Document Title	Generic Structure Template
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	202
Document Classification	Auxiliary

Document Version	2.1.1
Document Status	Final
Part of Release	3.1
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
23.06.2008	2.1.1	AUTOSAR Administration	• Legal disclaimer revised • Rename document from “Template Modelling Patterns” to “Generic Structure Template”
20.12.2007	2.1.0	AUTOSAR Administration	• Updated Attributes of Identifiable • Added "Hint to the Users" • Added document identification no • Added document classification
07.02.2007	2.0.2	AUTOSAR Administration	• Legal disclaimer revised • “Advice for users” revised • “Revision Information” added
27.06.2006	2.0.1	AUTOSAR Administration	Layout Adaptations
19.05.2006	2.0.0	AUTOSAR Administration	Second release
18.10.2005	1.0.0	AUTOSAR Administration	Initial release

Special Note:

The current version of this document describes only a subset of the common modeling elements which are available in the package *GenericStructure* of the Meta-Model for AUTOSAR R3.0 [4].

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.

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 Introduction

1.1 Origins and Goals

AUTOSAR attempts to allow for a very flexible yet stable and reliable software engineering lifecycle through precise and formal description of all relevant aspects of a distributed system of embedded controllers and the corresponding executed software units.
The descriptions range from high level requirements on interfaces of software components to low level constraints on certain bits of a specific bus message. Various work packages in AUTOSAR determine the information that needs to be captured in the different descriptions.
For instance, in [5] it is defined how AUTOSAR software components need to be described, e.g. when a requirements specification is exchanged between OEM and supplier. The collection of attributes required specifying various AUTOSAR relevant artifacts like software components, ECUs and so on is called an *AUTOSAR template*. Once information is available a template is said to be filled out, leading to an *AUTOSAR description*.
The AUTOSAR template modeling guide [3] provides the formal means to define an AUTOSAR template. The various templates in AUTOSAR typically require a certain set of common features (e.g. a common byte order definition). Those features are collected in this document.
Therefore the common patterns for template model can be understood as an AUTOSAR standard library of modeling elements and patterns.

1.2 Change Process

Due to the nature of the document it is expected that not for each and every addition to the set of known patterns a full milestone cycle needs to be passed. Instead, new features may be requested for inclusion through AUTOSAR change requests.

1.3 Terminology

In this specification the key words MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when EMPHASIZED are to be interpreted as described in RFC 2119 [1].

The attributes of the classes introduced in this document are listed in form of class tables. They have the form shown in the example of the top-level element *AUTOSAR*:

Class	**AUTOSAR**			
Package	M2::AUTOSARTemplates::GenericStructure::Infrastructure::Autosar			
Class Description	Root element of an AUTOSAR description, also the root element in corresponding XML documents.			
Base Class(es)	ARObject			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
adminData	AdminData	0..1	aggregation	--
topLevelPackage	ARPackage	0..*	aggregation	Top level packages of a description.

The headers in the table have the following meaning:

Class: The name of the class as defined in the UML model.

Package: The UML package the class is defined in. This is only listed to help locating the class in the overall metamodel.

Class Description: The comment the modeler gave for the class.

Base Classes: If applicable, the list of direct base classes.

Attribute: The name of an attribute of the class. Note that AUTOSAR does not distinguish between class attributes and owned association ends.

Datatype: The datatype of an attribute of the class.

Multiplicity: The assigned multiplicity of the attribute, i.e. how many instances of the given datatype are associated with the attribute.

Link Type: Specifies, whether the attributes is part of the class (aggregation) or just referenced by it (reference). Instance references are also indicated in this field (refer to [3] for details).

Attribute Description: The comment the modeler gave for the class attribute.

2 Common Patterns

The nature of the common patterns given below is similar to the standard library of a compiler: a set of predefined structures and elements to be used in an AUTOSAR template model.

2.1 Primitive Types

In addition to the fundamental primitive types given in [3], more content related types are offered for usage in template models. They are:

DateTime: This datatype represents a timestamp. While the actual representation of its values depends on the platform used (e.g. XML, database systems, …) the definitions of timestamps defined in ISO 8601 [7] SHOULD be followed.

Identifier: This datatype represents a string that follows the rules of C-Identifiers.

