Document Title	Requirements on Interaction with Behavioral Models
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	102
Document Classification	Auxiliary

Document Version	1.0.4
Document Status	Final
Part of Release	3.1
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
23.06.2008	1.0.4	AUTOSAR Administration	Legal disclaimer revised
31.10.2007	1.0.3	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
24.01.2007	1.0.2	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added
28.11.2006	1.0.1	AUTOSAR Administration	Legal disclaimer revised
10.05.2006	1.0.0	AUTOSAR Administration	Initial release

Page left intentionally blank

Disclaimer

This document of a specification as released by the AUTOSAR Development Partnership is intended **for the purpose of information only**. The commercial exploitation of material contained in this specification requires membership of the AUTOSAR Development Partnership or an agreement with the AUTOSAR Development Partnership. The AUTOSAR Development Partnership will not be liable for any use of this specification. Following the completion of the development of the AUTOSAR specifications commercial exploitation licenses will be made available to end users by way of written License Agreement only.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher." The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Copyright © 2004-2008 AUTOSAR Development Partnership. All rights reserved.

Advice to users of AUTOSAR Specification Documents:

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 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 extend 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 "ATREQ" (meaning **A**uthoring **T**ools **REQ**uirement).

1.3.1 Structure

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

Initiator:	< number of originating work package, company, etc >
Date:	< date of last change >
Requirement:	< the normative text of the requirement >
Description:	< detailed description of the requirement >
Rationale:	< why is this necessary, what its omission could cause >
Use Case:	< example to a scenario that makes the requirement necessary or useful >
Dependencies:	< reference to depending and depended-on requirements >
Conflicts:	< reference to conflicting requirement >
Supporting Material:	< links to other documents >
Comment:	< 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 [ATREQ_015] Define interaction

Initiator:	WP1.2
Date:	04.02.2005
Requirement:	Define interaction
Description:	Concept for interaction between behavior models and AUTOSAR descriptions have to be provided.
Rationale:	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.
Use Case:	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.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Comment:	--