

<b>Document Title</b>	Modeling Guidelines of Basic Software UML Model
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	117
<b>Document Classification</b>	Auxiliary

<b>Document Version</b>	1.2.2
<b>Document Status</b>	Final
<b>Part of Release</b>	3.2
<b>Revision</b>	1

<b>Document Change History</b>			
<b>Date</b>	<b>Version</b>	<b>Changed by</b>	<b>Change Description</b>
23.03.2011	1.2.2	AUTOSAR Administration	Legal disclaimer revised
23.06.2008	1.2.1	AUTOSAR Administration	Legal disclaimer revised
12.11.2007	1.2.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Added description for range stereotype</li> <li>• Change Requirements for function parameter and structure attributes</li> <li>• Document meta information extended</li> <li>• Small layout adaptations made</li> </ul>
05.12.2006	1.1.0	AUTOSAR Administration	<ul style="list-style-type: none"> <li>• Usage of packages clarified</li> <li>• Sequence diagram modelling clarified</li> <li>• Legal disclaimer revised</li> </ul>
27.06.2006	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	Scope of this Document.....	5
2	Related Documentation .....	6
2.1	Deliverables of AUTOSAR work packages .....	6
2.2	Related standards and norms.....	6
3	Terms and abbreviations .....	7
4	Requirements on the modeling of the Basic Software .....	8
4.1	General .....	8
4.1.1	[BSW_UMLGuide_00017] UML 2.0 .....	8
4.1.2	[BSW_UMLGuide_00065] Application of the BSW UML Profile .....	8
4.1.3	[BSW_UMLGuide_00001] Allowed elements.....	9
4.1.4	[BSW_UMLGuide_00002] Allowed relationships .....	10
4.1.5	[BSW_UMLGuide_00053] Allowed set of diagrams.....	11
4.1.6	[BSW_UMLGuide_00060] Documentation of elements .....	11
4.1.7	[BSW_UMLGuide_00047] Links between diagrams shall be hyperlinks. ....	11
4.2	Structural Design .....	12
4.2.1	[BSW_UMLGuide_00054] Use of Packages.....	12
4.2.2	[BSW_UMLGuide_00003] Diagrams usage.....	13
4.2.3	[BSW_UMLGuide_00038] Component diagram appearance options.....	13
4.2.4	[BSW_UMLGuide_00039] Component diagram appearance of BSW module diagrams .....	14
4.2.5	[BSW_UMLGuide_00004] Header File Modeling.....	15
4.2.6	[BSW_UMLGuide_00005] Basic Software Module Modeling.....	16
4.2.7	[BSW_UMLGuide_00052] Interface creation .....	17
4.2.8	[BSW_UMLGuide_00006] Interface Modeling .....	18
4.2.9	[BSW_UMLGuide_00009] Version numbers of software modules.....	18
4.2.10	[BSW_UMLGuide_00010] Component Definition .....	19
4.2.11	[BSW_UMLGuide_00011] Accessing interfaces of other components ...	19
4.2.12	[BSW_UMLGuide_00055] Use of parameter kind .....	20
4.2.13	[BSW_UMLGuide_00061] Definition of return type.....	20
4.2.14	[BSW_UMLGuide_00037] Definition of pointer types .....	20
4.2.15	[BSW_UMLGuide_00027] Definition of structures .....	21
4.2.16	[BSW_UMLGuide_00025] 'Language' definition of Components.....	22
4.2.17	[BSW_UMLGuide_00026] Definition of enumerations .....	23
4.2.18	[BSW_UMLGuide_00028] Definition of simple types.....	24
4.2.19	[BSW_UMLGuide_00059] Definition of typedefs .....	25
4.2.20	[BSW_UMLGuide_00066] Defintion of ranges for typedefs .....	25
4.2.21	[BSW_UMLGuide_00062] Definition of functions and callbacks.....	25
4.2.22	[BSW_UMLGuide_00063] Definition of scheduled functions .....	26
4.2.23	[BSW_UMLGuide_00035] Sub elements of BSW modules .....	26
4.3	Behavioral Design.....	27
4.3.1	General .....	27
4.3.1.1	[BSW_UMLGuide_00030] Usage of Sequence Diagrams.....	27
4.3.1.2	[BSW_UMLGuide_00031] Usage of State Machine Diagrams .....	28
4.3.2	Sequence Diagrams .....	28

4.3.2.1	[BSW_UMLGuide_00012] Location of Sequence Diagrams .....	28
4.3.2.2	[BSW_UMLGuide_00020] Packages to contain sequence diagrams.....	28
4.3.2.3	[BSW_UMLGuide_00021] Commenting of Sequence Diagrams.....	29
4.3.3	[BSW_UMLGuide_00057] Parameter values in sequence diagrams.....	29
4.3.3.1	[BSW_UMLGuide_00058] Return values in sequence diagrams .....	30
4.3.3.2	[BSW_UMLGuide_00018] Modeling of data copying.....	31
4.3.3.3	[BSW_UMLGuide_00019] Labeling returns.....	32
4.3.3.4	[BSW_UMLGuide_00036] Linking sequence diagrams.....	32
4.3.4	State Machine Diagrams.....	33
4.3.4.1	[BSW_UMLGuide_00041] States shall have thick lines .....	33
4.3.4.2	[BSW_UMLGuide_00042] A trigger condition shall be defined for each transition .....	34
4.3.4.3	[BSW_UMLGuide_00043] Transitions may be modeled with sub- activities.....	34
4.3.4.4	[BSW_UMLGuide_00046] Links to parent diagrams shall be drawn as hyperlink diagram references.....	36
4.3.5	Activity Diagrams .....	36
4.3.5.1	[BSW_UMLGuide_00048] Activities shall have thin lines .....	36
4.3.5.2	[BSW_UMLGuide_00049] Conditions to be defined for each branch.....	37
4.3.5.3	[BSW_UMLGuide_00050] Activities to be re-used in sequence diagrams should also be drawn as sequence diagrams .....	37
4.4	Model synchronization .....	38
4.4.1	[BSW_UMLGuide_00013] Creating a Design Master .....	38
4.4.2	[BSW_UMLGuide_00023] Design Master naming convention.....	38
4.4.3	[BSW_UMLGuide_00014] Creating replicas from the Design Master.....	39
4.4.4	[BSW_UMLGuide_00022] Replica naming convention.....	39
4.5	Documentation generation.....	39
4.5.1	[BSW_UMLGuide_00067] Providing an alternative name for generated tables .....	39
5	BSW UML Profil.....	41
5.1.1	Stereotypes callback, function and scheduled function.....	41
5.1.2	Stereotypes module, type, typedef and structure.....	41
5.1.3	Stereotypes mandatory, configurable and optional.....	42
5.1.4	Stereotype range .....	43
6	Administrative Info .....	44

## 1 Scope of this Document

This Modeling Guideline contains guidelines for the usage of the Enterprise Architect UML Modeling Tool (EA) that is used for the detailed architecture design of the AUTOSAR Basic Software.

Each guideline has its unique identifier starting with the prefix “BSW\_UMLGuide” (BSW = Basic Software). For any review annotations, remarks or questions please refer to this unique ID rather than chapter or page numbers!

## **2 Related Documentation**

### **2.1 Deliverables of AUTOSAR work packages**

- [1] List of Basic Software Modules  
AUTOSAR\_BasicSoftwareModules.pdf
- [2] General Requirements on Basic Software Modules  
AUTOSAR\_SRS\_General.pdf
- [3] Layered Software Architecture  
AUTOSAR\_LayeredSoftwareArchitecture.pdf

### **2.2 Related standards and norms**

- [4] UML 2.0, Unified Modeling Language: Superstructure, Version 2.0, OMG document formal/05-07-04."

### 3 Terms and abbreviations

<b>Terms</b>	<b>Definitions</b>
BSW Module Component	Each BSW module is modeled using one "UML Component" and several "Interface Classes" within the BSW UML model. The "UML Component" represents the internal behavior or C-file(s) of the BSW module. It is called "BSW Module Component"
UML Component	Model element defined by <a href="#">[4]</a> .
Interface class	UML 2.0 class with stereotype "interface".
BSW Module Interface	Each BSW module is modeled using one "UML Component" and several "Interface Classes" within the BSW UML model. The "Interface classes" represent the header files of a specific BSW module. They are called "BSW Module Interfaces"
Tree view	The "project view" window within the Enterprise Architect is called "Tree view".

## 4 Requirements on the modeling of the Basic Software

### 4.1 General

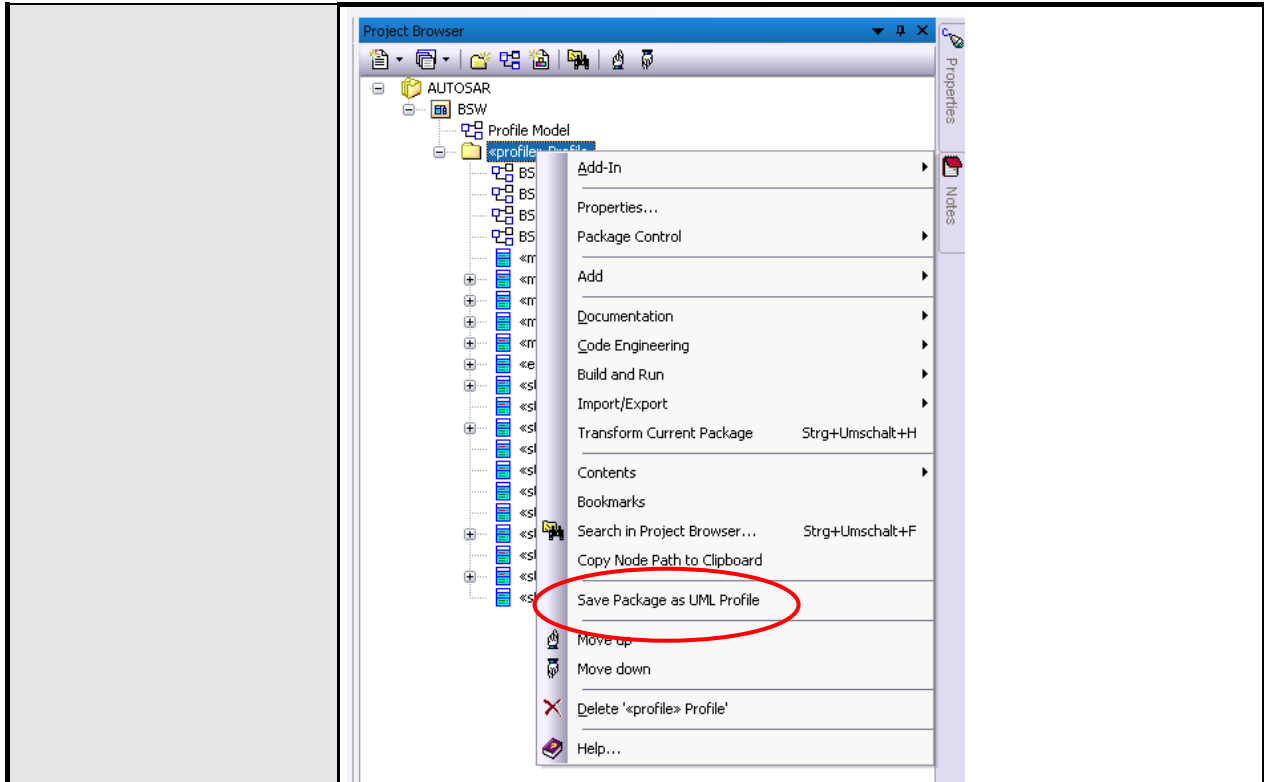
#### 4.1.1 [BSW\_UMLGuide\_00017] UML 2.0