2.2 Identifiable Properties

The base class *Identifiable* defined in [3] has further content related attributes, which are shown in Figure 1:

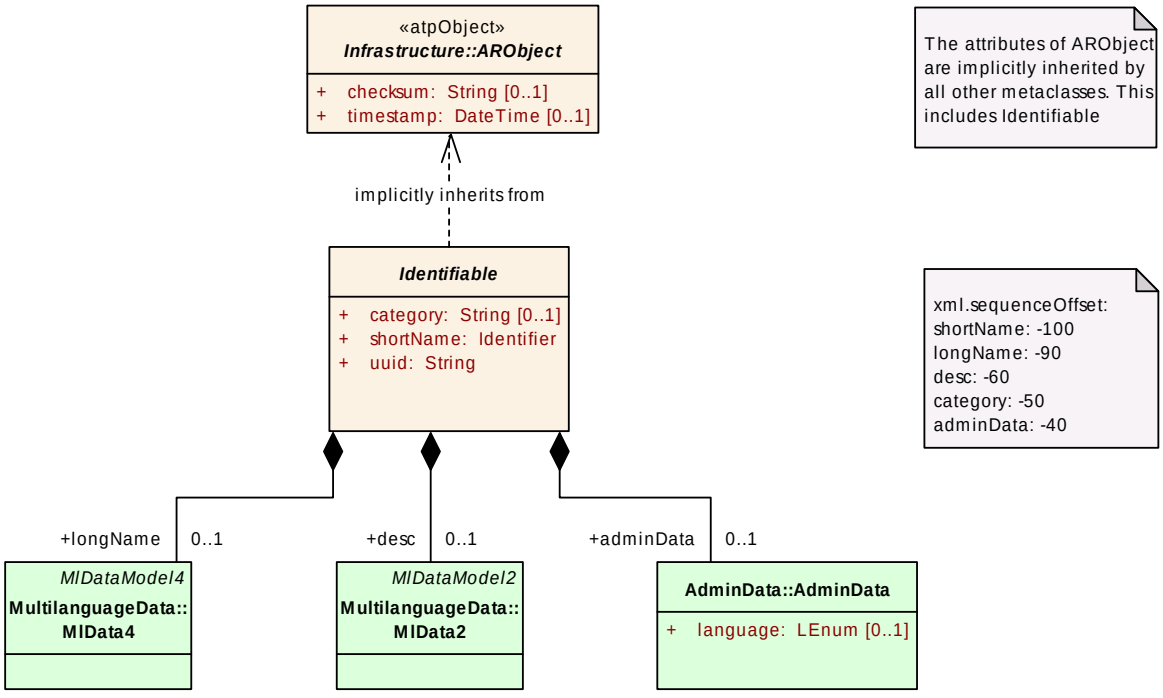

Figure 1: Base class *Identifiable* and its content relevant attributes.

The attributes are given in the following class tables:

Class	**Identifiable {abstract}**			
Package	M2::AUTOSARTemplates::GenericStructure::Infrastructure::Identifiable			
Class Description	Instances of this class can be referred to by their identifier (while adhering to namespace borders).			
Base Class(es)	ARObject			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
shortName	Identifier	1	aggregation	Use <shortName> to generate a short name for the context element, which enables it to be ** .
longName	MlData4	0..1	aggregation	Use <longName> to create a comprehensive name for the context element.
desc	MlData2	0..1	aggregation	<desc> represents a general but brief description of the object in question.
category	String	0..1	aggregation	This element assigns a category to the parent element. The category can be used by a semantic checker in post-processes to ensure that the parent object is defined correctly i.e. has the right number of elements for example.
adminData	AdminData	0..1	aggregation	<adminData> can be used to set administrative information for an element. This administration information is to be treated as metadata such as revision id or state of the file. There are basically four kinds of metadata * The language and/or used laguages. * Revision information covering e.g. revision number, state, release date, changes. Note that this information can be given in general as well as related to a particular company. * Document metadata specific for a company * Formatting controls that can affect layouts for example. * Revision information for the element.
uuid	String	1	aggregation	The purpose of this attribute is to provide a globally unique identifier for an instance of a metaclass. The values of this attribute should be globally unique strings prefixed by the type of identifier. For example, to include a DCE UUID as defined by The Open Group, the UUID would be preceded by "DCE:". The values of this attribute may be used to support merging of different AUTOSAR models. The form of the UUID (Universally Unique Identifier) is taken from a standard defined by the Open Group (was Open Software Foundation). This standard is widely used, including by Microsoft for COM (GUIDs) and by many companies for DCE, which is based on CORBA. The method for generating these 128-bit IDs is published in the standard and the effectiveness and uniqueness of the IDs is not in practice disputed. If the id namespace is omitted, DCE is assumed. An example is "DCE:2fac1234-31f8-11b4-a222-08002b34c003".

2.2.1 ShortName

The shortName of an identifiable element is defined by the following regular expression: ([a-zA-Z][a-zA-Z0-9_]{0,31})+

2.2.2 LongName

The longName of identifiable elements can be described in different languages. For each language a L4 element may be described. This element indicates the language of the long name.

