Document Title	Requirements on Graphical Notation
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	103
Document Classification	Auxiliary

Document Version	1.0.3
Document Status	Final
Part of Release	3.0
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
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

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 document “Graphical Notation”.

1.2 Terminology

- The **AUTOSAR metamodel** is a UML2.0 model that defines the language for describing AUTOSAR systems. The AUTOSAR metamodel is a graphical representation of a template. UML2.0 class diagrams are used to describe the attributes and their interrelationships. Stereotypes and OCL (object constraint language) are used for defining specific semantics and constraints.
- An **AUTOSAR model** is an instance of the AUTOSAR metamodel. The information contained in the AUTOSAR model can be anything that is representable according to the AUTOSAR metamodel. The AUTOSAR model can be stored in many different ways: it might be a set of files in a file system, an XML stream, a database or memory used by some running software tools, etc.
- The **AUTOSAR XML Schema** is a W3C XML schema that defines the language for exchanging AUTOSAR models. This Schema is derived from the AUTOSAR metamodel and defines the **AUTOSAR data exchange format**.
- An **AUTOSAR XML description** describes the XML representation of an AUTOSAR model. The AUTOSAR XML description can consist of several fragments (e.g. files). Each individual fragment must validate successfully against the AUTOSAR XML schema.
- An **AUTOSAR authoring tool** is a software tool which supports interpreting, processing and creating of AUTOSAR XML descriptions.
- **Metadata** includes pertinent information **about** data, including information about the authorship, versioning, access-rights, timestamps etc.

1.3 About Requirements

Each requirement has its unique identifier starting with the prefix "ATIREQ" (meaning **A**uthoring **T**ools **I**nteroperability **Req**uirements).

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_020] Define a graphical notation

Initiator:	WP1.2
Date:	02.08.2005
Requirement:	AUTOSAR SHALL provide a definition of graphical symbols for a set of AUTOSAR metamodel elements.
Description:	All tools SHALL support a common graphical notation for AUTOSAR model. AUTOSAR SHALL define a common graphical notation reduces the learning curve for all users for different tools. The document SHALL focus on an agreed set data exchange use-cases.
Rationale:	Within the AUTOSAR methodology AUTOSAR models will be exchanged between different parties. Each party could use different tools which best fit to the step in the methodology. In order to facilitate seamless exchange of AUTOSAR models, a standardized AUTOSAR graphical notation is required.
Use Case:	Facilitate the use of different tools from different tool vendors within the AUTOSAR methodology.
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Comment:	--

2.2 [ATREQ_019] Define diagrams

Initiator:	WP1.2
Date:	02.08.2005
Requirement:	AUTOSAR SHALL provide a description how the graphical notations can be grouped for diagrams.
Description:	AUTOSAR SHALL provide a definition on how to group graphical symbols into a diagram.
Rationale:	The mere definition of graphical symbols is not enough for modeling. Not all graphical symbols can be combined in a sensible way. A “diagram taxonomy” defines how the symbols can be used to draw different diagram types.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Comment:	--

2.3 [ATREQ_021] Define diagram exchange

Initiator:	WP1.2
Date:	02.08.2005
Requirement:	AUTOSAR SHOULD provide a concept for the interchange of diagrams between different tools.
Description:	All tools that make use of a graphical notation SHOULD be able to exchange the layout of their diagrams to preserve the readability within the steps of the methodology. The document “Graphical Notation” shall provide the concepts to be used for that interchange. Since the standard data exchange in AUTOSAR is based on import/export of XML files, it is reasonable to define layout information exchange in a similar manner.
Rationale:	Diagram exchange is needed to avoid loss of layout information between tools supporting graphical representation of AUTOSAR models. Diagram exchange method must be defined because layout information is not part of the Metamodel. Therefore it is not included in the AUTOSAR descriptions either.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Comment:	--

2.4 [ATREQ_023] Base Graphical on UML2

Initiator:	PL team
Date:	02.08.2005
Requirement:	AUTOSAR SHOULD define a graphical notation that is based on UML 2.
Description:	AUTOSAR SHOULD define a graphical notation that is based on industry standards. UML 2 is the most prominent modelling standard in the software engineering industry.
Rationale:	A graphical notation based on a widely used notation is easily understood and can be understood quickly.
Use Case:	--
Dependencies:	--
Conflicts:	--
Supporting Material:	--
Comment:	--