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	14.10.2004
<b>Short Description:</b>	UML 2.0 shall be used for modeling the BSW UML model.
<b>Type:</b>	Changed according to bug #7277
<b>Importance:</b>	high
<b>Description:</b>	The UML specification 2.0 shall be used for modeling the AUTOSAR Basic Software with the tool Enterprise Architect (EA).
<b>Rationale:</b>	Not defining new modeling techniques when there are techniques already available and standardized.
<b>Use Case:</b>	Modeling the Basic Software of AUTOSAR.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

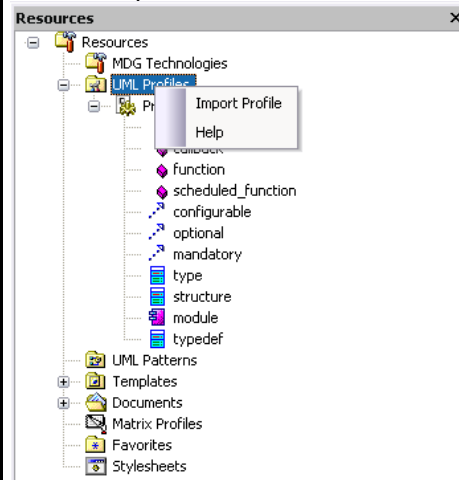
#### 4.1.2 [BSW\_UMLGuide\_00065] Application of the BSW UML Profile

<b>Initiator:</b>	Technical Office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	BSW UML profile
<b>Type:</b>	Application of the BSW UML Profile
<b>Importance:</b>	high
<b>Description:</b>	The latest version of the BSW UML profile has to be applied to the BSW UML model. When a new version of the BSW UML profile has been released, it has to be reapplied to the BSW UML model. To apply the profile, it has to be exported to XML via the "Save Package as UML Profile" menu entry.





The exported XML file then has to be reimported on the BSW UML model file via the “Import Profile” menu.



<b>Rationale:</b>	The BSW UML profile is needed to store additional BSW specific information in the model.
<b>Use Case:</b>	--
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	The latest version of the BSW UML profile is located at:

### 4.1.3 [BSW\_UMLGuide\_00001] Allowed elements

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Allowed elements
<b>Type:</b>	

<b>Importance:</b>	high
<b>Description:</b>	<p>The following elements of UML are allowed to use within the Basic software overall UML model:</p> <ul style="list-style-type: none"> <li>- package</li> <li>- class ('typedef') for classes describing C typedefs</li> <li>- class ('type') for classes describing C types</li> <li>- class ('structure') for classes describing C structs</li> <li>- enumeration for C enums</li> <li>- operation ('function') for functions</li> <li>- operation ('scheduled_function') for scheduled functions</li> <li>- operation ('callback') for callbacks</li> <li>- component ('module') for components describing the interfaces of a BSW module</li> <li>- interface</li> <li>- lifeline</li> <li>- fragment</li> <li>- note</li> <li>- node (with stereotype peripheral or cluster)</li> <li>- state (including initial, final, fork/join, choice, exit)</li> <li>- action</li> <li>- decision</li> <li>- activity</li> <li>- boundary</li> </ul> <p>Other elements shall not be used.</p>
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of the complete communication stack
<b>Dependencies:</b>	<a href="#">[BSW_UMLGuide_00053]</a> Allowed diagrams
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.1.4 [BSW\_UMLGuide\_00002] Allowed relationships

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Allowed relationships
<b>Type:</b>	Changed due to bug #6794
<b>Importance:</b>	high
<b>Description:</b>	<p>The following relationships are allowed to use within the Basic software overall UML model:</p> <ul style="list-style-type: none"> <li>- realize</li> <li>- nesting</li> <li>- dependency ('mandatory') : for mandatory interfaces of a component</li> <li>- dependency ('optional') : for optional interfaces of a component</li> <li>- dependency ('configurable') : for configurable interfaces of a component</li> <li>- message</li> <li>- Self-message</li> <li>- Call</li> <li>- transition</li> <li>- activity edge</li> </ul> <p>Other relationships shall not be used.</p>
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of the complete communication stack
<b>Dependencies:</b>	<a href="#">[BSW_UMLGuide_00053]</a> Allowed diagrams
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.1.5 [BSW\_UMLGuide\_00053] Allowed set of diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Allowed set of diagrams
<b>Type:</b>	Changed according to bug #10724
<b>Importance:</b>	high
<b>Description:</b>	<p>Only a reduced set of diagrams shall be used.</p> <p>Allowed structural diagrams are :</p> <ul style="list-style-type: none"> <li>- Package diagrams and</li> <li>- Component diagrams.</li> </ul> <p>Allowed sequence diagrams are:</p> <ul style="list-style-type: none"> <li>- Sequence diagrams</li> <li>- Activity diagrams and</li> <li>- State machine diagrams.</li> </ul>
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of the complete communication stack
<b>Dependencies:</b>	<a href="#">[BSW_UMLGuide_00001]</a> Allowed elements <a href="#">[BSW_UMLGuide_00002]</a> Allowed relationships
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.1.6 [BSW\_UMLGuide\_00060] Documentation of elements

<b>Initiator:</b>	Technical office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Elements used for generating SWS tables must be documented
<b>Type:</b>	
<b>Importance:</b>	High
<b>Description:</b>	<p>Documentation ('Notes') must be provided for the following element types:</p> <ul style="list-style-type: none"> <li>• class ('type', 'typedef', 'structure')</li> <li>• attributes</li> <li>• operation ('function', 'scheduled_function', 'callback')</li> <li>• enumeration</li> <li>• parameter</li> </ul>
<b>Rationale:</b>	BSW documentation generator needs description for these elements to be able to generate the tables.
<b>Use Case:</b>	
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.1.7 [BSW\_UMLGuide\_00047] Links between diagrams shall be hyperlinks

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	Links between diagrams shall be hyperlinks

<b>Type:</b>	Changed (11.05.2005, WP 1.1.2)
<b>Importance:</b>	High
<b>Description:</b>	If the relationship between two diagrams shall be visualized, then the link shall be modeled as a hyperlink.
<b>Rationale:</b>	Easier navigation between diagrams.
<b>Use Case:</b>	
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

## 4.2 Structural Design

### 4.2.1 [BSW\_UMLGuide\_00054] Use of Packages

<b>Initiator:</b>	Technical Office
<b>Date:</b>	31.07.2006
<b>Short Description:</b>	Use of Packages
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	Packages may be used in three ways: (1) To group (only) sub-packages, (2) to represent one BSW module, grouping the BSW module component and its interfaces or (3) placed below a package of the second type to group additional elements detailing a BSW module. Packages of the first or second type shall only be added by the technical office.

<b>Rationale:</b>	Clear structure of the BSW UML model.
<b>Use Case:</b>	Modeling of the complete software architecture
<b>Dependencies:</b>	<a href="#">[BSW_UMLGuide_00003]</a> Diagrams usage <a href="#">[BSW_UMLGuide_00009]</a> Version numbers of software modules <a href="#">[BSW_UMLGuide_00035]</a> Sub elements of BSW modules
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.2 [BSW\_UMLGuide\_00003] Diagrams usage

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Diagram usage
<b>Type:</b>	Changed according to bug #10724.
<b>Importance:</b>	high
<b>Description:</b>	<p>Each package containing only sub-packages shall at least have one structural "Component" diagram which shows the contents and, if possible, the relationships of the packages which it contains. Each package representing a BSW module shall at least have one structural "Component" diagram which shows at least the "realize", "mandatory", "optional" and "configurable" relationships of the BSW module component which it contains. The name of this diagram shall be equal to the BSW module component name.</p> <p>This diagram shall be placed below the appropriate diagram within the tree view.</p>
<b>Rationale:</b>	Have for each structural element a diagram showing the elements containing it.

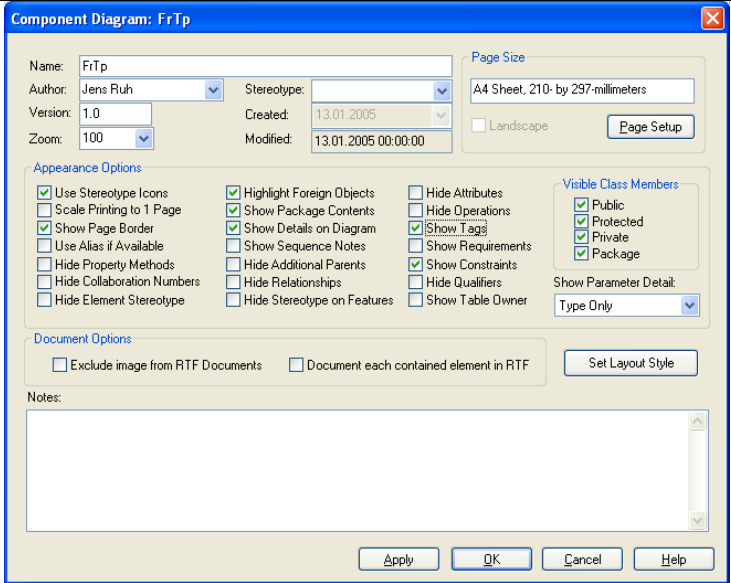
#### 4.2.3 [BSW\_UMLGuide\_00038] Component diagram appearance options

<b>Initiator:</b>	WP1.1.2																
<b>Date:</b>	08.02.2005																
<b>Short Description:</b>	Component diagram appearance options																
<b>Type:</b>	New																
<b>Importance:</b>	high																
<b>Description:</b>	<p>In general only the following appearance options of component diagrams shall be set:</p> <table border="0"> <tr> <td>- Use stereotype icons</td> <td>- yes</td> </tr> <tr> <td>- Scale printing to 1 page</td> <td>- optional</td> </tr> <tr> <td>- Show page border</td> <td>- yes</td> </tr> <tr> <td>- Highlight foreign objects</td> <td>- yes</td> </tr> <tr> <td>- Show package contents</td> <td>- optional</td> </tr> <tr> <td>- Show details on diagram</td> <td>- yes</td> </tr> <tr> <td>- Hide operations</td> <td>- yes</td> </tr> <tr> <td>- Hide attributes</td> <td>- yes</td> </tr> </table> <p>If a user requires more options, he shall ask the technical office for clearance: <a href="mailto:technical.office@autosar.org">technical.office@autosar.org</a>.</p>	- Use stereotype icons	- yes	- Scale printing to 1 page	- optional	- Show page border	- yes	- Highlight foreign objects	- yes	- Show package contents	- optional	- Show details on diagram	- yes	- Hide operations	- yes	- Hide attributes	- yes
- Use stereotype icons	- yes																
- Scale printing to 1 page	- optional																
- Show page border	- yes																
- Highlight foreign objects	- yes																
- Show package contents	- optional																
- Show details on diagram	- yes																
- Hide operations	- yes																
- Hide attributes	- yes																
<b>Rationale:</b>	Harmonization of diagram appearance.																