Class	**MlData4**
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::MultilanguageData
Class Description	MultilanguageLongName
Base Class(es)	ARObject, MlDataModel4

Class	**MlDataModel4 {abstract}**			
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::LanguageDataModel			
Class Description	MultilanguageDataModelForLongNames			
Base Class(es)	ARObject			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
l4	L4	1..*	aggregation	--

Description

Class	**<<atpMixedString>> L4**			
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::LanguageDataModel			
Class Description	MixedContentForLongNames in one particular language. The language is dentoted in the attribute l.			
Base Class(es)	ARObject, MixedContent4			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
l	LEnum	0..*	aggregation	--

2.2.3 Description

The description of identifiable elements can be described in different languages. For each language a L4 element may be described. This element indicates the language of the description.

Class	**MlData2**
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::MultilanguageData
Class Description	MultiLanguageOverviewParagraph This is the content of a multilingual paragraph in an overview item.
Base Class(es)	ARObject, MlDataModel2

Class	**MlDataModel2 {abstract}**			
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::LanguageDataModel			
Class Description	MultilanguageDataModelForOverviewParagraph			
Base Class(es)	ARObject			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
l2	L2	1..*	aggregation	--

Class	**<<atpMixedString>> L2**			
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::LanguageDataModel			
Class Description	MixedContentForOverviewParagraph in one particular language. The language is denoted in the attribute l.			
Base Class(es)	ARObject, MixedContent2			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
l	LEnum	0..*	aggregation	--

2.2.4 AdminData

Class	**AdminData**			
Package	M2::AUTOSARTemplates::GenericStructure::CommonPatterns::AdminData			
Class Description	<adminData> can be used to set administrative information for an element. This administration information is to be treated as metadata such as revision id or state of the file. There are basically four kinds of metadata * The language and/or used laguages. * Revision information covering e.g. revision number, state, release date, changes. Note that this information can be given in general as well as related to a particular company. * Document metadata specific for a company * Formatting controls that can affect layouts for example. * Revision information for the element.			
Base Class(es)	ARObject			
Attribute	***Datatype***	***Mul.***	***Link Type***	***Attribute Description***
language	LEnum	0..1	aggregation	This attribute specifies the master language of the document or the document fragment. The master language is the one in which the document is maintained and from which the other languages are derived from.
usedLanguages	MlData10	0..1	aggregation	Use <usedLanguages> to enter all other languages, in addition to the document language, which are used an a document or a section of a document.
docRevision	DocRevision	0..*	aggregation	Use <docRevision> , to generate information on the corresponding document version.
sdg	Sdg	0..*	aggregation	<sdg> (Special Data Group) is a backdoor used to handle elements that has not yet been defined in a DTD. The <sdg> is a containder for one or several <sd> that defines new elements and carries the information. Special Data should only be used moderately since all elements should be defined in the dtd. Thereby should SDG be considered as a temporary sollution when elements are missing. If a <sdgCaption> element is created along with a <shortName> it is possible to reference the <sdg> structure via a <xref>.

2.3 Byte Order

When more than one byte is stored in the memory the order of those bytes may differ depending on the architecture of the processing unit. If the least significant byte is stored at the lowest address this architecture is called little endian and otherwise it is called big endian.
ByteOrder is very important in case of communication between different PUs or ECUs.
If specifying a byte-order, template classes MUST use the following enumeration:

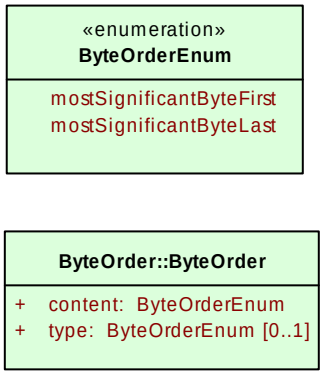

Figure 2: Enumeration to indicate byte order of a processing unit.

3 References

In this section the references used in this specification are listed. They are separated into normative and non-normative (informative) references.

3.1 Normative References

[1] Key words for use in RFCs to Indicate Requirement Levels. Network Working Group, S. Brandner, Harvard University, 1997.
http://www.ietf.org/rfc/rfc2119.txt

[2] Guide to the SI, with a focus on usage and unit conversions: NIST Special Publication 811, 1995 Edition, by Barry N. Taylor. Guide for the Use of the International System of Units (SI).

[3] Template UML Profile and Modeling Guide
https://svn2.autosar.org/repos2/22_Releases/
AUTOSAR_TemplateModelingGuide.pdf.

[4] AUTOSAR Meta-Model
https://svn2.autosar.org/repos2/22_Releases/
AUTOSAR_MetaModel.EAP

3.2 Informative References

[5] Software Component Template,
https://svn2.autosar.org/repos2/22_Releases/
AUTOSAR_SoftwareComponentTemplate.pdf

[6] MSR-SW Element Attribute Documentation V2.2.2.
http://www.msr-wg.de/medoc/download/

[7] ISO 8601:2004, Representation of dates and times, published ISO standard,
http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874