, thus it can be regarded as the counterpart of a software component template as introduced by AUTOSAR.

## 1.3 About Requirements

Each requirement has its unique identifier starting with the prefix "RS\_ATBM\_" (meaning **RE**quirement **of** **A**uthoring **T**ool).

### 1.3.1 Structure

Each requirement is defined as a table. The structure of the tables is as follows:

<b>Initiator:</b>	< number of originating work package, company, etc >
<b>Date:</b>	< date of last change >
<b>Requirement:</b>	< the normative text of the requirement >
<b>Description:</b>	< detailed description of the requirement >
<b>Rationale:</b>	< why is this necessary, what its omission could cause >
<b>Use Case:</b>	< example to a scenario that makes the requirement necessary or useful >
<b>Dependencies:</b>	< reference to depending and depended-on requirements >
<b>Conflicts:</b>	< reference to conflicting requirement >
<b>Supporting Material:</b>	< links to other documents >
<b>Comment:</b>	< additional remarks >

### 1.3.2 Conventions used

In requirements, the following specific semantics are used (taken from Request for Comment RFC 2119 from 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 described in RFC 2119. Note that the requirement level of the document in which they are used modifies the force of these words.

- **MUST:** This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- **MUST NOT:** This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the specification.
- **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.)

### 1.3.3 Guidelines

Existing specifications shall be referenced (in form of a single requirement). Differences to these specifications are specified as additional requirements.

All Requirements shall have the following properties:

- **Redundancy**  
Requirements shall not be repeated within one requirement or in other requirements
- **Clearness**  
All requirements shall allow one possibility of interpretation only. Used technical terms that are not in the glossary must be defined.
- **Atomicity**  
Each Requirement shall only contain one requirement. A Requirement is atomic if it cannot be split up in further requirements.

- Testability  
Requirements shall be testable by analysis, review or test.
- Traceability  
The source and status of a requirement shall be visible at all times.

## 2 Requirements

This chapter provides a definition of the relevant requirements.

### 2.1 [RS\_ATBM\_00015] Define interaction

<b>ID:</b>	RS_ATBM_00015
<b>Initiator:</b>	WP Authoring Tools
<b>Date:</b>	04.02.2005
<b>Requirement:</b>	Define interaction
<b>Description:</b>	Concept for interaction between behavior models and AUTOSAR descriptions have to be provided.
<b>Rationale:</b>	To support model driven approaches, the coupling between AUTOSAR interface descriptions of software components and behavioral models, modeled in behavior modeling tools like Simulink, TargetLink, ASCET-SD, have to be defined.
<b>Use Case:</b>	Modeling of the behavior of software components with behavior modeling tools, based on the interface description of the software component. Take an existing behavioral model and create a software-component according to the structure of the behavioral model.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--
<b>Comment:</b>	--
<b>Contributes to:</b>	--