<b>Use Case:</b>	
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.4 [BSW\_UMLGuide\_00039] Component diagram appearance of BSW module diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Component diagram appearance of BSW module diagrams
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Diagrams placed below the BSW module package shall apply the following changes to the appearance options compared to <a href="#">BSW UMLGuide 00038</a> : - Hide attributes - No - Hide operations - No - Show constraints - yes - Show Tags - yes
<b>Rationale:</b>	Visualizing all attributes, operations and constraints of a BSW module component.

<b>Use Case:</b>	
<b>Dependencies:</b>	<a href="#">BSW_UMLGuide_00038</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.5 [BSW\_UMLGuide\_00004] Header File Modeling

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Header File Modeling
<b>Type:</b>	Changed due to bug #6747 and bug #9820
<b>Importance:</b>	High
<b>Description:</b>	<p>Each header file providing external interfaces of a basic software module shall be modeled as an own interface class (a "BSW Module Interface"). The element representing the basic software module shall have a "realize" relationship to all interface classes specifying its external interfaces. If the realize relation is drawn operations/interfaces of the component representing the BSW module shall neither be overwritten nor implemented (select "Cancel" in "Override Operations/Interfaces" dialog).</p> <p>The names of the interface classes shall follow the following naming convention:</p> <pre>&lt;module prefix&gt;_&lt;name of the interface as specified within the SWS&gt;</pre> <p>&lt;name of the interface as specified within the SWS&gt;: Partly defined by general SRS (e.g. BSW00370)</p> <p>If no interface definition is available the default name is:  <pre>&lt;module prefix&gt;_MissingSWS</pre>                 When the interface specification is available the UML element representing it shall only be RENAMED (<u>not deleted</u>).</p> <p>If an interface specification is available, but the file structure has not been refined the name of the interface is:  <pre>&lt;module prefix&gt;</pre> </p>

	<p>Note: UML 2 does not require showing the explicit &lt;&lt;realize&gt;&gt; stereotype. But because the Enterprise architect automatically adds the stereotype this should be done in a harmonized way throughout the model.</p>
<b>Rationale:</b>	Enabling the showing of "include" relationships between modules.
<b>Use Case:</b>	Modeling of the sequences of API calls during a Lin frame transmission
	<div style="border: 1px solid black; padding: 5px;"> <p><b>id MemHwADependencies</b></p> <p>Name: MemHwADependencies Package: MemHwA Version: 1.0 Author: Jens Zimmermann</p> <pre> classDiagram     package MemHwA {         package Fee::Fee {             Fee             Fee_Cbk         }         package Ea::Ea {             Ea             Ea_Cbk         }         package Memif {             Memif         }         package Memif_Types {             Fee             Ea         }         package Nvm_Cbk {             Nvm_Cbk         }         package FIs {             FIs         }         package Eep {             Eep         }     }     Fee::Fee --&gt; Fee : realize     Fee::Fee --&gt; Fee_Cbk : realize     Ea::Ea --&gt; Ea : realize     Ea::Ea --&gt; Ea_Cbk : realize     Memif --&gt; Memif : realize     Memif --&gt; Memif_Types : realize     Memif_Types --&gt; Fee : realize     Memif_Types --&gt; Ea : realize     Nvm_Cbk --&gt; Fee : use     Nvm_Cbk --&gt; Ea : use     Fee_Cbk --&gt; FIs : use     Ea_Cbk --&gt; Eep : use             </pre> </div>
<b>Dependencies:</b>	BSW00370, BSW00346, BSW00353, BSW00361
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

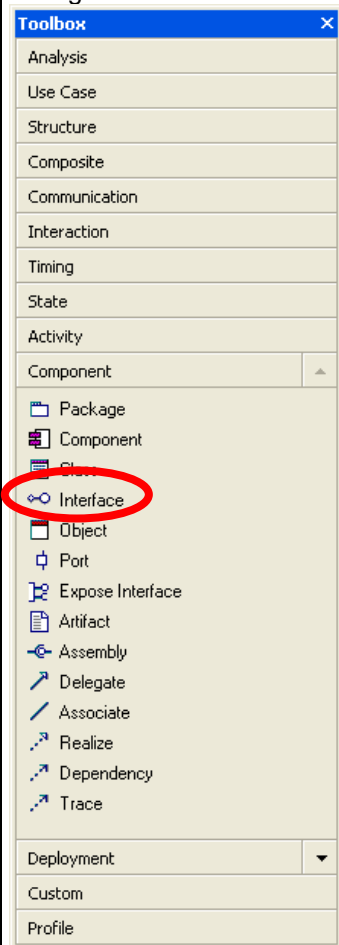
#### 4.2.6 [BSW\_UMLGuide\_00005] Basic Software Module Modeling

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Basic Software Module Modeling
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Each basic software module source code file(s) shall be modeled as an UML package containing one component with stereotype "module" (the "BSW Module Component") and the appropriate interfaces (the "BSW Module Interfaces"). The name of the package and component shall be the Prefix of the basic software module (specified within the basic software list).
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of ECU State Manager. Name of this component: EcuM



<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.7 [BSW\_UMLGuide\_00052] Interface creation

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.11.2005
<b>Short Description:</b>	Interface creation
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	<p>Interfaces shall only be created by dragging "Interface" from the toolbox into a diagram.</p> 
<b>Rationale:</b>	<p>There are two ways in EA to create an interface class:</p> <p>(a) Create a regular class and afterwards add the stereotype &lt;&lt;interface&gt;&gt;.</p> <p>(b) Directly drag a new interface into a diagram (e.g. from the toolbox window).</p> <p>Only (b) leads to a real interface in the EA sense:</p> <ul style="list-style-type: none"> <li>o correct icon in the project view</li> <li>o class and all operations are enforced to be abstract</li> </ul>
<b>Use Case:</b>	Modeling of the complete communication stack
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--

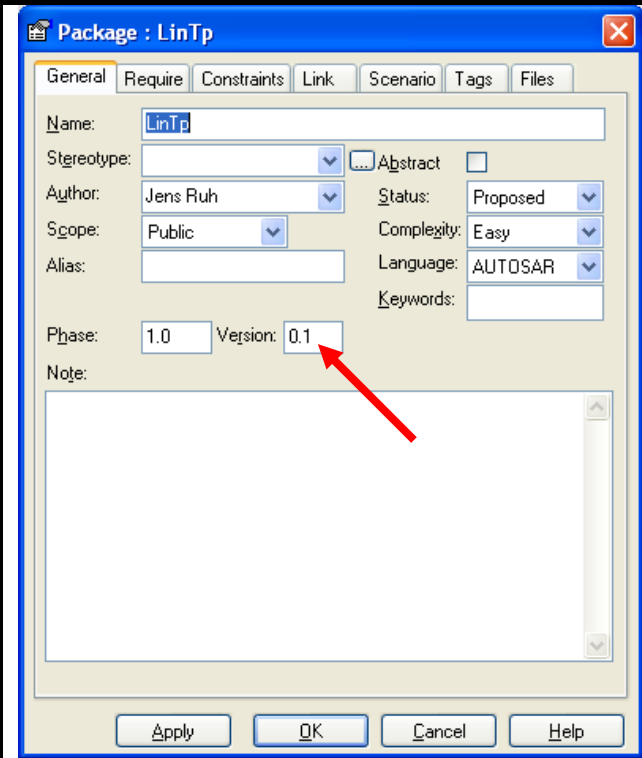
<b>Supporting Material:</b>	--
-----------------------------	----

#### 4.2.8 [BSW\_UMLGuide\_00006] Interface Modeling

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Interface Modeling
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Each external interface class shall be modeled in "circle notation" within the "component diagram" of a package containing the basic software module components and/or classes.
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of ECU State manager.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.9 [BSW\_UMLGuide\_00009] Version numbers of software modules

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Version numbers of software modules
<b>Type:</b>	Changed according to bug #10724
<b>Importance:</b>	High
<b>Description:</b>	For each component which has the 'module' stereotype (represents a BSW module) the version must be specified according to the modules version number.
<b>Rationale:</b>	Model everything in the same style so that it is easier to understand
<b>Use Case:</b>	Modeling of FIs Interfaces:

	
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.10 [BSW\_UMLGuide\_00010] Component Definition

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	08.10.2004
<b>Short Description:</b>	Component Definition
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Each "BSW Module Component" in the cluster diagram shall be marked as "Composite Element".
<b>Rationale:</b>	Easier navigation within the model
<b>Use Case:</b>	Look into the details of one specific component.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.11 [BSW\_UMLGuide\_00011] Accessing interfaces of other components

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Accessing interfaces of other components
<b>Type:</b>	Changed due to bug #6779.
<b>Importance:</b>	high
<b>Description:</b>	If a basic software module requires the access of another module this relation shall be modeled as a "mandatory" dependency between the

	component of the basic software module requiring the access and the appropriate Interface class of the other basic software module. If the interface to be accessed is not in the same package a link to the interface shall be copied into the appropriate diagram. The link shall be modeled in circle notation.
<b>Rationale:</b>	Restriction of modeling techniques
<b>Use Case:</b>	Eep Interface "uses" interface of Eep driver.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.12 [BSW\_UMLGuide\_00055] Use of parameter kind

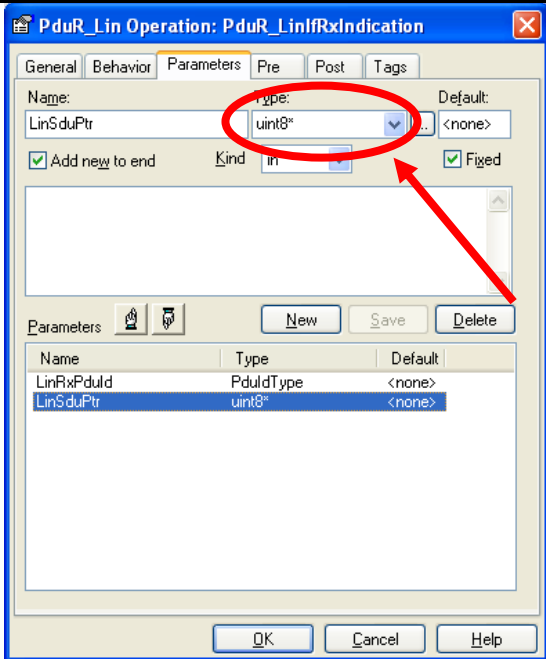
<b>Initiator:</b>	Technical Office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Use of parameter kind
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	The drop down list for the 'kind' of function parameters should be set to the appropriate value (in, out, in/out, return). In the case that EA adds a "*" to the parameter type (for 'out') although it is already a pointer (by a typedef), this should be annotated in the field 'notes' of the respective function.
<b>Rationale:</b>	Use the in/out feature and work around the EA bug.
<b>Use Case:</b>	Modeling of the com stack types
<b>Dependencies:</b>	<a href="#">[BSW_UMLGuide_00037]</a> Definition of pointer types
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.13 [BSW\_UMLGuide\_00061] Definition of return type

<b>Initiator:</b>	Technical Office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Definition of return type
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	The return type of a function must not be specified. Instead a parameter with kind 'return' must be used.
<b>Rationale:</b>	Return parameters must be documented. This is not possible when just the return type is specified.
<b>Use Case:</b>	
<b>Dependencies:</b>	
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.14 [BSW\_UMLGuide\_00037] Definition of pointer types

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Definition of pointer types

<b>Type:</b>	new									
<b>Importance:</b>	high									
<b>Description:</b>	If a parameter or a return value of a module interface represents a pointer the asterix(es) shall be placed directly after the original type.									
<b>Rationale:</b>	Readability of the module (specific views will otherwise filter out that information). Harmonization of modeling techniques.									
<b>Use Case:</b>	 <table border="1" data-bbox="542 795 1037 1086"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>LinRxPduId</td> <td>PduIdType</td> <td>&lt;none&gt;</td> </tr> <tr> <td>LinSduPtr</td> <td>uint8*</td> <td>&lt;none&gt;</td> </tr> </tbody> </table>	Name	Type	Default	LinRxPduId	PduIdType	<none>	LinSduPtr	uint8*	<none>
Name	Type	Default								
LinRxPduId	PduIdType	<none>								
LinSduPtr	uint8*	<none>								
<b>Dependencies:</b>	--									
<b>Conflicts:</b>	--									
<b>Supporting Material:</b>	--									

#### 4.2.15 [BSW\_UMLGuide\_00027] Definition of structures

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Definition of structures
<b>Type:</b>	
<b>Importance:</b>	high
<b>Description:</b>	Each type definition which represents a structure declaration shall be modeled as a class with the stereotype 'structure'. All possible entries shall be defined as attributes of that class. The attributes shall have the scope "public". The ordering of the attributes shall be the same as expected in the generated table. For each attribute the appropriate type shall be specified:

<b>Rationale:</b>	All types have to be defined in the same way, so that there are no inconsistencies within the model.
<b>Use Case:</b>	The example shown in the description represents the following Std_Types enumeration: <pre>typedef struct PduInfoType {     uint8* SduDataPtr,     uint16 Length };</pre>
<b>Dependencies:</b>	Definition of simple types <a href="#">[BSW_UMLGuide_00028]</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

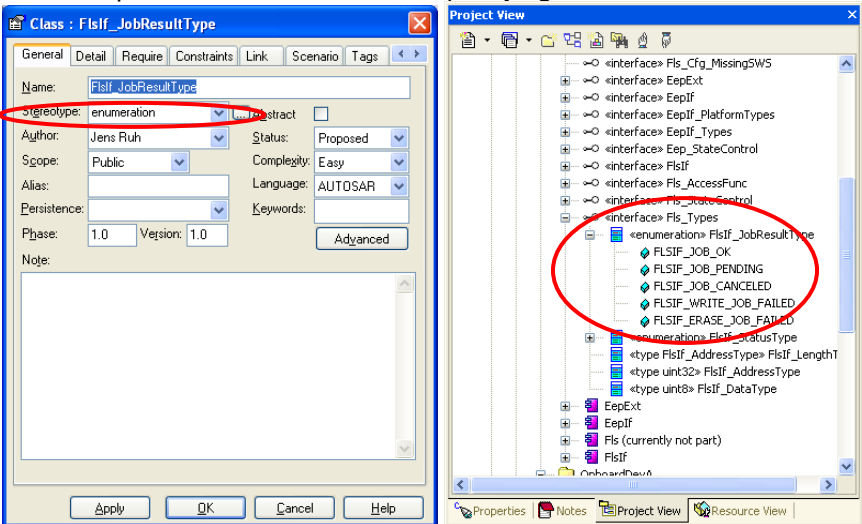
#### 4.2.16 [BSW\_UMLGuide\_00025] 'Language' definition of Components

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	26.11.2004
<b>Short Description:</b>	'Language' definition of Components
<b>Type:</b>	Changed (supporting material added)
<b>Importance:</b>	high
<b>Description:</b>	The 'Language' Attribute of at least each Component which represents a Software module shall be set to AUTOSAR.

<b>Rationale:</b>	The AUTOSAR language defines a set of basic types which shall be used within AUTOSAR.
<b>Use Case:</b>	Using AUTOSAR type uint8 as return value.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	<p>If you set in the options menu AUTOSAR to the default language the correct language will be set automatically: Tools -&gt; Options -&gt;</p>

#### 4.2.17 [BSW\_UMLGuide\_00026] Definition of enumerations

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	26.11.2004
<b>Short Description:</b>	Definition of enumerations
<b>Type:</b>	Changed due to bug #6711
<b>Importance:</b>	high

<p><b>Description:</b></p>	<p>Each type definition representing an enumeration shall be modeled as a n UML enumeration.</p> <p>All possible entries have to be defined as attributes of this class. The order of the attributes from top to bottom shall represent the order of the enumeration specified.</p> <p>The Attributes shall have no type and the 'scope' has to be set to 'public'. It shall be placed below the interface specifying it.</p>  <p>For attribute names, the AUTOSAR-conform attribute values shall be used. If for this attribute value a code number exist, this code number shall be placed in the "initial" value entry of a type's detailed properties form in EA.</p>
<p><b>Rationale:</b></p>	<p>All types have to be defined in the same way, so that there are no inconsistencies within the model.</p>
<p><b>Use Case:</b></p>	<p>The example shown in the description represents the following Std_Types enumeration:</p> <pre>typedef enum FlsIf_JobResultType {     FLSIF_JOB_OK,     FLSIF_JOB_PENDING,     ... };</pre> <p>Need to represent DCM's Neg Resp Codes (NRCs), for which an enum type is defined in SWS DCM v1.1.6, section 8.1.2.8. Place the NRC "DEM_E_xxx" in [attribute]"name" and the 0x value in "initial" value.</p>
<p><b>Dependencies:</b></p>	<p>Definition of simple types <a href="#">[BSW_UMLGuide_00028]</a></p>
<p><b>Conflicts:</b></p>	<p>--</p>
<p><b>Supporting Material:</b></p>	<p>--</p>

#### 4.2.18 [BSW\_UMLGuide\_00028] Definition of simple types

<p><b>Initiator:</b></p>	<p>WP1.1.2</p>
<p><b>Date:</b></p>	<p>27.04.2007</p>
<p><b>Short Description:</b></p>	<p>Definition of simple types</p>
<p><b>Type:</b></p>	<p></p>
<p><b>Importance:</b></p>	<p>high</p>
<p><b>Description:</b></p>	<p>Each type definition shall be modeled as a separate class with stereotype 'type'. If the type has a hardware and configuration independent type (e.g. 'unit8'), the tagged value 'range' has to specify the range as text.</p>



<b>Rationale:</b>	All types have to be defined in the same way, so that there are no inconsistencies within the model.
<b>Use Case:</b>	--
<b>Dependencies:</b>	Definition of structures <a href="#">[BSW_UMLGuide_00028]</a> , Definition of enumerations <a href="#">[BSW_UMLGuide_00026]</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	

#### 4.2.19 [BSW\_UMLGuide\_00059] Definition of typedefs

<b>Initiator:</b>	Technical Office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Definition of typedefs
<b>Type:</b>	
<b>Importance:</b>	high
<b>Description:</b>	Each type definition shall be modeled as a separate class with stereotype 'typedef'. If the typedef is independent from the configuration, the typedef must be a specialization of the type it is referring to. If the typedef can refer to different types depending on the configuration, the typedef must be a specialization of those types.
<b>Rationale:</b>	The documentation generator has extract informations regarding the typedefs from the UML model.
<b>Use Case:</b>	--
<b>Dependencies:</b>	Definition of structures <a href="#">[BSW_UMLGuide_00028]</a> , Definition of enumerations <a href="#">[BSW_UMLGuide_00026]</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.20 [BSW\_UMLGuide\_00066] Defintion of ranges for typedefs

<b>Initiator:</b>	Technical Office
<b>Date:</b>	23.07.2007
<b>Short Description:</b>	Definition of ranges for typedefs
<b>Type:</b>	
<b>Importance:</b>	high
<b>Description:</b>	If a typedef (class with stereotype <<typedef>>) has a restricted set of ranges, an attribute with stereotype <<range>> has to be created for each such range. The name of the attribute specifies the range label and the notes field describes the range.
<b>Rationale:</b>	The documentation generator has to extract information regarding the range of typedefs from the UML model.
<b>Use Case:</b>	See «typedef» Fim_FunctionIdType.
<b>Dependencies:</b>	Definition of structures <a href="#">[BSW_UMLGuide_00028]</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.21 [BSW\_UMLGuide\_00062] Definition of functions and callbacks

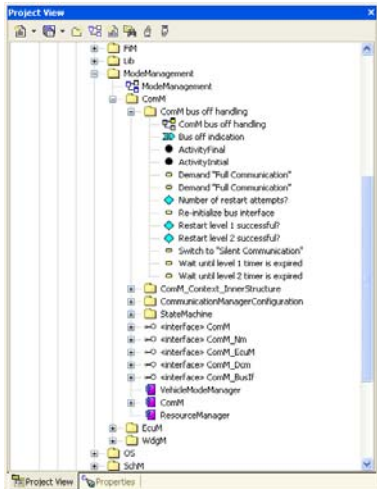
<b>Initiator:</b>	Technical Office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Definition of functions
<b>Type:</b>	
<b>Importance:</b>	high
<b>Description:</b>	<p>Each BSW function must be represented by an operation with stereotype 'function'.</p> <p>Each BSW callback must be represented by an operation with stereotype 'callback'.</p> <p>The following tagged values must be specified:</p> <ul style="list-style-type: none"> <li>• ServiceID: numeric id of the function</li> <li>• Synchronous: must be set to 'true' if this function is synchronous, must be set to 'false' otherwise</li> <li>• Reentrancy: description of the reentrancy for this function. Should be set to "Reentrant" for reentrant functions. Should be set to "Non-Reentrant" for non-reentrant functions.</li> </ul>
<b>Rationale:</b>	The documentation generator needs this information to generate the function tables.
<b>Use Case:</b>	--
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.22 [BSW\_UMLGuide\_00063] Definition of scheduled functions

<b>Initiator:</b>	Technical Office
<b>Date:</b>	27.04.2007
<b>Short Description:</b>	Definition of functions
<b>Type:</b>	
<b>Importance:</b>	High
<b>Description:</b>	<p>Each BSW scheduled function must be represented by an operation with stereotype 'scheduled function'. In addition to the tagged values for functions, the following tagged values must be specified:</p> <ul style="list-style-type: none"> <li>• schedule: Must be set to one of the following values                             <ul style="list-style-type: none"> <li>○ FIXED_CYCLIC</li> <li>○ ON_PRE_CONDITION</li> <li>○ VARIABLE_CYCLIC</li> <li>○ VARIABLE_CYCLIC_OR_ON_PRE_CONDITION</li> </ul> </li> </ul>
<b>Rationale:</b>	The documentation generator needs this information to generate the function tables.
<b>Use Case:</b>	--
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.2.23 [BSW\_UMLGuide\_00035] Sub elements of BSW modules

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Sub elements of BSW modules
<b>Type:</b>	Changed according to bug #10724

<b>Importance:</b>	high
<b>Description:</b>	<p>The internal behaviour of BSW modules may be modeled in two ways:</p> <p>(1) A package may be added to the package representing the BSW module or</p> <p>(2) elements can be placed below the BSW module component.</p> <p>In case of optional functionality or scaled functionality the respective dependency in the respective diagram shall be commented by selecting 'Attach Note or Constraint'.</p>
<b>Rationale:</b>	<p>The Sub elements of a component could be of type:</p> <ul style="list-style-type: none"> <li>- separation of concerns / grouping of functionality</li> <li>- optionally / scalability</li> </ul>
<b>Use Case:</b>	<p>Refinement of ComM.</p> 
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

## 4.3 Behavioral Design

### 4.3.1 General

#### 4.3.1.1 [BSW\_UMLGuide\_00030] Usage of Sequence Diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Usage of Sequence Diagrams
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Only sequence diagrams shall be used for modeling interactions of different modules.
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of the sequences of API calls during a LIN frame transmission.
<b>Dependencies:</b>	<a href="#">BSW_UMLGuide_00007</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.3.1.2 [BSW\_UMLGuide\_00031] Usage of State Machine Diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Usage of State Machine Diagrams
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Only state machine diagrams shall be used for modeling state dependencies within and in between elements.
<b>Rationale:</b>	Restriction of different modeling techniques.
<b>Use Case:</b>	Modeling of ECU Manager state changes.
<b>Dependencies:</b>	<a href="#">BSW_UMLGuide_00007</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

### 4.3.2 Sequence Diagrams

#### 4.3.2.1 [BSW\_UMLGuide\_00012] Location of Sequence Diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Location of Sequence Diagrams
<b>Type:</b>	Renamed due to bug #6749.
<b>Importance:</b>	High
<b>Description:</b>	All sequence diagrams have to be placed within the "Interaction View Package"
<b>Rationale:</b>	Definition of similar model structures
<b>Use Case:</b>	Modeling of the AUTOSAR COM stack
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.3.2.2 [BSW\_UMLGuide\_00020] Packages to contain sequence diagrams

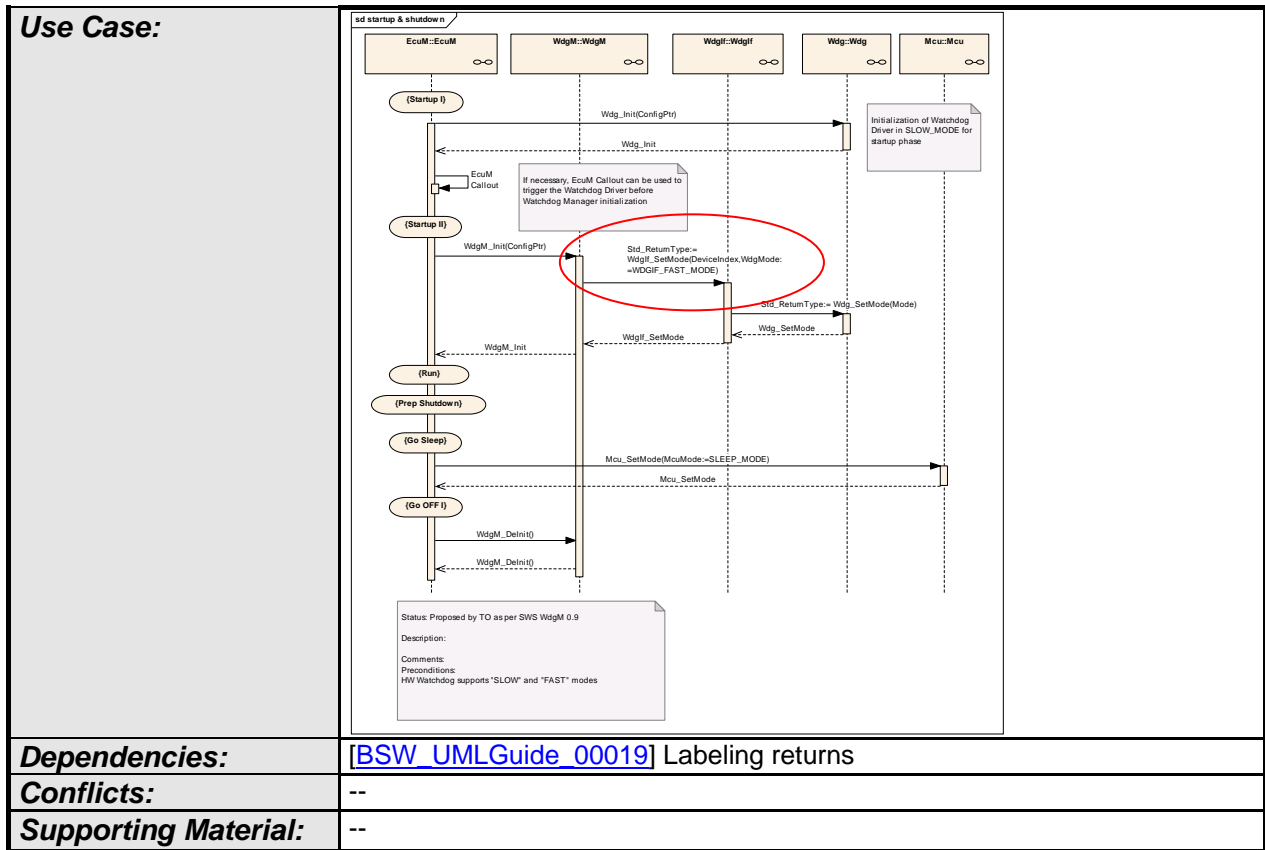
<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	25.01.2005
<b>Short Description:</b>	Packages to contain sequence diagrams
<b>Type:</b>	Changed according to bug #6790
<b>Importance:</b>	High
<b>Description:</b>	A new sequence diagram shall be put into an appropriate package. If no such package is available, it shall be requested from the technical office: <a href="mailto:technical.office@autosar.org">technical.office@autosar.org</a> .
<b>Rationale:</b>	Guarantee of readability and correct placement of the sequence tree.
<b>Use Case:</b>	--
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.3.2.3 [BSW\_UMLGuide\_00021] Commenting of Sequence Diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	19.10.2004
<b>Short Description:</b>	Commenting of Sequence Diagrams
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	<p>Each Sequence diagram shall have a comment placed as 'note' within the diagram that contains the following items:</p> <ul style="list-style-type: none"> <li>• Status (open – proposed – approved – conflict – rejected)</li> <li>• Description</li> <li>• Comment</li> </ul> <p>If a sequence diagram is rejected or on conflict, the reason shall be described within the comment.</p>
<b>Rationale:</b>	<p>Give other people the chance to understand.</p> <p>Traceability</p>
<b>Use Case:</b>	<p>Status: approved</p> <p>Description: A CAN frame is received and indicated to the upper layer in interrupt context.</p> <p>Comment: -none-</p>
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

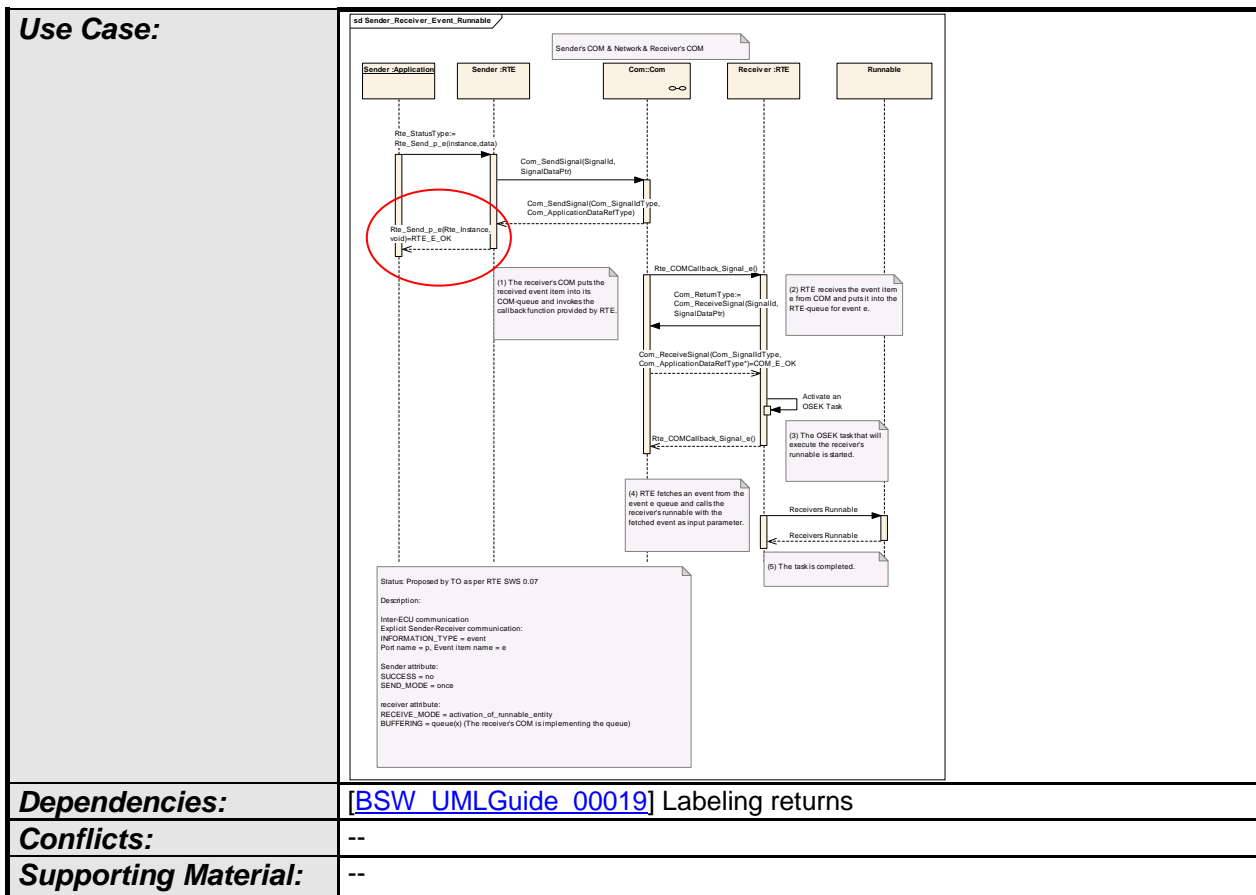
#### 4.3.3 [BSW\_UMLGuide\_00057] Parameter values in sequence diagrams

<b>Initiator:</b>	Technical Office
<b>Date:</b>	31.07.2006
<b>Short Description:</b>	Parameter values in sequence diagrams
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	<p>If a function is called with a fixed value for one or more of its parameters in a sequence diagram, then this should be modeled by writing 'ParName:=value' in the field 'Parameters' of the respective message.</p>
<b>Rationale:</b>	Unified message modeling.

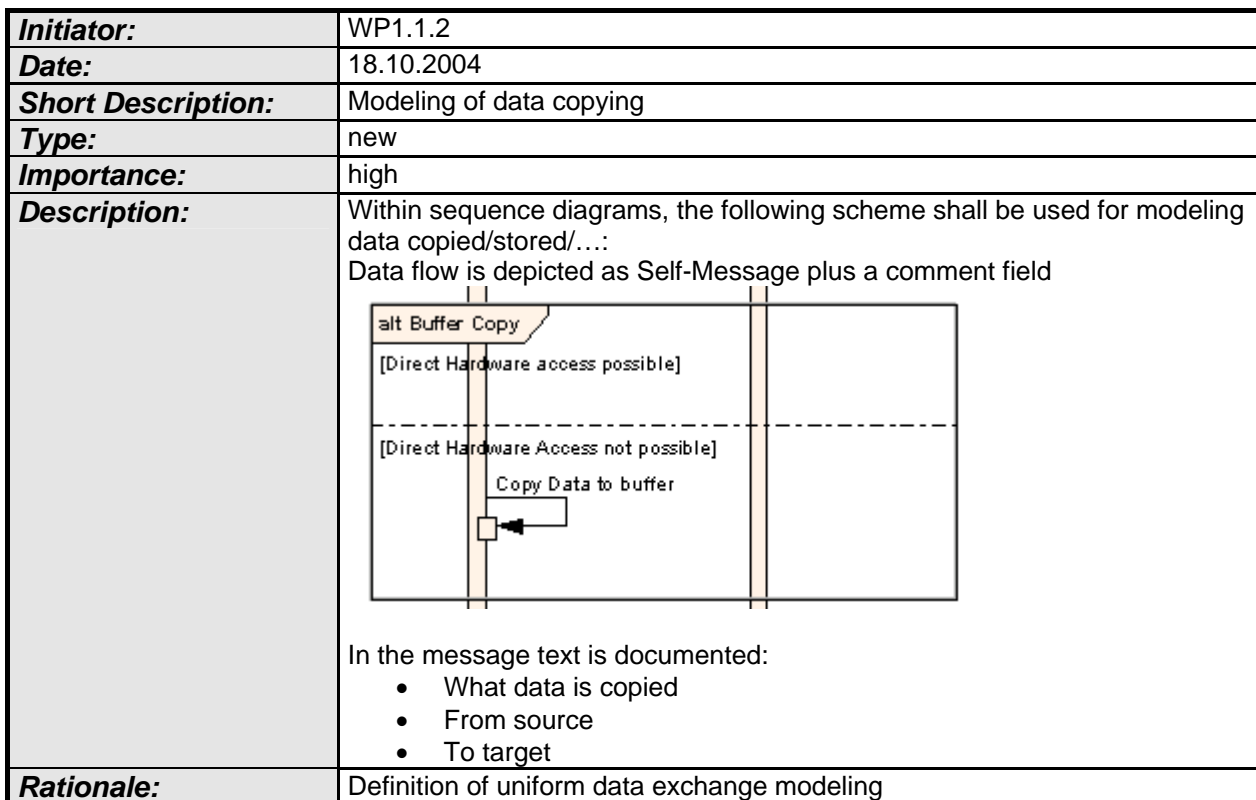


#### 4.3.3.1 [BSW\_UMLGuide\_00058] Return values in sequence diagrams

<b>Initiator:</b>	Technical Office
<b>Date:</b>	31.07.2006
<b>Short Description:</b>	Return values in sequence diagrams
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	If the return of a function should be shown to give a specific value, then this should be modeled by writing 'FuncName=value' in the field 'Message' of the respective return-message.
<b>Rationale:</b>	Unified message modeling.

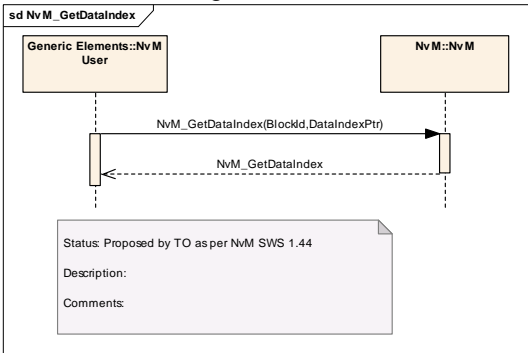


**4.3.3.2 [BSW\_UMLGuide\_00018] Modeling of data copying**



<b>Use Case:</b>	Modeling of data exchange between buffers of different layers
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.3.3.3 [BSW\_UMLGuide\_00019] Labeling returns

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	07.10.2004
<b>Short Description:</b>	Labeling returns
<b>Type:</b>	Changed according to bug #10724
<b>Importance:</b>	High
<b>Description:</b>	<p>Returns shall be labeled. The label shall be the name of the function it belongs to without parameter names or -types. Returns shall be marked as "Is Return".</p> <p>It is announced that in future return labels can be automatically be selected within the Enterprise architect. If this is possible no hand-made adaptations shall be done after selection.</p>
<b>Rationale:</b>	Allow easy identification to which function a return belongs
<b>Use Case:</b>	<p>A UML message has the name "Transmit CAN PDU". The return message has the same name.</p> 
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.3.3.4 [BSW\_UMLGuide\_00036] Linking sequence diagrams

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	13.01.2005
<b>Short Description:</b>	Linking sequence diagrams
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	<p>Each package within the 'Interaction Views' package shall contain dedicated diagrams containing descriptions and links to sub diagrams.</p> <p>These diagrams will be generated by the model owner. The author of a sequence diagram shall therefore provide a short description of contents of generated packages and diagrams to the model owner.</p>
<b>Rationale:</b>	Readability of overall module
<b>Use Case:</b>	--



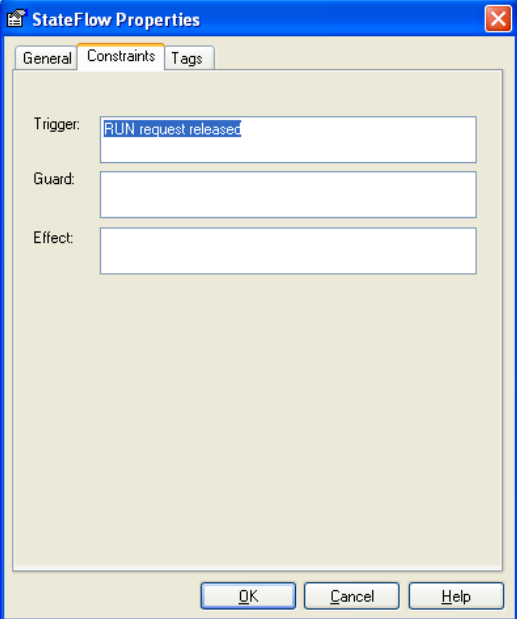
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

### 4.3.4 State Machine Diagrams

#### 4.3.4.1 [BSW\_UMLGuide\_00041] States shall have thick lines

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	States shall have thick lines
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	The state boxes shall have an outline of 2 points
<b>Rationale:</b>	Allow easier distinction between states and activities
<b>Use Case:</b>	<p>(to see what is meant please zoom to 200%)</p>
<b>Dependencies:</b>	<a href="#">BSW_UMLGuide_00048</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

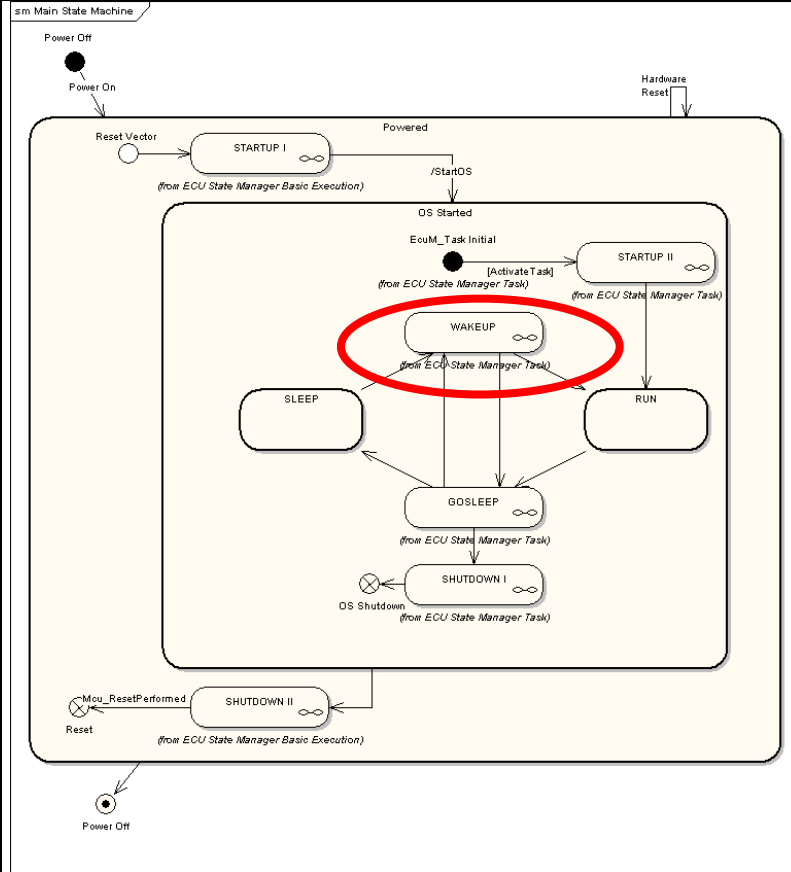
**4.3.4.2 [BSW\_UMLGuide\_00042] A trigger condition shall be defined for each transition**

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	A trigger condition shall be defined for each transition
<b>Type:</b>	Changed due to bug #6794
<b>Importance:</b>	High
<b>Description:</b>	In each transition between two states the trigger of the transition (the condition to make a transition) shall be defined in the 'Trigger' field of the 'State Flow Properties'.
<b>Rationale:</b>	Necessary for complete behavioral description
<b>Use Case:</b>	
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

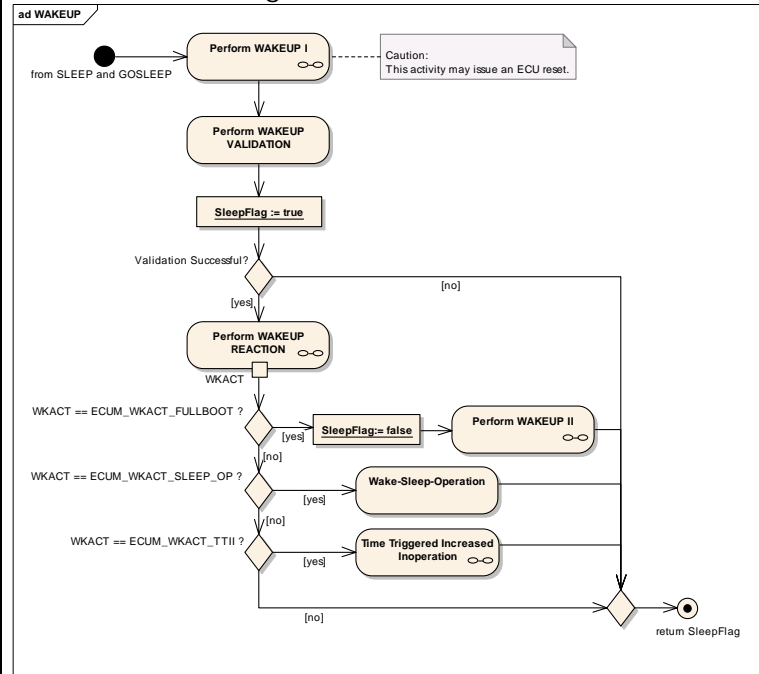
**4.3.4.3 [BSW\_UMLGuide\_00043] Transitions may be modeled with sub-activities**

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	Transitions may be modeled with sub-activities
<b>Type:</b>	Changed due to bug #6794
<b>Importance:</b>	High
<b>Description:</b>	To reduce complexity of diagrams Activities may be modeled in a hierarchical way by using sub-activities.
<b>Rationale:</b>	In some cases complex activities could trigger a transition. In these cases the diagrams become rather complex. These sub-activities may be visualized as sub activities to reduce complexity in one diagram.

**Use Case:**



**Overview State diagram.**



**Refined activity 'WAKEUP'.**

**Dependencies:**

--

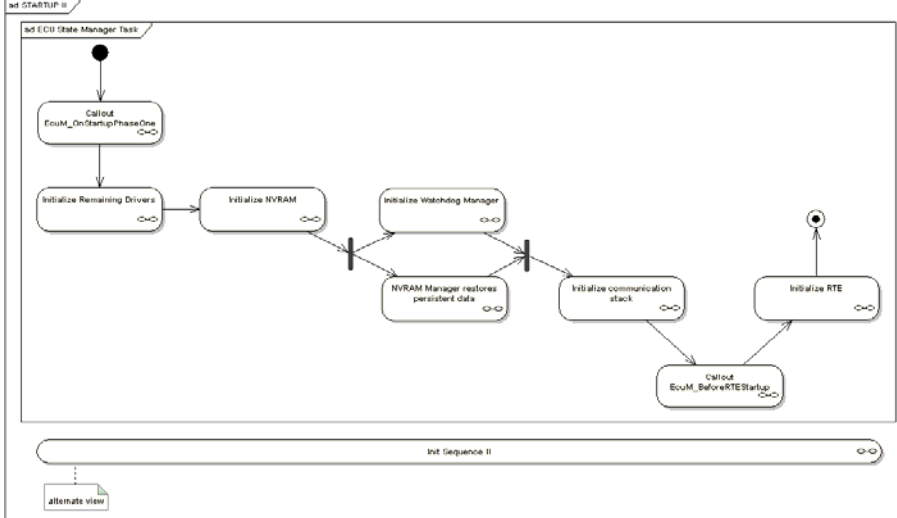
**Conflicts:**

--

**Supporting Material:**

--

#### 4.3.4.4 [BSW\_UMLGuide\_00046] Links to parent diagrams shall be drawn as hyperlink diagram references

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	Links to parent diagrams shall be drawn as diagram references
<b>Type:</b>	Changed due to bug #7379
<b>Importance:</b>	medium
<b>Description:</b>	 <p>The shown diagram (which is only an example) shows an activity diagram of a sub-activity of a larger system. The frame around the diagram indicates the link to the parent diagram. Applies to state and activity diagrams similarly.</p>
<b>Rationale:</b>	Simple forward/backward navigation
<b>Use Case:</b>	--
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	Diagram references do not work in the HTML-Report!

### 4.3.5 Activity Diagrams

#### 4.3.5.1 [BSW\_UMLGuide\_00048] Activities shall have thin lines

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	Activities shall have thin lines
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	The activity boxes shall have an outline of 1 point
<b>Rationale:</b>	Allow easier distinction between states and activities
<b>Use Case:</b>	--
<b>Dependencies:</b>	<a href="#">BSW_UMLGuide_00041</a>
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

### 4.3.5.2 [BSW\_UMLGuide\_00049] Conditions to be defined for each branch

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	Conditions to be defined for each branch
<b>Type:</b>	Changed according to bug #6795
<b>Importance:</b>	High
<b>Description:</b>	<p>If a flow branches because of a condition, the 'Decision' element shall be used. All outgoing control flows must have set a 'Guard' constraint in 'Control Flow Properties'.</p> <p>If different flows shall be merged, also the "Decision" element shall be used.</p>
<b>Rationale:</b>	--
<b>Use Case:</b>	<p>The diagram shows the flow of an activity named 'ad WAKEUP'. It starts with an initial node leading to 'Perform WAKEUP I', which has a note: 'Caution: This activity may issue an ECU reset.'. This is followed by 'Perform WAKEUP VALIDATION' and 'SleepFlag := true'. A decision diamond asks 'Validation Successful?'. If 'yes', it goes to 'Perform WAKEUP REACTION'. If 'no', it bypasses the reaction and goes to a final merge diamond. 'Perform WAKEUP REACTION' leads to a join node 'WKACT'. From 'WKACT', three decision diamonds branch out based on 'WKACT == ECUM_WKACT_FULLBOOT?', 'WKACT == ECUM_WKACT_SLEEP_OP?', and 'WKACT == ECUM_WKACT_TTII?'. The first decision has a 'yes' path to 'SleepFlag := false' and a 'no' path to the final merge diamond. The second decision has a 'yes' path to 'Wake-Sleep-Operation' and a 'no' path to the final merge diamond. The third decision has a 'yes' path to 'Time Triggered Increased Inoperation' and a 'no' path to the final merge diamond. All paths eventually merge at a diamond labeled 'return SleepFlag'.</p>
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	Similar rules apply for forks/joins. The idea is to have the control flow clearly defined. Object flow follows slightly different rules. Such as strict rules make UML clumsy for use with object flow elements.

### 4.3.5.3 [BSW\_UMLGuide\_00050] Activities to be re-used in sequence diagrams should also be drawn as sequence diagrams

<b>Initiator:</b>	4.2.2.1.9 (CZ)
<b>Date:</b>	14.02.2005
<b>Short Description:</b>	Activities to be re-used in sequence diagrams must also be drawn as sequence diagrams
<b>Type:</b>	new
<b>Importance:</b>	medium
<b>Description:</b>	If an activity is to be referenced within a sequence diagram the drawing messages will not look nice. Therefore this type of activities should also be modeled as sequence diagrams.
<b>Rationale:</b>	Nice modeling.

<b>Use Case:</b>	--
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

## 4.4 Model synchronization

All guidelines related to the design master are not intended for 'normal' users, because the master of the BSW UML model will not be distributed. This chapter will be refined as soon as the process of collaborative work on the BSW UML model has been agreed.

### 4.4.1 [BSW\_UMLGuide\_00013] Creating a Design Master

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	14.10.2004
<b>Short Description:</b>	Creating a Design Master
<b>Type:</b>	Changed according to bug #6796
<b>Importance:</b>	High
<b>Description:</b>	Convert the base project to a Design Master using the Make Design Master option in the Tools ( Manage .EAP File submenu).
<b>Rationale:</b>	--
<b>Use Case:</b>	This shall be done once for the project by AUTOSAR Technical Office only (already done – no more actions required).
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

### 4.4.2 [BSW\_UMLGuide\_00023] Design Master naming convention

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	14.10.2004
<b>Short Description:</b>	Design Master naming convention
<b>Type:</b>	New
<b>Importance:</b>	High
<b>Description:</b>	The file name of the Design Master shall have the following naming: Master_AR_BasicSWArchitecture.eap and to be managed by a version management tool.
<b>Rationale:</b>	--
<b>Use Case:</b>	Master_AR_BasicSWArchitecture.eap
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.4.3 [BSW\_UMLGuide\_00014] Creating replicas from the Design Master

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	14.10.2004
<b>Short Description:</b>	Creating replicas from the Design Master
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Create replicas from the design master using the Create New Replica option in the Tools ( Manage .EAP File submenu. Further work must be done on the replica.
<b>Rationale:</b>	--
<b>Use Case:</b>	To create your own local working model. This has to be done only once.
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

#### 4.4.4 [BSW\_UMLGuide\_00022] Replica naming convention

<b>Initiator:</b>	WP1.1.2
<b>Date:</b>	14.10.2004
<b>Short Description:</b>	Replica naming convention
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	Replicas of the design master shall have the following naming convention: Replica_AR_BasicSWArchitecture_<Version number>_<Author short name>.eap
<b>Rationale:</b>	--
<b>Use Case:</b>	Replica_AR_BasicSWArchitecture_V24.1_CMA.eap
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	--

### 4.5 Documentation generation

#### 4.5.1 [BSW\_UMLGuide\_00067] Providing an alternative name for generated tables

<b>Initiator:</b>	TO
<b>Date:</b>	05.09.2007
<b>Short Description:</b>	Providing an alternative name for generated tables
<b>Type:</b>	new
<b>Importance:</b>	high
<b>Description:</b>	The documentation generator for the API tables uses the element name as table name for the inclusion in the SWS Word document. This name has a limited length in Word, which some elements in the BSW UML model exceed. An alternative shorter name can be added by editing the tagged value "aName".
<b>Rationale:</b>	Limitation on the length of anchors in Word.
<b>Use Case:</b>	--

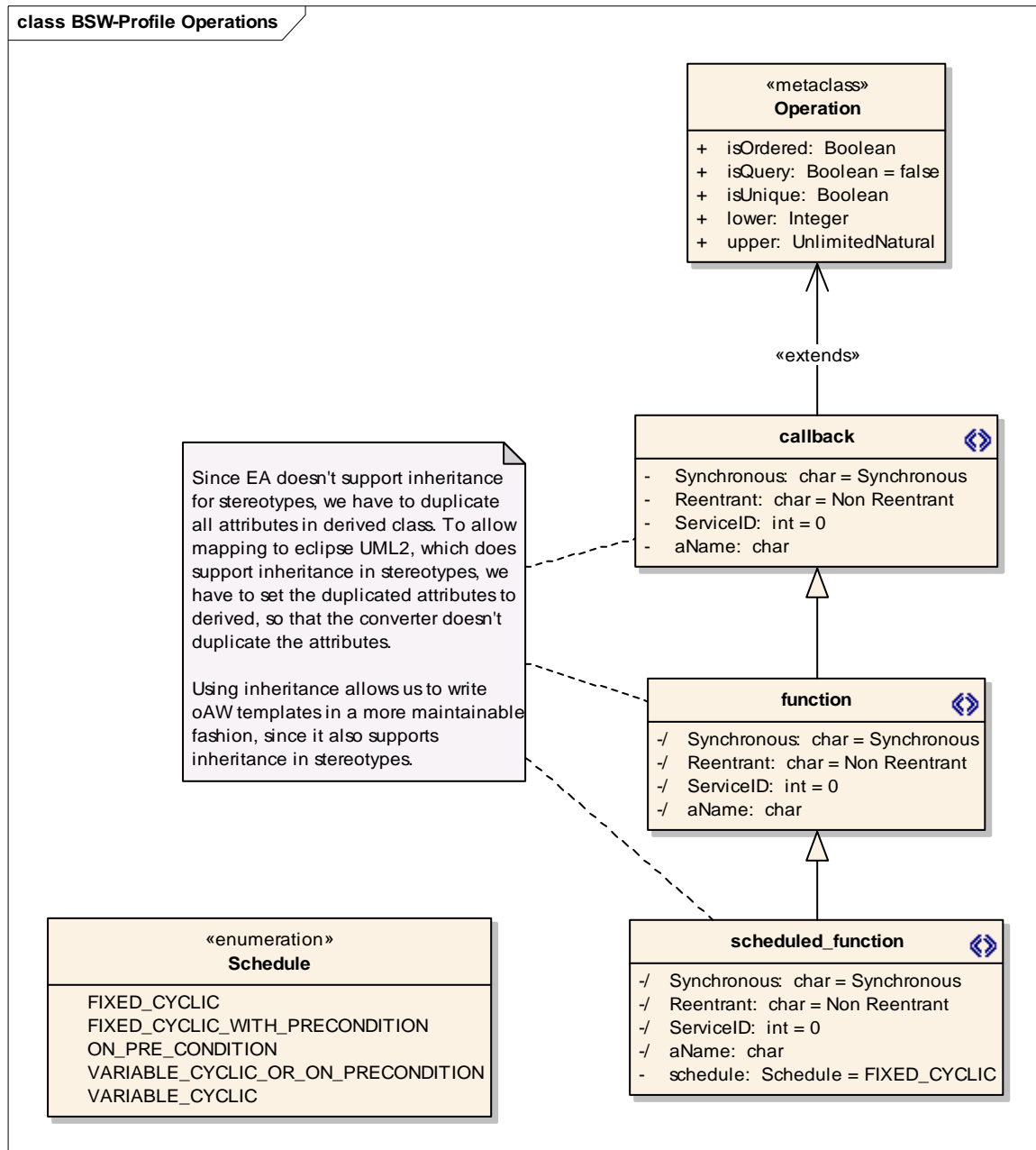
<b>Dependencies:</b>	--
<b>Conflicts:</b>	--
<b>Supporting Material:</b>	AUTOSAR_SWS_DEM.doc



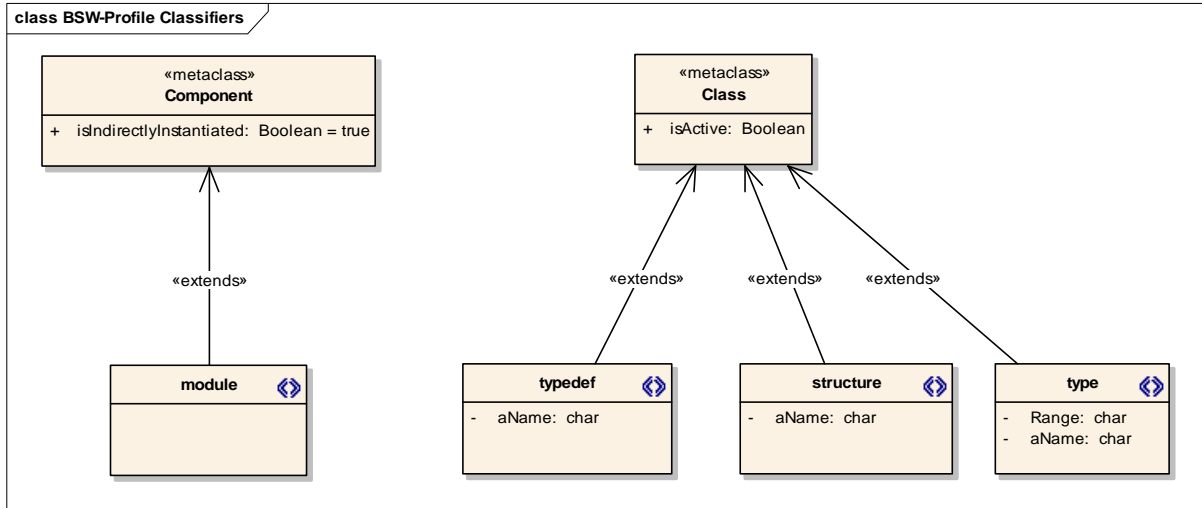
## 5 BSW UML Profil

This section provides an overview of the profile.

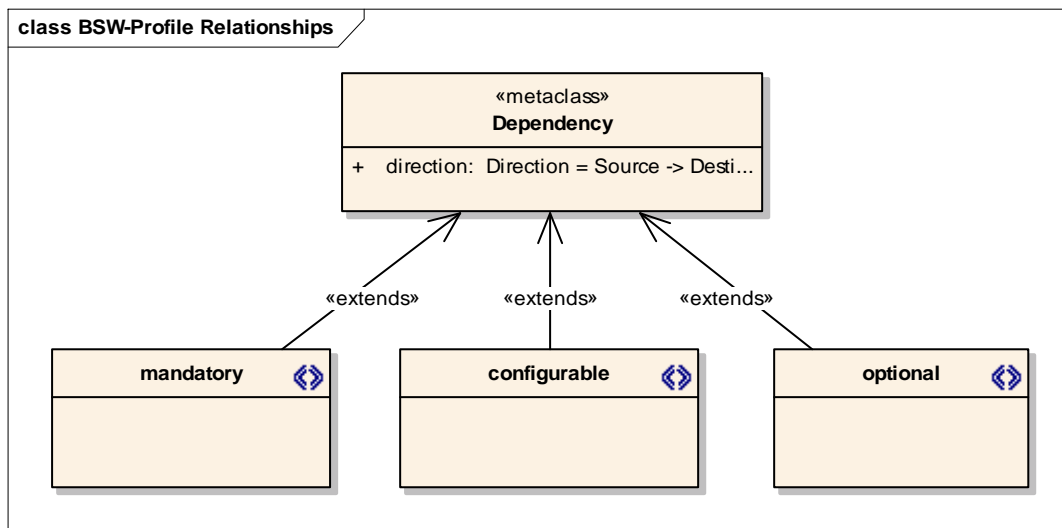
### 5.1.1 Stereotypes callback, function and scheduled function



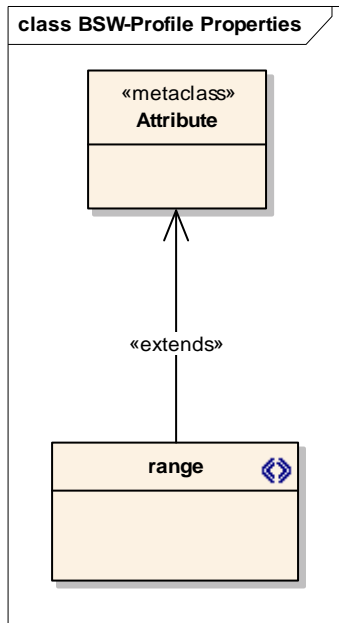
### 5.1.2 Stereotypes module, type, typedef and structure



### 5.1.3 Stereotypes mandatory, configurable and optional



### 5.1.4 Stereotype range



## 6 Administrative Info

Last used Requirements ID is [BSW\_UMLGuide\_00067]